HP Network Node Manager i Software

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

Software Version: NNMi 9.0x Patch 2 (9.01)

Online Help: Help for Administrators

Document Release Date: September 2010

Software Release Date: September 2010

Disclaimer for PDF Version of Online Help

This document is a PDF version of the online help. This PDF file is provided so you can easily print multiple topics from the help information or read the online help in PDF format.

Note: Some topics do not convert properly to PDF format. You may encounter formatting problems or unreadable text in certain document locations. Those problem topics can be successfully printed from within the online help.

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notice

© Copyright 2008–2010 Hewlett-Packard Development Company, L.P.

Trademark Notices

Acrobat® is a trademark of Adobe Systems Incorporated.

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

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

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

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

UNIX® is a registered trademark of The Open Group.

Oracle Technology — Notice of Restricted Rights

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

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

Acknowledgements

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

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

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

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

Table of Contents

Disclaimer for PDF Version of Online Help 2
Legal Notices 3
Table of Contents 4
Introduction for NNMi Administrators 20
Administrator Tools in the Console 22
Quick Start Configuration Wizard 23
Configuration Workspaces 23
Enable or Disable Configurations 26
Lookup Fields 27
Use the Quick Find Window 28
Use Autocomplete 28
Create a Configuration Object Instance Using the Form Toolbar 29
Delete One or More Objects 29
Actions Provided by NNMi 30
NNMi Processes and Services 44
About Each NNMi Process 45
Verify that NNMi Processes Are Running 45
Stop or Start an NNMi Process 45
About Each NNMi Service 46
Verify that NNMi Services are Running 47
Stop or Start NNMi Services 49
Introduction to IPv6 in NNMi-Advanced 50
Use NNMi Help Anywhere, Anytime 51
Connecting Multiple NNMi Management Servers (NNMi Advanced) 52
Regional Manager: Create a Forwarding Filter (Limit the available Node information) 53
Global Manager: Connect to a Regional Manager 55
Global Manager: Configure the Regional Manager Connection 56
Disconnect Communication with a Regional Manager 59
Troubleshoot Global Network Management 61
Clock Synchronization Issues (SSO / Global Network Management) 61
Determine the State of the Connection to a Regional Manager 62
Check the Health of Global Managers and Regional Managers 63

Error Messages About Regional Managers (NNMi Advanced) 64
Configuring Communication Protocol 66
Configure Default SNMP and ICMP Protocol Settings 67
Timeout / Retry Behavior Example for SNMP 72
Timeout / Retry Behavior Example for ICMP 73
Configure Default Community Strings (SNMPv1 or SNMPv2c) 73
Default Read Community String Form 75
Configure Default SNMPv3 Settings 76
Default SNMPv3 Settings form 77
Configure the Default Device Credentials (NNM iSPI NET) 78
Configure Regions (Communication Settings) 79
Communication Region Form 80
Configure Address Ranges for Regions 86
Configure Hostname Filters for Regions 89
Configure SNMPv1/v2 Community Strings for Regions 90
Configure SNMPv3 Settings for Regions 92
Communication Region SNMPv3 Settings form 93
Configure Credential Settings for Regions (NNM iSPI NET) 94
Configure Specific Nodes (Communication Settings) 95
Specific Node Settings Form (Communication Settings) 96
Configure SNMPv1/v2 Community Strings for a Specific Node 103
Configure SNMPv3 Settings for a Specific Node 104
Configure Credential Settings for a Specific Node (NNM iSPI NET) 105
Load Specific Node Settings from a File 106
Troubleshooting Communication Settings 108
Verify That All Nodes Support SNMP 108
Verify a Node's Communication Settings 109
Identify Unexpected Overrides of Communication Settings 110
Resolve Authentication Errors 111
Discovering Your Network 113
How Spiral Discovery Works 114
Rediscovery Intervals 116
Discovery Node Name Choices 116
Node Name Decision Tree 118
Discovery Seeds (as a starting point) 119

Ping Sweep (as a starting point) 119
Auto-Discovery Rules 120
Filters to Exclude Certain IP Addresses from Discovery 121
Filters to Exclude Certain Interfaces from Discovery 121
Subnet Connection Rules 122
Device Profiles and Discovery 124
Prerequisites for Discovery 124
SNMP Prerequisites 124
Well-Configured DNS Prerequisite 125
Use nslookup to Verify DNS Server Configurations 125
Exclude Problem Devices from nmlookup 126
IPv6 Addresses Prerequisite (NNMi Advanced) 126
Determine Your Approach to Discovery 127
Do Not Use Auto-Discovery Rules 128
Routers and Switches Discovered 128
All SNMP Devices Discovered 129
Everything Discovered 130
All Devices from a Specific Vendor Discovered 132
Limit Sources of Neighbor Information 133
Exclude Problem IP Addresses from Discovery 134
Exclude Problem Interfaces from Discovery 134
Specific System Object IDs Not Discovered 134
Configure Device Profiles 135
Configure Discovery 136
Adjust the Rediscovery Interval 138
Configure Whether to Delete Unresponsive Nodes 138
Configure Ping Sweep Global Settings 139
Configure the Node Name Strategy 140
Configure Auto-Discovery Rules 141
Configure Basic Settings for the Auto-Discovery Rule 143
IP Address Ranges for Auto-Discovery 146
SNMP System Object ID Ranges for Discovery 150
Configure Subnet Connection Rules 152
Subnet Connection Rules Provided by NNMi 154
Configure an Excluded IP Addresses Filter 155

Configure an Excluded Interfaces Filter ... 157
Specify Discovery Seeds: Initial Routers or Specific Nodes to be Discovered ... 158
In the Console, Configure Discovery Seeds ... 159
With a Seed File, Add Multiple Discovery Seeds ... 162
From the Command Line, Add Discovery Seeds ... 164
Examine Discovery Results ... 166
Check Initial Progress of Discovery ... 166
Node Discovery State Check ... 166
Verify Success of Discovery Seeds ... 167
Discovery Seed Results ... 167
Examine Discovery Inventory ... 169
Examine Layer 2 Discovery Results ... 170
Examine Layer 3 Discovery Results ... 170
Keep Your Topology Accurate ... 171
Delete Nodes ... 172
Delete Discovery Seeds ... 173
Detect Interface Changes (renumbering issues) ... 174
Add or Delete a Layer 2 Connection ... 176
Start Discovery On-Demand ... 179
Creating Groups of Nodes or Interfaces ... 181
Create Node Groups ... 181
In the Console, Create Node Groups ... 182
Specify Node Group Additional Filters ... 183
Node Groups of IPv4 or IPv6 Addresses ... 190
Guidelines for Creating Additional Filters for Node Groups ... 191
Add Boolean Operators in the Additional Filters Editor ... 193
In a CSV File, Define Node Groups ... 196
Create Interface Groups ... 197
Specify Interface Group Additional Filters ... 198
Interface Groups of IPv4 or IPv6 Addresses ... 206
Guidelines for Creating Additional Filters for Interface Groups ... 207
Add New IfTypes (Interface Types) to the List ... 208
Node Groups Provided by NNMi ... 209
Node Groups As Predefined View Filters ... 209
Island Node Groups ... 211

Interface Groups Provided by NNMi ... 212
Add Custom Attributes to a Node or Interface Object ... 213
Add Custom Attributes to Multiple Nodes or Interfaces ... 214
Monitoring Network Health ... 215
About the State Poller ... 215
The NNMi Causal Engine and Monitoring ... 216
Configure Monitoring Behavior ... 216
Set Global Monitoring ... 217
Set Default Monitoring ... 219
Configure Interface Monitoring ... 223
Configure Threshold Monitoring for Interfaces (HP Network Node Manager iSPI Performance for Metrics Software) ... 228
Determine Reasonable Threshold Settings (HP Network Node Manager iSPI Performance for Metrics Software) ... 231
Examples of Threshold Monitoring (HP Network Node Manager iSPI Performance for Metrics Software) ... 232
Configure Node Monitoring ... 234
Configure Threshold Monitoring for Nodes (HP Network Node Manager iSPI Performance for Metrics Software) ... 240
Configure Node Group Status ... 243
Configure Percentage Values for the Target Status ... 244
Node Group Status Settings Form ... 244
Monitor Router Redundancy Groups (NNMi Advanced) ... 246
Current Health of the State Poller Service ... 246
Verify Monitoring Configuration Settings ... 246
Monitor Status Distribution for Network Objects ... 249
Stop or Start Managing a Node, Interface, Card, Address, or Node Component ... 250
View the Management Mode for Objects in Your Network ... 251
Unmanaged Nodes View ... 252
Unmanaged Interfaces View ... 253
Unmananaged Addresses View ... 253
Unmanaged Cards View ... 253
Unmanaged Node Components View ... 254
How NNMi Assigns the Management Mode to an Interface, Card, Node Component, or Address ... 254
How the NNMi Administrators Change a Management Mode ... 256
Understand the Effects of Setting the Management Mode to Not Managed or Out of Service ... 257

Configuring the NNMi User Interface ... **259**
Controlling Access to NNMi ... 262
Configure Sign-In Access ... 262
Roles Provided in NNMi ... 263
Determine which NNMi Role to Assign ... 263
Control Menu Access ... 264
Control Access with NNMi Accounts ... 266
Change Password, Name, or Role Assignment ... 267
Control Access Using Both Directory Service and NNMi ... 268
Change Role Assignment for a Directory Service User Name ... 270
Control Access with a Directory Service ... 271
Disable a User's Access to NNMi (User Principals) ... 272
Using the Principal Form ... 272
Troubleshoot NNMi Access ... 273
View the Users who are Signed In to NNMi ... 273
Audit NNMi User Activity ... 274
Restore the Administrator Role ... 275
Restore the System Role ... 275
Set Up Command Line Access ... 275
Communicate to Your Team ... 277
Open the Console ... 277
Sign Into the NNMi Console ... 278
Sign Out from the Console ... 278
Configure Maps ... 279
Define Default Map Settings ... 279
Define Node Group Map Settings ... 281
Node Group Map Settings Form ... 281
Configure Basic Settings for a Node Group Map ... 282
Configure the Connectivity to be Displayed for a Node Group Map ... 286
Configure Background Image Information for a Node Group Map ... 287
Background Image Sources in Node Group Maps ... 290
Scale Background Images in Node Group Maps ... 291
Troubleshoot URLs When Specifying a Background Image ... 291
Configure a Path View Map ... 291
Configure Default Settings for Line Graph ... 295

Configure Menus ... 296
Configure Menu Items ... 296
Configuring Trap Forwarding ... 297
Configure NNMi Security Settings for SNMPv3 Trap Forwarding ... 297
Configure Trap Forwarding Filters ... 299
Trap Forwarding Filter Form ... 300
Filter Form ... 300
Configure Trap Forwarding Destinations ... 301
Trap Forwarding Destination Form ... 302
Destination Filter Form ... 304
Trap Varbinds Provided by NNMi ... 305
Configuring Incidents ... 306
Manage Incidents Using Incident Configurations ... 307
How NNMi Gathers Incidents ... 307
The NNMi Causal Engine and Incidents ... 308
About the Event Pipeline ... 309
How NNMi Closes Incidents ... 310
Incident Configurations Provided by NNMi ... 312
Custom Incident Attributes Provided by NNMi (for Administrators) ... 312
SNMP Trap Incident Configurations Provided by NNMi ... 315
Remote NNM 6.x/7.x Event Configurations Provided by NNMi ... 328
Management Event Configurations Provided by NNMi ... 330
Incident Pair (Pairwise) Configurations Provided by NNM ... 339
Manage the Number of Incoming Incidents ... 341
Establish Criteria or Relationships for Incoming Incidents ... 342
Correlate Duplicate Incidents (Deduplication Configuration) ... 346
Deduplication Comparison Parameters Form ... 346
Track Incident Frequency (Rate: Time Period and Count) ... 347
Rate Comparison Parameters Form ... 347
About Pairwise Configurations ... 347
Incident Pair (Pairwise) Configurations Provided by NNM ... 348
Prerequisites for Pairwise Configurations ... 350
Pairwise Configuration Form (Correlate Pairs of Incidents) ... 351
Pair Item Configuration Form (Identify Incident Pairs) ... 353
Suppress Incident Configurations ... 355

Enrich Incident Configurations ... 355
Dampening Incident Configurations ... 356
Configure Custom Correlations ... 356
Configure a Correlation Rule ... 358
Configure a Parent Incident Filter for a Correlation Rule ... 360
Configure a Child Incident Filter for a Correlation Rule ... 367
Configure a Correlation Filter ... 375
Correlation Rule Example ... 382
Configure a Causal Rule ... 385
Configure a Child Incident for a Causal Rule ... 387
Configure a Child Incident Filter for a Causal Rule ... 389
Configure a Source Object Filter for a Causal Rule ... 397
Configure a Source Node Filter for a Causal Rule ... 404
Causal Rule Example ... 410
Configure an Action for an Incident ... 414
Lifecycle Transition Action Form ... 415
Valid Parameters for Configuring Incident Actions (Management Events) ... 415
Handling Special Characters in Action Arguments ... 419
Configure Diagnostics for an Incident (NNM iSPI NET) ... 421
Diagnostic Selections Form (NNM iSPI NET) ... 422
Diagnostics (Flows) Provided by NNM iSPI NET ... 422
Incident Configurations You Might Want to Enable ... 425
Generate Interface Disabled Incidents ... 426
Generate Card Disabled Incidents ... 426
Generate Performance Threshold Incidents (HP Network Node Manager iSPI Performance for Metrics Software) ... 426
Manage Incoming SNMP Traps ... 427
Configure Network Devices to Send SNMP Notifications to NNMi ... 428
Load SNMP Trap Incident Configurations ... 429
Load SNMP Trap Incident Configurations from the Command Line ... 429
Load SNMP Trap Incident Configurations using the Console ... 430
Control which Incoming Traps Are Visible in Incident Views ... 431
Handle Unresolved Incoming Traps ... 432
Analyze Trap Information (NNM iSPI NET) ... 433
Configure SNMP Trap Incidents ... 436

SNMP Trap Configuration Form ... 437
Configure Basic Settings for an SNMP Trap Incident ... 438
Specify the Incident Configuration Name (SNMP Trap Incident) ... 441
Specify the SNMP Object ID ... 441
SNMP Object ID Format for SNMP Traps ... 441
SNMP Object ID Format for SNMPv1 Generic Traps ... 442
SNMP Object ID Format for a Specific SNMPv1 Trap ... 442
Display an SNMP Trap or an NNM 6.x/7.x Event as a Root Cause Incident ... 443
Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident) ... 443
Create an Incident Category (SNMP Trap Incident) ... 444
Create an Incident Family (SNMP Trap Incident) ... 445
Specify the Incident Severity (SNMP Trap Incident) ... 446
Specify Your Incident Message Format (SNMP Trap Incident) ... 446
Valid Parameters for Configuring Incident Messages (SNMP Trap Incident) ... 446
Include Custom Incident Attributes in Your Message Format (SNMP Trap Incident) ... 452
Specify a Description for Your Incident Configuration (SNMP Trap Incident) ... 453
Configure Interface Settings for an SNMP Trap Incident ... 453
Configure Incident Suppression Settings for an Interface Group (SNMP Trap Incident) ... 454
Configure Incident Enrichment Settings for an Interface Group (SNMP Trap Incident) ... 462
Configure Custom Incident Attributes to Enrich an Incident Configuration (Interface Settings) (SNMP Trap Incidents) ... 465
Configure a Payload Filter to Enrich an Incident Configuration (Interface Settings) (SNMP Trap Incidents) ... 467
Configure Incident Dampening Settings for an Interface Group (SNMP Trap Incident) ... 472
Configure Incident Actions for an Interface Group (SNMP Trap Incident) ... 482
Configure a Payload Filter for an Incident Action (Interface Settings) (SNMP Trap Incidents) ... 483
Configure Node Settings for an SNMP Trap Incident ... 488
Configure Incident Suppression Settings for a Node Group (SNMP Trap Incident) ... 489
Configure Incident Enrichment Settings for a Node Group (SNMP Trap Incident) ... 501
Configure Custom Incident Attributes to Enrich an Incident Configuration (Node Settings) (SNMP Trap Incidents) ... 504
Configure a Payload Filter to Enrich an Incident Configuration (Node Settings) (SNMP Trap Incidents) ... 506
Configure Incident Dampening Settings for a Node Group (SNMP Trap Incident) ... 511
Configure Incident Actions for a Node Group (SNMP Trap Incident) ... 521

Configure a Payload Filter for an Incident Action (Node Settings) (SNMP Trap Incidents) ... 522
Configure Diagnostics Selections for a Node Group (SNMP Trap Incident) (NNM iSPI NET) ... 527
Configure Suppression Settings for an SNMP Trap Incident ... 529
Configure Enrichment Settings for an SNMP Trap Incident ... 542
Configure Dampening Settings for an SNMP Trap Incident ... 545
Configure Forward to Global Manager Settings for an SNMP Trap Incident (NNMi Advanced) ... 554
Configure Deduplication for an SNMP Trap Incident ... 562
Deduplication Comparison Parameters Form (SNMP Trap Incident) ... 566
Configure Rate (Time Period and Count) for an SNMP Trap Incident ... 567
Rate Comparison Parameters Form (SNMP Trap Incident) ... 569
Configure Actions for an SNMP Trap Incident ... 571
Lifecycle Transition Action Form (SNMP Trap Incidents) ... 572
Configure a Payload Filter for an Action (SNMP Trap Incidents) ... 574
Valid Parameters for Configuring Incident Actions (SNMP Trap Incident) ... 578
Configure Remote NNM 6.x/7.x Events ... 583
Configure Remote NNM 6.x and 7.x Management Stations ... 583
Remote NNM 6/x/7.x Event Configuration Form ... 584
Configure Basic Settings for a Remote NNM 6.x/7.x Event Incident ... 586
Specify the Incident Configuration Name (Remote 6.x/7.x Event) ... 588
Display an SNMP Trap or an NNM 6.x/7.x Event as a Root Cause Incident ... 588
Specify Category and Family Attribute Values for Organizing Your Incidents (Remote NNM 6.x/7x Events) ... 589
Create an Incident Category (Remote NNM 6.x/7.x Event) ... 589
Create an Incident Family (Remote NNM 6/x.7.x Event) ... 590
Specify the Incident Severity (Remote NNM 6.x/7.x Events) ... 591
Specify Your Incident Message Format (Remote NNM 6.x/7.x Events) ... 592
Valid Parameters for Configuring Incident Messages (Remote NNM 6.x/7.x Events) ... 592
Include Custom Incident Attributes in Your Message Format (Remote NNM 6.x/7.x Events) ... 597
Specify a Description for Your Incident Configuration (Remote NNM 6.x/7.x Events) ... 598
Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident ... 599
Configure Incident Suppression Settings for an Interface Group (Remote NNM 6.x/7.x Events) ... 600
Configure Incident Enrichment Settings for an Interface Group (Remote NNM 6.x/7.x Events) ... 608

Configure Custom Incident Attributes to Enrich an Incident Configuration (Interface Settings) (Remote NNM 6.x/7.x Events) 611

Configure a Payload Filter to Enrich an Incident Configuration (Interface Settings) (Remote NNM 6.x/7.x Events) 613

Configure Incident Dampening Settings for an Interface Group (Remote NNM 6.x/7.x Events) 618

Configure Incident Actions for an Interface Group (Remote NNM 6.x/7.x Event) 628

Configure a Payload Filter for an Incident Action (Interface Settings) (Remote NNM 6.x/7.x Events) 629

Configure Node Settings for a Remote NNM 6.x/7.x Event Incident 634

Configure Incident Suppression Settings for a Node Group (Remote NNM 6.x/7.x Events) 636

Configure Incident Enrichment Settings for a Node Group (Remote NNM 6.x/7.x Events) 647

Configure Custom Incident Attributes to Enrich an Incident Configuration (Node Settings) (Remote NNM 6.x/7.x Events) 650

Configure a Payload Filter to Enrich an Incident Configuration (Node Settings) (Remote NNM 6.x/7.x Events) 652

Configure Incident Dampening Settings for a Node Group (Remote NNM 6.x/7.x Events) 657

Configure Incident Actions for a Node Group (Remote NNM 6.x/7.x Events) 667

Configure a Payload Filter for an Incident Action (Node Settings) (Remote NNM 6.x/7.x Events) 668

Configure Diagnostics Selections for a Node Group (Remote NNM 6.x/7.x Events) 673

Configure Suppression Settings for a Remote NNM 6.x/7.x Event Incident 675

Configure Enrichment Settings for a Remote NNM 6.x/7.x Event Incident 684

Configure Dampening Settings for a Remote NNM 6.x/7.x Event Incident 687

Configure Forward to Global Managers Settings for a Remote 6.x/7.x Event Incident (NNMi Advanced) 696

Configure Deduplication for a Remote NNM 6.x/7.x Event Incident 707

Deduplication Comparison Parameters Form (Remote NNM 6.x/7.x Events) 711

Configure Rate (Time Period and Count) for a Remote NNM 6.x/7.x Event Incident 712

Rate Comparison Parameters Form (Remote NNM 6.x/7.x Events) 714

Configure Actions for a Remote NNM 6.x/7.x Event Incident 716

Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events) 717

Configure a Payload Filter for an Action (Remote NNM 6.x/7.x Events) 718

Valid Parameters for Configuring Incident Actions (Remote NNM 6.x/7.x Events) 723

Configure Management Events 728

Management Event Form 728

Configure Basic Settings for a Management Event Incident ... 730
Specify the Incident Configuration Name (Management Events) ... 732
Specify Category and Family Attribute Values for Organizing Your Incidents (Management Events) ... 733
Create an Incident Category (Management Events) ... 733
Create an Incident Family (Management Events) ... 734
Specify the Incident Severity (Management Events) ... 735
Specify Your Incident Message Format (Management Events) ... 736
Valid Parameters for Configuring Incident Messages (Management Events) ... 736
Include Custom Incident Attributes in Your Message Format (Management Events) ... 741
Specify a Description for Your Incident Configuration (Management Events) ... 742
Configure Interface Settings for a Management Event Incident ... 743
Configure Incident Suppression Settings for an Interface Group (Management Events) ... 744
Configure Incident Enrichment Settings for an Interface Group (Management Events) ... 752
Configure Custom Incident Attributes to Enrich an Incident Configuration (Interface Settings) (Management Events) ... 755
Configure a Payload Filter to Enrich an Incident Configuration (Interface Settings) (Management Events) ... 757
Configure Incident Dampening Settings for an Interface Group (Management Events) ... 762
Configure Incident Actions for an Interface Group (Management Events) ... 772
Configure a Payload Filter for an Incident Action (Interface Settings) (Management Events) ... 773
Configure Node Settings for a Management Event Incident ... 778
Configure Incident Suppression Settings for a Node Group (Management Events) ... 779
Configure Incident Enrichment Settings for a Node Group (Management Events) ... 787
Configure Custom Incident Attributes to Enrich an Incident Configuration (Node Settings) (Management Events) ... 790
Configure a Payload Filter to Enrich an Incident Configuration (Node Settings) (Management Events) ... 792
Configure Incident Dampening Settings for a Node Group (Management Events) ... 797
Configure Incident Actions for a Node Group (Management Events) ... 807
Configure a Payload Filter for an Incident Action (Node Settings) (Management Events) ... 808
Configure Diagnostics Selections for a Node Group (Management Events) ... 813
Configure Suppression Settings for a Management Event Incident ... 815
Configure Enrichment Settings for a Management Event Incident ... 826
Configure Dampening Settings for a Management Event Incident ... 829
Configure Deduplication for a Management Event Incident ... 841

Deduplication Comparison Parameters Form (Management Events) ... 845
Configure Rate (Time Period and Count) for a Management Event Incident ... 846
Rate Comparison Parameters Form (Management Events) ... 848
Configure Actions for a Management Event Incident ... 850
Lifecycle Transition Action Form (Management Events) ... 851
Configure a Payload Filter for an Action (Management Events) ... 852
Valid Parameters for Configuring Incident Actions (Management Events) ... 859
Troubleshoot Incident Configurations ... 863
View an Incident Configuration Report ... 864
Using Route Analytics Management Software (RAMS) with NNMi Advanced ... 868
Configure HP Route Analytics Management Software (NNMi Advanced) ... 868
HP RAMS MPLS WAN Configuration (NNMi Advanced) ... 869
Enabling the HP NNMi – HP RAMS MPLS WAN Integration ... 869
Prerequisites: ... 870
Changing the HP NNMi – HP RAMS MPLS WAN Integration Configuration ... 870
Disabling the HP NNMi – HP RAMS MPLS WAN Integration ... 870
HP NNMi – HP RAMS MPLS WAN Integration Configuration Form Reference ... 870
Extending NNMi Capabilities ... 872
Control the Actions Menu ... 872
Create Menu Nesting (in the Actions Menu) ... 873
Configure Menu Item Basic Details ... 874
Configure Menu Item Context Basic Details ... 877
Configure Launch Actions ... 878
W3C Rules for URLs ... 882
Attributes per Object Type for Full URLs ... 883
Capability Attributes in Full URLs ... 886
Custom Attributes in Full URLs ... 887
Custom Incident Attributes (CIAs) in Full URLs ... 888
Database Object Identifiers for Full URLs ... 889
Path View Attributes for Full URLs ... 890
Configure SNMP Line Graph Actions ... 890
MIB Specification Form ... 893
Configure JavaScript Actions ... 896
Configure Java Actions ... 897
Specify Optional Menu Item Enablement Filters ... 898

Examine Available MIBs and MIB Variables ... 911
Determine the MIBs Supported for a Node (for Administrators) ... 911
View the MIBs Loaded on the NNMi Management Server ... 912
Loaded MIBs View ... 912
MIB Form ... 913
Determine the MIB Variables Supported for a Node (for Administrators) ... 913
Display a MIB File's Contents (Administrators) ... 915
Upload MIB Files from the Console ... 915
Load MIBs ... 916
Load MIBs from the Console ... 916
Load MIBs from the Command Line ... 919
Configure MIB Expressions ... 919
MIB Expressions View ... 920
MIB Expression Form (Line Graph) ... 920
Test a MIB Expression (Line Graph) ... 924
Use the MIB Expression Editor (Line Graph) ... 925
Configure Custom Polling ... 929
Enable or Disable Custom Poller ... 930
Create a Custom Poller Collection ... 930
Configure Basic Settings for a Custom Poller Collection ... 932
Specify the MIB Variable Information for a Custom Poller Collection ... 935
MIB Expressions Form (Custom Poller) ... 936
Test a MIB Expression (Custom Poller) ... 938
Use the MIB Expression Editor (Custom Poller) ... 939
Configure Threshold Information for a Custom Poller Collection ... 943
Configure Comparison Maps for a Custom Poller Collection ... 945
Create a Policy ... 947
Create a Report Group (HP Network Node Manager iSPI Performance for Metrics Software) ... 950
Create a Report Collection (HP Network Node Manager iSPI Performance for Metrics Software) ... 950
Purchase an HP Network Node Manager i Smart Plug-in ... 953
Integrations with Other HP Products ... 954
Integration Configuration Form ... 954
Integrating NNMi Elsewhere with URLs ... 954
W3C Rules for URLs ... 954

Pass Environment Attributes ... 955
Launch the Console (showMain) ... 957
Launch a View (showView) ... 957
Launch an Incident View ... 961
Launch the All Incidents View Filtered by Node ... 963
Launch a Topology Maps Workspace View ... 965
Launch a Monitoring Workspace View ... 972
Launch a Troubleshooting Workspace View ... 975
Launch an Inventory Workspace View ... 982
Launch a Management Mode Workspace Views ... 985
Launch a Configuration Workspace View ... 988
Launch a Form (showForm) ... 990
Launch a Node Form ... 990
Launch an Interface Form ... 993
Launch an IP Address Form ... 994
Launch a Subnet Form ... 995
Launch an Incident Form ... 997
Launch a Node Group Form ... 998
Launch a Configuration Form ... 1000
Launch Menu Items (runTool) ... 1001
Launch the Actions: Communication Configuration Command ... 1002
Launch the Actions: Configuration Poll Command ... 1003
Launch the Actions: Line Graph (showLineGraph) ... 1004
Launch the Actions: Monitoring Settings Command ... 1006
Launch the Actions: Ping Command ... 1009
Launch the Actions: Status Details Command (for Node Groups) ... 1010
Launch the Actions: Status Poll Command ... 1011
Launch the Actions: Trace Route Command ... 1012
Actions: Execute a Launch Action ... 1013
Launch the Tools: MIB Browser (showMibBrowser) ... 1014
Launch the Tools: NNMi Status Command ... 1015
Launch the File: Sign-Out Command ... 1016
Launch the Tools: Sign-In/Out Audit Log Command ... 1016
Confirm that NNMi Is Running (cmd=isRunning) ... 1017
Maintaining NNMi ... 1018

Check NNMi Health ... 1018
Track Your NNMi Licenses ... 1019
Extend a Licensed Capacity ... 1020
About Environment Variables ... 1021
Export and Import Configuration Settings ... 1022
Export/Import Behavior and Dependencies ... 1022
Export a Snapshot of Your Configuration Settings ... 1027
Import Configuration Files to Restore Previous Settings ... 1028
Transfer Configuration Settings to Another NNMi Management Server ... 1030
Troubleshooting Imports of Configuration Files ... 1032
Back Up and Restore NNMi ... 1036
Archive and Delete Incidents ... 1037
Appendix A: Glossary Terms ... 1041
Appendix B: Index ... 1045

Introduction for NNMi Administrators

As an NNMi administrator, you can use the console to configure the items described in the following table.

Configure NNMi

What You Can Configure	Description
Device Profiles	HP provides well over three thousand pre-configured Device Profiles, one for each known sysObjectID at the time NNMi released. NNMi uses Device Profiles (which equate to sysObjectIDs) to control certain types of behavior. Using the **Device Profiles** option in the **Configuration** workspace, you can update Device Profile information. See "Configure Device Profiles" (on page 135)for more information.
Discovery	Using the **Discovery Configuration** option in the **Configuration** workspace, configure NNMi to discover only those devices that are important to you and your team. See "Discovering Your Network" (on page 113) for more information.
Filters	Using the **Node Groups** and **Interface Groups** options in the **Configuration** workspaces, define filters. These filters identify groups of devices. Use the filters to quickly locate information in views. See "Creating Groups of Nodes or Interfaces" (on page 181) for more information. You can also monitor the health of each group, see "Configure Monitoring Behavior" (on page 216).
Global Network Management	(*NNMi Advanced - Global Network Management feature*) Using the **Global Network Management** option in the **Configuration** workspace, you can configure NNMi to share the workload among multiple NNMi management servers in your network environment. See "Connecting Multiple NNMi Management Servers (NNMi Advanced)" (on page 52) .
ICMP and SNMP Communication Protocols	Using the **Communication Configuration** option in the **Configuration** workspace, provide the SNMPv1 or SNMPv2c community strings (read and write) for your network environment, or provide the SNMPv3 User Names for your network environment. Configure NNMi settings for timeout, retry, and port usage for ICMP and SNMP traffic. See "Configuring Communication Protocol" (on page 66) for more information.
Incidents	Using the **Incident Configuration** option in the **Configuration** workspace, review the many predefined incident configurations provided by NNMi . Edit any of the configurations provided by NNMi or create your own . See "Configuring Incidents" (on page 306) for more information.
Interface Groups	Using the **Interface Groups** option in the **Configuration** workspace, identify important devices. Interface Groups are filters for interface and IP address views. Interface Groups can also control how NNMi monitors network devices. See "Create Interface Groups" (on page 197) for more information.
Interface Types	Interface Type definitions cover all known industry-standard IANA ifType-MIB variables at the time of the release of NNMi. Using the **IfTypes** option in the **Configuration** workspace, add a new interface type to the NNMi list of Interface Type definitions. This option is useful if your team acquires new devices that contain new interface types not yet provided by NNMi. See "Add New IfTypes (Interface Types) to the List" (on page 208)for more information.

What You Can Configure	Description
Management Stations (6.x/7.x)	Using the **Management Stations (6.x/7.x)** option in the **Configuration** workspace, configure how events that are received from NNM 6.x or 7.x management stations are handled by NNMi . See "Configure Remote NNM 6.x and 7.x Management Stations" (on page 583) for more information.
MIB Expressions	Using the MIB Expressions option in the Configuration workspace, take a proactive approach to network management by using SNMP MIB Expressions to specify additional information that NNMi should poll. See "Configure MIB Expressions" (on page 919) for more information.
Monitoring	Using the **Monitoring Configuration** option in the **Configuration** workspace, define how and how often important devices are monitored by NNMi . See "Monitoring Network Health" (on page 215) for more information.
Node Groups	Using the **Node Groups** option in the **Configuration** workspace, identify important devices. You can then filter node, interface, IP address, and incident views by Node Group. You can also specify Node Groups when configuring monitoring and incidents. See "Create Node Groups" (on page 181) for more information.
Node Group Map Settings	Using the **User Interface Configuration** option in the **Configuration** workspace, specify the Node Group map configuration including the Node Group and background image to be used in a Node Group map. See "Define Node Group Map Settings" (on page 281) for more information.
Route Analytic Management Servers (RAMS)	(*NNMi Advanced*) Using the **RAMS Servers** option in the **Configuration** workspace, configure sources of Route Analytics Management Software data for NNMi to use. See "Using Route Analytics Management Software (RAMS) with NNMi Advanced" (on page 868).
Sign-In Access to NNMi	Using the **User Interface Configuration** option in the **Configuration** workspace, provide and control access to NNMi . Configure a name and password for each user. Assign a role to each user. **Tip**: If your environment manages user names and passwords with a directory service, NNMi can be configured to use Lightweight Directory Access Protocol (LDAP) instead of the settings in the **User Accounts** tab of the **Configuration** → **User Interface Configuration** option. See "Controlling Access to NNMi" (on page 262) for more information.
Status	Using the **Status Configuration** option in the **Configuration** workspace, configure how Node Group Status is calculated. You can choose to assign the Node Group the most severe status of any Node Group member or configure the percentage thresholds for one or more Node Group target statuses. See "Configure Node Group Status" (on page 243)for more information.
Trap Forwarding	Using the **Trap Forward Configuration** option in the **Configuration** workspace, configure trap forwarding filters and destinations. See "Configuring Trap Forwarding" (on page 297) for more information.
User Interface	Using the **User Interface Configuration** option in the **Configuration** workspace, configure the following user interface features: • User accounts • Default map settings

What You Can Configure	Description
	• Node Group map settings • Default Line Graph settings • Menus and menu items

NNMi provides a variety of tools to assist you with these configuration tasks. Each of these tools is described in the following table. You can extend NNMi using HP Network Node Manager i Software Smart Plug-ins (iSPIs) as described in "Extending NNMi Capabilities" (on page 872).

NNMi Administrator Tools

Tool	Description
Configuration Workspaces	The console provides a workspace for each kind of item you can configure in NNMi . See the preceding "Configure NNMi " table for more information.
Lookup Fields	Provided in forms, fields that include the icon provide access to a list of all available attribute values, and in some locations enable you to create attribute values. See "Lookup Fields" (on page 27) for more information.
Actions	Used to perform automated tasks on a single object or on a group of objects. For example, you can use the Actions menu to change the Management Mode of one or more nodes from **Managed** to **Out of Service**. Actions are available from table views, map views, and forms. See "Actions Provided by NNMi" (on page 30) for more information
NNMi Processes and Services	NNMi is built on a group of processes and services. You can list these processes and services. You can stop and start individual processes and services. See "NNMi Processes and Services" (on page 44) for more information.

Administrator Tools in the Console

When configuring settings for NNMi, you create configuration object instances. For example, to create a new URL action, you must create a new URL action instance. As another example, to specify configuration settings for discovery, you might create object instances that contain ranges of IP addresses that you want NNMi to use as hints for Spiral Discovery.

The console provides the following tools to assist you with configuration tasks:

- "Configuration Workspaces" (on page 23)
- "Lookup Fields" (on page 27)
- "Create a Configuration Object Instance Using the Form Toolbar" (on page 29)
- "Delete One or More Objects" (on page 29)

Quick Start Configuration Wizard

Note: Before you use the Quick Start Configuration Wizard, complete the initial configuration checklist. See **Help → Documentation Library → Installation Guide** for more information.

The Quick Start Configuration Wizard automatically runs immediately after Network Node Manager (NNMi) installation completes. Use the Quick Start Configuration Wizard to configure NNMi in a limited (or test) environment. The Quick Start Configuration Wizard helps you to complete the following initial set up tasks:

- Provide the *read community strings* for your SNMPv1 or SNMPv2c environment to enable "Get" commands
- Provide the USM settings for your SNMPv3 environment
- Discover a limited range of network nodes
- Set up an initial administrator account

You can launch the wizard using the following URL:

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/quickstart/`

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

Note: HP recommends that you run the Quick Start Configuration Wizard only one time immediately after NNMi installation.

After using the Quick Start Configuration Wizard to set up a test network, see "Configuration Workspaces" (on page 23) for information about completing additional NNMi configuration tasks.

Configuration Workspaces

NNMi administrators use the Configuration workspaces to configure the following items related to NNMi.

Note: On tables in configuration forms, if the cursor changes to indicate a hyperlink when you mouse over a column heading, you are able to sort the column's data. You cannot change the sort on some of the tables on the forms in the configuration workspace.

NNMi Configuration Workspaces

Name	Description
Communication Configuration	Use to configure how NNMi uses ICMP and SNMP in your network environment. See "Configuring Communication Protocol" (on page 66).
Discovery Configuration	Use to specify the devices to be discovered. See "Discovering Your Network" (on page 113).
Monitoring Configuration	Use to enable the NNMi State Poller. See "Monitoring Network Health" (on page 215).
Custom Poller	Use to configure SNMP MIB Expressions that specify additional information NNMi

Name	Description
Configuration	should poll. See "Configure Custom Polling" (on page 929).
Incident Configuration	Use to specify the information displayed with an incident, including its name, the message you want to be displayed, the way it should be categorized, its initial status, and how you want to identify duplicate traps. See "Configuring Incidents" (on page 306).
Trap Forwarding	Use to forward SNMP trap to other servers in your network environment. See "Configuring Trap Forwarding" (on page 297).
Status Configuration	Use to configure Node Group status calculations using either of the following methods: • Assign the Node Group the most severe status of any Node Group member. This is the default. • Configure the percentage thresholds for one or more Node Group target statuses. See "Configure Node Group Status" (on page 243).
User Interface Configuration	Use to configure many user interface features: • The NNMi console time-out interval. • The initial view that you want NNMi to display. • Specify that NNMi users must provide one of the following in the URL for accessing NNMi: ▪ The Fully Qualified Domain Name (FQDN) of the NNMi management server. ▪ Any hostname or IP address associated with the NNMi management server (NNMi automatically redirects these to the FQDN) • Whether NNMi displays unlicensed features that require a special license, such as NNMi Advanced. See "Configuring the NNMi User Interface" (on page 259). **Default Map Settings** tab - Use to configure the default settings for map views. These settings can be overridden for a specific map using the Node Group Map Settings tab. See "Configure Maps" (on page 279). **Default Line Graph Settings** tab - Use to configure the SNMP MIB data that you want to make available to your network operators in a graph format. This graph is available through the Actions menu and displays in real time. See "Configure Default Settings for Line Graph" (on page 295). **User Accounts** tab - Use to assign NNMi users to NNMi roles. See "Configure Sign-In Access" (on page 262). **Note**: If NNMi role assignments are stored in your environment's directory service database, this view is not needed and is not visible. See "Control Access with a Directory Service" (on page 271). **User Principals** tab - This view stores NNMi user names. Principal object instances are automatically created when you configure access to the NNMi console (using the User Accounts and Roles view) according to the following scenarios: "Control Access with NNMi Accounts" (on page 266) and "Control Access Using Both

Name	Description
	Directory Service and NNMi" (on page 268). **Note**: If NNMi role assignments are stored in your environment's directory service database, this view is not needed and should remain empty. See "Control Access with a Directory Service" (on page 271). Directory Service and NNMi" (on page 268). Directory Service and NNMi" (on page 268).
	Node Group Map Settings tab - Use to specify the Node Group and background image to be used in a Node Group map. Map settings include the following: • Node group name • The order in which Node Group maps should appear in the Topology workspace • Minimum role for saving edited locations for each node in the map • Refresh information • Connectivity information • Background image URL • Background image scale
	Menu Items tab - Use to make changes or additions to the items available in the Actions menu. See "Configure Menu Items" (on page 296).
Global Network Management	(*NNMi Advanced - Global Network Management feature*) Use to configure communication between Global Managers and Regional Managers in your network environment. See "Connecting Multiple NNMi Management Servers (NNMi Advanced)" (on page 52).
Node Groups	Use to group your devices for viewing and monitoring purposes. See "Create Node Groups" (on page 181).
Interface Groups	Use to group your devices for viewing and monitoring purposes. See "Create Interface Groups" (on page 197).
IfTypes	Use to determine the list of available interface types. NNMi administrators use these ifTypes to define Interface Groups. See "Add New IfTypes (Interface Types) to the List" (on page 208).
Device Profiles	Use to see and edit device profile information. Device profile information includes the SNMP object ID, model, and vendor. See "Configure Device Profiles" (on page 135).
Loaded MIBs	Use to determine the MIBs loaded on the NNMi management server. See "Examine Available MIBs and MIB Variables" (on page 911).
MIB Expressions	Use to determine the MIB Expressions available for Custom Poller or Line Graphs. See "Create a Custom Poller Collection" (on page 930) and "Configure SNMP Line Graph Actions" (on page 890).
RAMS Servers	(*NNMi Advanced*) Use to configure sources of Route Analytics Management Software data for NNMi to use. See "Using Route Analytics Management Software (RAMS) with NNMi Advanced" (on page 868).

Name	Description
Management Stations (6.x/7.x)	Use to configure NNMi to receive data from NNM 6.x or 7.x management stations in your network environment. See "Configure Remote NNM 6.x/7.x Events" (on page 583).

Enable or Disable Configurations

Using the **Actions** menu, you can enable or disable one or more of the following configurations:

Note: When you enable or disable a configuration, NNMi assigns the value **Customer** as the Author name. See Author form for important information.

Enable or Disable NNMi Configurations

Configuration	Configuration Workspace Option
SNMP Traps	Incident Configuration
Remote NNM 6.x/7.x Events	Incident Configuration
Management Events	Incident Configuration
Pairwise	Incident Configuration
Menus	User Interface Configuration
Menu Items	User Interface Configuration

To enable an NNMi configuration:

1. Navigate to the table view of the configurations you want to change. For example, select **User Interface Configuration** from the **Configuration** workspace and select the **Menus** tab.
2. To enable a configuration, click the Open icon in the row representing the configuration you want to enable.
3. Select **Actions** → **Enable Configuration**.

   If you are in the configuration form, NNMi selects Enabled ☑.

   If you are in the table view, NNMi displays a ✔check in the Enabled column for each instance selected.

To disable an NNMi configuration:

1. Navigate to the table view of the configurations you want to change. For example, select **User Interface Configuration** from the **Configuration** workspace and select the **Menus** tab.
2. Do one of the following:

   a. To disable a configuration, click the Open icon in the row representing the configuration you want to edit.

   b. To disable more than one configuration, click the Show View in New Window icon to open the table view in a new window. Select each configuration instance that you want to disable.
3. Select **Actions** → **Disable Configuration**.

   If you are in the configuration form, NNMi removes the check mark from Enabled ☐.

If you are in the table view, NNMi removes the ✔check mark in the Enabled column for each instance selected.

Lookup Fields

Lookup fields have the following icon: .

The Lookup field represents an associated object instance. For example, an Incident form has an associated Source Node attribute. Information about this source node is available in and accessed through the Lookup field.

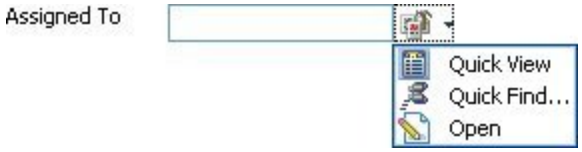
Possible Drop-Down Menu Options in Lookup Fields

Option	Description
Quick View	See a subset of information about the selected object. (Use Open to see all available information about this object.)
Quick Find	Display a list of valid choices for populating the current attribute field.
Open	Open the form for the related object instance that is currently selected in the lookup field. Review all attributes of the related object. Depending on your role, you can edit these attributes.
New	Create a new object instance to relate to the current object.

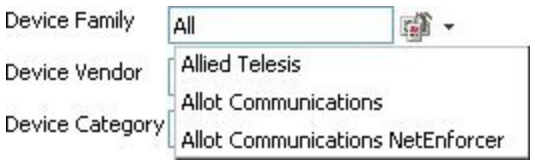
You can use Lookup fields in a variety of ways:

- **Read-only fields - to provide additional information about the associated object**. Click Quick View or Open to see the details of this object.

- **Selection fields - to change the association to another object instance**. Click Quick Find to select from a list of previously configured objects ("Use the Quick Find Window" (on page 28)).



Or type a case-sensitive string into the input box ("Use Autocomplete" (on page 28)).



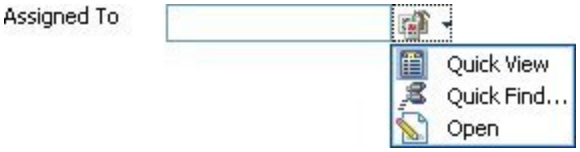
- **Read-write fields - create an entirely new object instance for this association**. Click New. An empty form opens for you to fill in, creating a new object instance.

Use the Quick Find Window

The Quick Find option is available only in Lookup fields that are modifiable. Use the Quick Find option to see the list of available object instances appropriate for populating the current Lookup field.

To list all existing object instances that could be related to the current object:

1. From the lookup field of interest, click the Look up icon:



2. Select Quick Find.

   NNMi displays a table view of object instances that are available to associate with to the current object instance.

3. In the Quick Find window, do one of the following:

   - Click the Make Empty icon to remove an association with this object. The Quick Find window closes, and the current lookup field is empty.
   - Click the Select This Item icon that precedes the table row containing an object instance. The Quick Find window closes, and the object instance you selected populates the current lookup field.
   - Click the Quick View icon to display a subset of available object attributes.
   - Click the Close icon to return to the previous form.

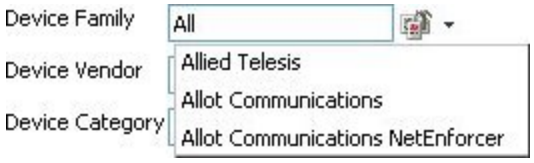
Use Autocomplete

The autocomplete feature is available only in Lookup fields that are modifiable. As you type, NNMi lists the available object instances for populating the current Lookup field.

To use the autocomplete feature:

1. Start typing the first few letters (case-sensitive) of the name of the object you want to associate with the current one.

The Lookup field displays a drop-down list below the input field. This list includes all potential existing objects with names that match the letters as you enter them.



2. Use the scroll arrows or the mouse to select from the displayed list.

   The selected object populates the lookup field and is now associated with the current object.

Create a Configuration Object Instance Using the Form Toolbar

You can save time by generating a new form from within another form. The new form is based on the object type for the original form and contains only the default values set by NNMi for particular attributes for that object. Any attributes that have no default value appear blank.

This tool is useful when you want to create multiple object instances that have similar attribute values.

To create a new object instance using the form toolbar:

1. Open the form representing the object of interest.
2. From the form toolbar, click the Save and New icon.

   A new form appears that contains the default attribute values for the object type represented by the original form.
3. Select the **Save and Close** icon to save your changes and return to the view.

Delete One or More Objects

Each row in a table view and each symbol in a map view represents an instance of the object type being displayed. For example, in a node view, each row of the table represents an instance of a node in your network.

NNMi administrators can delete object instances. For example, you might need to delete a node that is no longer being managed.

All objects related to the deleted object are also deleted. For example, if a node object is deleted, all of its interfaces, network addresses, and its SNMP Agent are deleted. Related connections with either zero or one end points remaining are deleted. Related subnets and VLANs with zero remaining members are deleted. The time required for NNMi to finish deleting depends on the number of objects or related objects being deleted.

To delete an object instance:

1. Select the object of interest:
   - In a table view, select the check box in the row that represents the object.
   - In a map view, click the map symbol.
   - In a form, proceed to step 2.
2. To delete the object, click the Delete icon.

   The object is deleted from the NNMi database and removed from the current view.

To delete multiple object instances:

1. Select the objects of interest:
   - In a table view, do one of the following:
     - Select the ☑ check box in the row that represents each object you want to delete.
     - Select the ☑ check box above the check-box column to select all objects in the view.
   - In a map view, CTRL-Click each map symbol.
2. To delete the objects, click the ✕ Delete icon.

   Note: For Node objects, you can use this method to delete up to 20 nodes at one time. You can use the command line method to delete any number of nodes. See "Delete Nodes" (on page 172) for more information.

   Tip: For all other objects, you can delete any number.

   Each object is deleted from the NNMi database and removed from the current view.

Actions Provided by NNMi

Note: (*NNMi Advanced - Global Network Management feature*) If your NNMi console is a Global Manager and the selected node is being managed by a Regional Manager (another NNMi management server in your network environment), some actions are not available.

The following tables describe the actions provided by NNMi:

Actions Provided for Incidents

Actions Provided for Nodes

Actions Provided for Interfaces

Actions Provided for Addresses

Actions Provided for Node Groups

Actions Provided for Interface Groups

Actions Provided for Custom Polled Instances

Actions Provided for Router Redundancy Member, Tracked Object, and Node Component

As shown in the table, the actions available depend on the object selected.

Note: You can also use the Actions menu to access views and possibly NNM 6.x/7.x features. See Access NNM 6.x and 7.x Features for more information about the available NNM 6.x/7.x actions.

Actions Provided for Incidents

Action	Description
Node Actions	Provides access to all of the actions available for a the Incident's Source Node. See Actions Provided for Nodes for more information.
Interface Actions	**Only available for incidents with the Source Object attribute value set to Interface**. Provides access to all of the actions available for an interface. See Actions Provided for Interfaces for more information.
IP Address Actions	**Only available for incidents with the Source Object attribute value set to IP Address.**

Action	Description
	Provides access to all of the actions available for an IP address. See Actions Provided for Addresses for more information.
Node Group Map	Displays the lowest level Node Group map to which the Source Node belongs. For example, if the node belongs to a *Child* Node Group, the *Child* Node Group displays. See Node Group Maps. If the Source Node is a member of more than one Node Group at the lowest level, NNMi prompts you to select the Node Group map you want to display. If the incident's Source Object is an Island Node Group, NNMi displays the Island Node Group map. See "Island Node Groups" (on page 211). **Note**: Incidents with the Source Object attribute value set to Island Node Group include **Remote site** in the incident message. See Island Node Group Map for more information. When the selected Source Node is not a member of any Node Group, and you select the **Node Group Map** action, NNMi displays an information message.
Path View	Displays a map showing the route between two specified nodes, using the Source Node as the starting point. **Note**: *NNMi Advanced*. Path View works only with IPv4 addresses. The NNMi Advanced IPv6 address values are not valid choices for Path View. Any devices in your network that are configured with IPv6 addresses cannot be displayed on Path View maps.
Source Node	Displays the Node form of the Source Node object instance.
Source Object	Displays the form of the source object instance.
Node Group Members	*Island Node Group incidents only*. Displays a table of the nodes that are members of the Island Node Group that is the Source Object for the selected incident. See "Island Node Groups" (on page 211). **Note**: Incidents with the Source Object attribute value set to Island Node Group include **Remote site** in the incident message.
Graph Custom Poller Results	Graphs all MIB expressions from each of the Custom Poller Collections associated with the selected incident's Source Node.
Delete	Deletes the selected Incident object or objects (maximum 20). To delete more than 20 nodes, see the nnmnodedelete.ovpl Reference Page.
Change Lifecycle → In Progress	Changes the lifecycle state to **In Progress** for the selected incident.
Change Lifecycle →Completed	Changes the lifecycle state to **Completed** for the selected incident.
Change Lifecycle	Changes the lifecycle state to **Closed** for the selected incident.

Action	Description
→Close	
Assign → Assign Incident	Displays a list of registered users to select from. This user name appears in the **Assigned To** column of the incident view.
Assign → Own Incident	Assigns the incident to the current user. This user name appears in the **Assigned To** column of the incident view.
Assign → Unassign Incident	Removes the user name from the **Assigned To** column of the incident view.
Incident Configuration Reports →	Displays a report of the configuration settings that define this Incident. See "View an Incident Configuration Report" (on page 864) for more information.
Open Incident Configuration	Displays the selected Incident's configuration form.
Run Diagnostics (iSPI NET only)	(*HP Network Node Manager iSPI Network Engineering Toolset Software*) When installed, NNM iSPI NET gathers diagnostic information from the Source Node.

Actions Provided for Nodes

Action	Description
Graphs	Displays a pre-configured graph of real-time data for a selected node. NNMi provides a set of Line Graph that are configured to display real-time SNMP data. See Line Graphs Provided by NNMi for more information. Line Graph graphs can also come from the following sources: • Your NNMi administrator might configure additional graphs. • NNM iSPI software.
Ping (from server)	Tests whether a node is reachable using the ping command. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Ping issues an ICMP request from the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Ping accesses that Regional Manager (NNMi management server) and issues the ICMP request. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Trace Route (from server)	Traces a route path from the using the traceroute command.

Action	Description
	(*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Trace Route issues a request from the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Trace Route accesses that Regional Manager (NNMi management server) and issues the request in a manner appropriate for the operating system in use on the Regional Manager. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Telnet (from client)	Uses Transmission Control Protocol (TCP) protocol from the computer that launched your current browser (not the NNMi management server) to open a Telnet (teletype network) virtual terminal command-line interface from the selected node or Source Node of the selected object. See Establish Contact with a Node.
Communication Settings	Displays the communication configuration information for the selected node. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Communication Settings displays a report, provided by the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Communication Settings accesses that Regional Manager (NNMi management server) and requests the report. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Monitoring Settings	Checks the monitoring configuration settings that were set for a particular node, interface, or IP address. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Monitoring Settings displays a report, provided by the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Monitoring Settings accesses that Regional Manager (NNMi management server) and requests the report. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
List Supported MIBs	Display a list of the MIBs (Management Information Base) supported by a selected node. See "Determine the MIBs Supported for a Node (for Administrators)" (on page 911) and Determine a Node's Supported MIBs (MIB Browser) for more information.
Browse MIB	The MIB Browser displays the responses to NNMi's SNMP requests made to a particular node in your network environment.

Action	Description
Status Poll	Instructs NNMi to gather real-time data for all the information that NNMi uses to calculate Status for each selected Node (maximum 10). A window for each Node displays with a report about which information was gathered. The NNMi administrator determines the list of information gathered by establishing Monitoring configuration settings. See "Monitoring Network Health" (on page 215) for more information. Note the following: • To see the resulting Node status, see Verify Current Status of a Device. • Using **Actions → Status Poll** does not affect the timing of the Polling interval configured for the device. **Tip**: The nnmstatuspoll.ovpl command line tool does the same thing. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Status Poll results are provided by the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Status Poll requests an updated *copy* of the configuration information from the Regional Manager, then the Global Manager displays the results. **Note**: You do not need to sign-in to the Regional Manager.
Configuration Poll	Runs a real-time configuration check of the selected device to detect any changes since the last discovery cycle. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Configuration Poll results are provided by the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Configuration Poll requests an updated *copy* of the configuration information from the Regional Manager, then the Global Manager displays the results. **Note**: You do not need to sign-in to the Regional Manager.
Open from Regional Manager	Issues a request to the Regional Manager (the NNMi management server that is responsible for monitoring the selected node) asking to display the Node form of the selected object. **Note**: You must sign into that Regional Manager unless your network environment provides Single Sign-On (SSO) to that Regional Manager.
Regional Manager Console	Issues a request to the Regional Manager (the NNMi management server that is responsible for monitoring the selected node) asking to display the NNMi console. **Note**: You must sign into that Regional Manager unless your network environment provides Single Sign-On (SSO) to that Regional Manager.

Action	Description
Delete	Deletes the selected object or objects. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Delete removes the Node object (and all related object data) from the Global Manager's database. • Node managed by a Regional Manager = Actions → Delete removes the *copy of the Node object* (and all related object data) from the Global Manager's database. **Note**: If you need to delete this Node object from the Regional Manager's database, click Actions → Open from Regional Manager and delete the Node object. You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Management Mode → Manage	Changes the Management Mode of the selected node to **Managed**. Leaves the Direct Management Mode of any contained interfaces and addresses unchanged. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Management Mode → Manage modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions → Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Management Mode → Manage (Reset All)	Changes the Management Mode of the selected node to **Managed**. Sets the Direct Management Mode of all contained interfaces and addresses to **Inherited**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Management Mode → Manage (Reset All) modifies the Node object plus all associated interface objects and address objects in the Global Manager's database. • Node managed by a Regional Manager = Use Actions → Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.

Action	Description
Management Mode → Unmanage	Changes the Management Mode of the node to **Not Managed**. Leaves the Direct Management Mode of any contained interfaces and addresses unchanged. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Management Mode → Unmanage modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions → Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Management Mode → Out of Service	Changes the Management Mode of the selected node to **Out of Service**. Leaves the Direct Management Mode of any contained interfaces or addresses unchanged. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Management Mode → Out of Service modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions → Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Run Diagnostics (iSPI NET only)	(*HP Network Node Manager iSPI Network Engineering Toolset Software*) When installed, NNM iSPI NET gathers diagnostic information on the current node.
Show Attached End Nodes	Displays information about the end nodes that NNMi determines are attached to the specified switch. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: The results are based on the current information in the NNMi database of the Global Manager (which contains *copies of Node objects* from all Regional Managers).

Actions Provided for Interfaces

Action	Description
Graphs	Displays a pre-configured graph of real-time data for a selected interface. NNMi provides a set of Line Graphs that are configured to display real-time SNMP data. See Line Graphs Provided by NNMi for more information. Line Graphs can also come from the following sources: • Your NNMi administrator might configure additional graphs.

Action	Description
	• NNMi SPI software.
Monitoring Settings	Checks the monitoring configuration settings that were set for a particular node, interface, or address. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Communication Settings displays a report, provided by the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Communication Settings accesses that Regional Manager (NNMi management server) and requests the report. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Communication Settings	Displays the communication configuration information for the Source node. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Communication Settings displays a report, provided by the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Communication Settings accesses that Regional Manager (NNMi management server) and requests the report. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Management Mode → Manage	Changes the Direct Management Mode of the interface to **Inherited**. Leaves the Direct Management Mode of any associated addresses unchanged. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Management Mode → Manage modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions → Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.

Action	Description
Management Mode →Manage (Reset All)	Changes the Management Mode of the interface to **Inherited**. Changes the Direct Management Mode of any associated addresses to **Inherited**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Management Mode → Manage (Reset All) modifies the Node object plus all associated interface objects and address objects in the Global Manager's database. • Node managed by a Regional Manager = Use Actions → Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Management Mode → Unmanage	Changes the Management Mode of the interface to Not Managed. Leaves the Direct Management Mode of any associated addresses unchanged. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Management Mode → Unmanage modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions → Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Management Mode → Out of Service	Changes the Management Mode of the interface to **Out of Service**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Management Mode → Out of Service modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions → Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.

Actions Provided for Addresses

Action	Description
Ping (from server)	Tests whether a node is reachable using the ping command. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Ping issues an ICMP request from the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Ping accesses that Regional Manager (NNMi management server) and issues the ICMP request. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Trace Route (from sever)	Traces a route path using the traceroute command. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Trace Route issues a request from the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Trace Route accesses that Regional Manager (NNMi management server) and issues the request in a manner appropriate for the operating system in use on the Regional Manager. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Telnet (from client)	Uses Transmission Control Protocol (TCP) protocol from the computer that launched your current browser (not the NNMi management server) to open a Telnet (teletype network) virtual terminal command-line interface from the selected node or Source Node of the selected object. See Establish Contact with a Node.
Communication Settings	Displays the communication configuration information for the Source node. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Communication Settings displays a report, provided by the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Communication Settings accesses that Regional Manager (NNMi management server) and requests the report. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Monitoring Settings	Checks the monitoring configuration settings that were set for a particular node, interface, or address. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Monitoring Settings displays a report, provided by the Global Manager (NNMi management server).

Action	Description
Ping (from server)	Tests whether a node is reachable using the ping command. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Ping issues an ICMP request from the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Ping accesses that Regional Manager (NNMi management server) and issues the ICMP request. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Trace Route (from sever)	Traces a route path using the traceroute command. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Trace Route issues a request from the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Trace Route accesses that Regional Manager (NNMi management server) and issues the request in a manner appropriate for the operating system in use on the Regional Manager. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Telnet (from client)	Uses Transmission Control Protocol (TCP) protocol from the computer that launched your current browser (not the NNMi management server) to open a Telnet (teletype network) virtual terminal command-line interface from the selected node or Source Node of the selected object. See Establish Contact with a Node.
	• Node managed by a Regional Manager = Actions → Monitoring Settings accesses that Regional Manager (NNMi management server) and requests the report. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.

Action	Description
Ping (from server)	Tests whether a node is reachable using the ping command. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Ping issues an ICMP request from the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Ping accesses that Regional Manager (NNMi management server) and issues the ICMP request. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Trace Route (from sever)	Traces a route path using the traceroute command. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Trace Route issues a request from the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Trace Route accesses that Regional Manager (NNMi management server) and issues the request in a manner appropriate for the operating system in use on the Regional Manager. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Telnet (from client)	Uses Transmission Control Protocol (TCP) protocol from the computer that launched your current browser (not the NNMi management server) to open a Telnet (teletype network) virtual terminal command-line interface from the selected node or Source Node of the selected object. See Establish Contact with a Node.
Configuration Poll	Runs a real-time configuration check of the selected device to detect any changes since the last discovery cycle. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Configuration Poll results are provided by the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Configuration Poll requests an updated *copy* of the configuration information from the Regional Manager, then the Global Manager displays the results. **Note**: You do not need to sign-in to the Regional Manager.
Management Mode → Manage	Changes the Direct Management Mode of the address to **Inherited**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Management Mode → Manage modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions → Open from Regional Manager to set Management Mode on the Regional Manager that is responsible

Action	Description
Ping (from server)	Tests whether a node is reachable using the ping command. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Ping issues an ICMP request from the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Ping accesses that Regional Manager (NNMi management server) and issues the ICMP request. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Trace Route (from sever)	Traces a route path using the traceroute command. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Trace Route issues a request from the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Trace Route accesses that Regional Manager (NNMi management server) and issues the request in a manner appropriate for the operating system in use on the Regional Manager. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Telnet (from client)	Uses Transmission Control Protocol (TCP) protocol from the computer that launched your current browser (not the NNMi management server) to open a Telnet (teletype network) virtual terminal command-line interface from the selected node or Source Node of the selected object. See Establish Contact with a Node.
	for this Node. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Management Mode → Unmanage	Changes the management mode of the address to **Not Managed**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Management Mode → Unmanage modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions → Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.

Action	Description
Ping (from server)	Tests whether a node is reachable using the ping command. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Ping issues an ICMP request from the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Ping accesses that Regional Manager (NNMi management server) and issues the ICMP request. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Trace Route (from sever)	Traces a route path using the traceroute command. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Trace Route issues a request from the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Trace Route accesses that Regional Manager (NNMi management server) and issues the request in a manner appropriate for the operating system in use on the Regional Manager. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.
Telnet (from client)	Uses Transmission Control Protocol (TCP) protocol from the computer that launched your current browser (not the NNMi management server) to open a Telnet (teletype network) virtual terminal command-line interface from the selected node or Source Node of the selected object. See Establish Contact with a Node.
Management Mode → Out of Service	Changes the management mode of the address to **Out of Service**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Management Mode → Out of Service modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions → Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note**: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.

Actions Provided for Node Groups

Action	Description
Node Group Map	Displays a current map of all nodes that belong to the selected Node Group.

Action	Description
Show Members	Displays a list of all nodes that belong to the selected Node Group.
Show All Incidents	Checks for any Incidents associated with the selected Node Group.
Show All Open Incidents	Checks for any open Incidents associated with the selected Node Group.
Status Details	Displays a report about the status of all members of the selected Node Group. See Check Status Details for a Node Group.

Actions Provided for Interface Groups

Action	Description
Show Members	Displays a list of all nodes that belong to the selected Node Group.

Actions Provided for Custom Polled Instances

Action	Description
Graph Polled Instance	Graphs the line representing the Custom Poll results for the selected Custom Polled Instance. See Display a Line Graph for a Custom Polled Instance.

Actions Provided for Router Redundancy Member, Tracked Object, and Node Component

Action	Description
Monitoring Settings	Checks the monitoring configuration settings that were set for the Router Redundancy Member, Tracked Object, or Node Component. See Verify Monitoring Configuration Settings and View the Monitoring Configuration Details.

NNMi Processes and Services

NNMi is built on a group of processes and services. For information about each process or service, see the following:

- "About Each NNMi Process" (on page 45)
- "About Each NNMi Service" (on page 46)

To verify that everything is running properly, you can use the ovstatus command:

- "Verify that NNMi Processes Are Running" (on page 45)

About Each NNMi Process

HP Network Node Manager Processes

Process Name	Description
OVsPMD	The control process that manages all the other NNMi processes.
pmd	Event Post Master daemon. This process routes events from the producers to the consumers. Producers of events are NNM 6.x/7.x management stations and processes. Consumers of events are the event pipeline and third party applications.
ovjboss	The process that controls the jboss application server that contains all of the NNMi Services (see "About Each NNMi Service" (on page 46) for more information).
nnmaction	The process that controls the Action Server. The NNMi Action Server runs any actions configured for incidents. See "Configure an Action for an Incident" (on page 414) for more information about incident actions. See also the nnmaction Reference Page for more information
nmsdbmgr	NMS Database Manager. Controls the NNMi embedded database, including periodic database connectivity testing.
nmsdbmgr	NMS Database Manager. Controls the NNMi embedded database, including periodic database connectivity testing.

Verify that NNMi Processes Are Running

After you install Network Node Manager, a group of processes run on the NNMi management server.

To verify that all NNMi processes are running, do one of the following:

1. Select **Tools** → **NNMi Status** to display a report.
2. At the command line, type: **ovstatus –c**

   See the ovstatus Reference Page for more information.

Review the list of processes to ensure that all are running. For more information about each process, see "About Each NNMi Process" (on page 45).

Stop or Start an NNMi Process

You can stop and start NNMi processes from the command line. See the ovstop and ovstart Reference Pages for more information.

Caution: If your NNMi management server participates in a high availability (HA) environment, under certain circumstances, you should not use `ovstop` or `ovstart`. Before using either of these commands, see the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

To stop or start an NNMi process:

At the command line, type the appropriate command (see "About Each NNMi Process" (on page 45)):

```
ovstop <process name>
```

```
ovstart <process name>
```

Note: If you use `ovstop` and `ovstart` without providing a process name, NNMi stops and starts all NNMi processes.

To generate a list of process names, see "Verify that NNMi Processes Are Running" (on page 45).

About Each NNMi Service

NNMi Services run inside the ovjboss process. The ovjboss process controls the jboss application server that contains all of the NNMi services.

HP Network Node Manager Services

ovjboss Service Name	Description
CommunicationParametersStatusService	Tracks internal statistics for measuring SNMP and ICMP configuration performance.
IslandSpotterService	Automatically discovers any Island Node Groups using Layer 2 connectivity information in the topology. An Island Node Group is a group of fully-connected nodes discovered by NNMi, and NNMi determines this group is not connected to the rest of the topology.
ManagedNodeLicenseManager	Responsible for ensuring that the number of managed nodes does not exceed the NNMi licensed capacity limit.
ModelChangeNotificationAdapter	Emits notifications when certain model changes happen (discovery seeds, global settings, Spiral Discovery configuration, management node).
MonitoringSettingsService	Calculates how to monitor each device based on the Monitoring Configuration settings.
NamedPoll	NMS Named Poll Service. Used to trigger immediate state polls for monitored objects. Used by the Causal Engine during neighbor analysis and interface up/down investigations.
NmsApa	NMS Active Problem Analyzer (APA) service determines the root cause of network problems and reports the root cause to the NMS Event Service. The Causal Engine is a key component of the NNMi APA service.
NmsDisco	NMS Discovery Service. Adds new devices to the database and keeps the configuration of the managed devices up to date in the database by periodically rechecking the configuration of the devices. State Poller uses the Discovery service results to determine what to monitor. The Causal Engine depends on the Discovery service to monitor node configurations. The Causal Engine uses the configuration information when calculating status and root cause. NNMi uses the information provided by the Discovery service to maintain current device configuration information.

ovjboss Service Name	Description
NmsEvents	NMS Events Service. Populates and manages the information displayed in the incident table. The information displayed comes from the other NNMi services that are running on your system. The incidents are filtered so you see only the most important information about your network.
NmsEventsConfiguration	Handles incident configuration changes.
NmsModel	NMS Topology Model Service. Enables communication between NNMi services and the NNMi database.
SpmdjbossStart	The SpmdjbossStart service interacts with the OVsPMD process during startup (ovstart), shutdown (ovstop), and reporting on the status of the ovjboss services (ovstatus –v ovjboss). **Caution**: If your NNMi management server participates in a high availability (HA) environment, under certain circumstances, you should not use `ovstop` or `ovstart`. Before using either of these commands, see the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals` .
StagedIcmp	Used by the State Poller to ping IP addresses using the Internet Control Message Protocol (ICMP). Also used by auto-discovery if Ping Sweep is enabled.
StagedSnmp	Used by the State Poller and Discovery to perform Simple Network Management Protocol (SNMP) read-only queries.
StatePoller	NMS State Poller Service. State Poller collects measurements that assess the current state of discovered devices. This information is provided for the Causal Engine to use when calculating device health.

HP Network Node Manager iSPI Network Engineering Toolset Software Services (*NNM iSPI NET*)

ovjboss Service Name	Description
RbaManager	(*HP Network Node Manager iSPI Network Engineering Toolset Software*) Used for diagnostics.

Verify that NNMi Services are Running

After you install Network Node Manager, a group of services run on the NNMi management server. For information about each service, see "About Each NNMi Service" (on page 46).

To verify that all NNMi services are running, do one of the following:

- Select **Tools** → **NNMi Status** to display a report.
- At the command line, type:

  ovstatus –v ovjboss

  See the ovstatus Reference Page for more information.

Review the list of services to ensure that all are running.

"`Service is started`" means this service is working properly.

"`Service is stopped`" means this service/process is not running.

If you see any of the messages in this list, investigate the log files and look for the keyword **Exception** (within the log file for the parent `ovjboss` process and the log file for the specific service, possible services are listed in the table below):

"Service is in created state"
"Service is in failed state"
"Service is in registered state"
"Service is in destroyed state"
"Service is in started state"
"Service is in starting state"
"Service is in stopped state"
"Service is in stopping state"
"Service is in unregistered state"

Log files are found in the following location. If your NNMi management server participates in a high availability (HA) environment, click here for more information:

1. Before opening the log file, first identify the `HA_MOUNT_POINT` for your NNMi environment.
2. At the command line, type:

   Windows:
   `%NnmInstallDir%/misc/nnm/ha/nnmhaclusterinfo.ovpl NNM -config -get HA_MOUNT_POINT`

   UNIX:
   `opt/OV/misc/nnm/ha/nnmhaclusterinfo.ovpl NNM -config -get HA_MOUNT_POINT`
3. At the command line, type the following (`/DataDir/` is the literal path):

   `<HA_MOUNT_POINT>/DataDir/log/nnm`

- **Windows:**
  `%NnmDataDir%\log\nnm\<name>.%g.%u.log`
- **UNIX:**
  `/var/opt/OV/log/nnm/<name>.%g.%u.log`

%g = Zero (0) for active log files. Any other number means an archived log file from previous restarts or from reaching log file size limits. See logging.properties for information about controlling the number of archives saved for each log file or for controlling the size of each log file.

%u = Zero (0) unless the parent `ovjboss` process failed during a logging session. While NNMi is logging information, it creates a file named `nnm.0.0.log.lck` (the lock file). NNMi deletes the lock file when it finishes logging. If a `nnm.0.0.log.lck` file already exists, NNMi creates `nnm.0.1.log.lck` file and writes to the `nnm.0.1.log` file.

The parent ovjboss process generates the following log files:

- `ovjboss.log` and `ovjboss.log.old`
- `jbossServer.log` and `jbossServer.<date>.log`

Note: Each restart creates a new `ovjboss.log` and overwrites the `ovjboss.log.old`. Each day a new `jbossServer.log` file is created, and the previous day's file is renamed by inserting a date stamp `jbossServer.<date>.log`

Stop or Start NNMi Services

You can stop or start all NNMi services at the same time. You cannot start and stop individual services. See the ovstop and ovstart Reference Page for more information.

Caution: If your NNMi management server participates in a high availability (HA) environment, under certain circumstances, you should not use `ovstop` or `ovstart`. Before using either of these commands, see the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

To stop or start the NNMi services:

At the command line, type the command:

```
ovstop ovjboss
```

```
ovstart ovjboss
```

Introduction to IPv6 in NNMi-Advanced

IPv6 upgrades IPv4 features and allows for vastly more address space, built-in security, and enhanced support for streaming media and peer-to-peer applications.

When your network environment includes both IPv4 and IPv6, your NNMi administrator can configure NNMi to automatically detect and monitor both types of addresses, whether devices are IPv4-only, IPv6-only, or dual-stack (both).

Note: The NNMi administrator must enable IPv6 with a setting in the `nms-jboss.properties` file. For more information, see the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

If enabled for IPv6, NNMi-Advanced allows IPv6 addresses and address ranges in the following:

- SNMP communication (address ranges and specific addresses), see "Configuring Communication Protocol" (on page 66). The NNMi administrator specifies whether NNMi prefers IPv4 or IPv6 addresses when selecting the Management Address (settings in the `nms-jboss.properties` file).
- Discovering address information about devices in your network (Include or exclude address ranges and specific addresses. Use IPv6 for discovery seeds.). See "Discovering Your Network" (on page 113) and "IPv6 Addresses Prerequisite (NNMi Advanced)" (on page 126).

  Note: IPv6 addresses are automatically excluded from Ping Sweep if you configure NNMi to use Ping Sweep as a starting point for discovery. NNMi detects the IPv6 addresses later in the discovery process. IPv6 address cannot be used when manually configuring representations of subnet connections that NNMi cannot otherwise detect (the optional Subnet Connection Rules aspect of Discovery).
- Monitoring a subset of the discovered devices using Node Group and Interface Group filters (through address filters and specific address lists), see "Monitoring Network Health" (on page 215).
- NNMi uses ICMPv6 communication for IPv6 Address fault monitoring.
- Configure NNMi fault monitoring to generate incidents about a subset of the discovered devices using Node Group and Interface Group filters (address filters and specific addresses), see "Configuring Incidents" (on page 306).

NNMi documents the associations between IPv6 Addresses, Subnets, Interfaces, and Nodes and presents consolidated IPv4 and IPv6 information. See Accessing Device Details.

NNMi provides Layer 2 Neighbor Views, Layer 3 Neighbor Views, and Topology Maps of IPv4 and IPv6 devices combined.

Note: Path View does not include IPv6 addresses.

The NNMi console's **Actions → Ping (from server)** and **Actions → Trace Route (from server)** menu items work with both IPv4 and IPv6. See Test Node Access (Ping) and Find the Route (traceroute).

Use NNMi Help Anywhere, Anytime

The NNMi Help system can run independently from the console. Simply unzip the files into any convenient location.

To locate the NNMi Help files, on the NNMi management server, navigate to the location appropriate for the NNMi management server's operating system (see table).

Location of the NNMi Help System

Operating System	NNMi Help System Files
Windows	%NnmInstallDir%\nonOV\jboss\nms\server\nms\deploy\nnmDocs_en.war
UNIX	/opt/OV/nonOV/jboss/nms/server/nms/deploy/nnmDocs_en.war

To access Help independently from the console:

1. Copy the web archive file `nnmDocs_en.war` to any convenient location.
2. At the command prompt, navigate to the directory where you placed the `nnmDocs_en.war` file. To extract the help directory structure and files, type:

   ```
   jar xvf nnmDocs_en.war
   ```

   Tip: You can also use WinZip on Windows to decompress the `nnmDocs_en.war` file.
3. Navigate to and open the `/htmlHelp/nmHelp/nmHelp.html` file.
4. The NNMi Help system runs as usual in the default browser window.

To Access a PDF version of the NNMi online help:

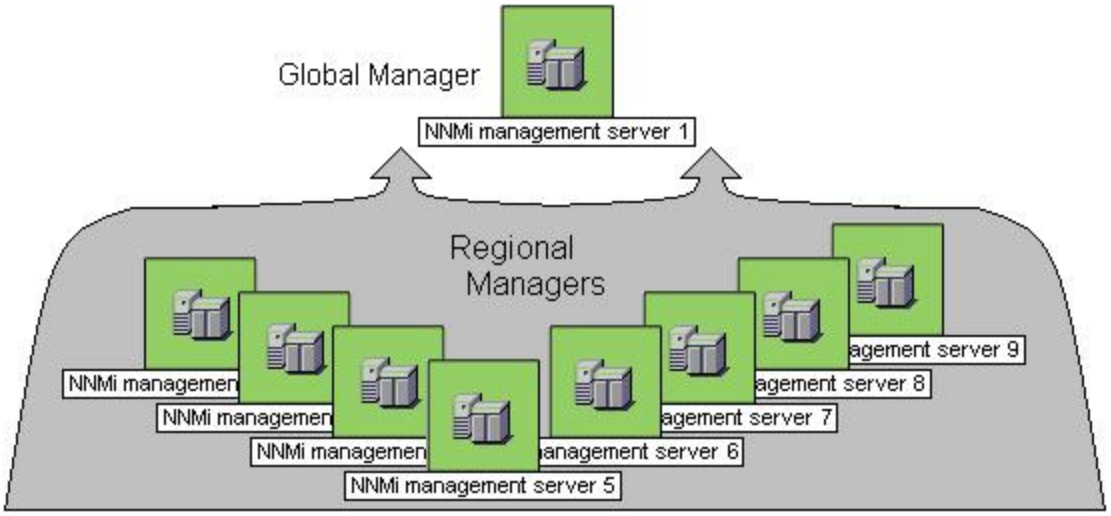
Go to: `http://h20230.www2.hp.com/selfsolve/manuals`

This site requires that you register for an HP Passport ID. To obtain an HP Passport ID, go to:

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

Connecting Multiple NNMi Management Servers (*NNMi Advanced*)

The Global Network Management feature enables you to configure NNMi to share the workload among multiple NNMi management servers in your network environment. For more information about the Global Network Management feature, click here:



(*NNMi Advanced*) There are many benefits to using the NNMi Global Network Management feature:

- Provides safe and secure communication among multiple NNMi management servers.
- Provides a central big-picture view of your corporate-wide network on the Global Manager for 24-hour/7-days-per-week coverage.
- Easy to set up:
  - Each Regional Manager administrator specifies *all Node object data* or *a specific Node Group* for participation at the Global Manager level.
  - Each Global Manager administrator specifies which Regional Managers are allowed to contribute information.
- Automatically combines topology from multiple NNMi management servers on the Global Manager, but keeps management responsibilities separate. (No duplication, the responsible NNMi Management server is clearly identified per Node.)
- Generates and manages Incidents independently on each server (generated within the context of topology available on each server).
- Regional Manager administrators can configure specific SNMP traps or NNM 6.x/7.x Events to be forwarded from Regional Managers to Global Managers.

Review the Global Network Management deployment choices in the *HP Network Node Manager i Software Deployment Reference* (available at: `http://h20230.www2.hp.com/selfsolve/manuals`).

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, the UNIX (HP-UX / Linux / Solaris) tool Network Time Protocol Daemon (NTPD) or one of the available Windows operating system tools.Review the Global Network Management deployment choices in the *HP*

Network Node Manager i Software Deployment Reference, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

To configure Global Network Management, do the following:

1. Navigate to the **Global Network Management** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Global Network Management**.
2. Do one of the following:
   - **Global Manager**. If you are the NNMi administrator for the **Global Manager**, decide which Regional Managers are allowed to forward network information to that Global Manager (NNMi management server). See "Global Manager: Connect to a Regional Manager" (on page 55). Each Regional Manager retains management responsibilities for the forwarded nodes. The Global Manager might or might not directly manage a set of network devices.
   - **Regional Manager**. If you are the NNMi administrator for the **Regional Manager**, you control the following aspects of communication with the Global Manager:
     - Forward information about *all* Node object data or *only data about Nodes belonging to one Node Group*. See "Regional Manager: Create a Forwarding Filter (Limit the available Node information)" (on page 53).

       Note: Incidents associated with the specified Nodes are not forwarded to the Global Manager. *Each server maintains an independent group of incidents.*
     - Forward specific SNMP traps and NNM 6.x/7.x Events to the Global Manager. See "Configure Forward to Global Manager Settings for an SNMP Trap Incident (NNMi Advanced)" (on page 554) and "Configure Forward to Global Managers Settings for a Remote 6.x/7.x Event Incident (NNMi Advanced)" (on page 696).
3. Click **Save and Close** to apply your changes.

After Global Network Management is set up in your network environment:

- To troubleshoot any issued with Global Network Management, see "Troubleshoot Global Network Management" (on page 61).
- To determine which Nodes are monitored by each NNMi management server, see View the NNMi Management Servers' Domain List.
- To determine which Incidents were forwarded to the Global Manager, see Monitor Incidents in a Global Network Management Environment (NNMi Advanced).

Regional Manager: Create a Forwarding Filter (Limit the available Node information)

(*NNMi Advanced - Global Network Management feature*) As administrator of the Regional Manager, you can specify which Node object data Global Managers can access:

To provide all Node object data to Global Managers in your environment, click here.

Do nothing. NNMi automatically forwards all Node object data unless a Forwarding Filter is defined.

To limit available Node object data by creating a Forwarding Filter, click here.

(*NNMi Advanced - Global Network Management feature*) The Global Manager and the Regional Manager

maintain separate sets of data. Conclusions about each Node are derived from the available data and can sometimes be different. Regional Managers forward the results of each Auto-Discovery cycle to the Global Manager. The Regional Manager can have a Node Group filter configured to limit the amount of data that is forwarded to the Global Manager. Filters are usually unnecessary for Global Network Management. Do not filter out nodes that are important for connectivity in your network environment to ensure NNMi has the data needed for accurate root cause analysis.

1. Navigate to the **Global Network Management** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Global Network Management** form.
2. Select the **Forwarding Filter** tab.
3. Click the **Node Group** Lookup icon and select one of the options from the drop-down menu:
   - Quick View to display summary information for the currently configured (selected) Node Group name.
   - Quick Find to view and select from the list of all existing Node Groups (for more information see "Use the Quick Find Window" (on page 28)).
   - Open to display the details of the currently configured (selected) Node Group (see Node Group form for more information).
   - New to create a new Node Group (see "Create Node Groups" (on page 181) for more information).
4. Click **Save and Close**.
5. Global Managers in your network environment can now access only information about the Nodes in the specified Node Group. If any Global Managers have previously gathered a wider range of Node object data, that extra data is automatically removed from the Global Managers database.

   To verify that your Forwarding Filter is working as expected, wait until the next NNMi rediscovery cycle finishes on your NNMi management server and then log onto the Global Manager. Follow the directions in View the NNMi Management Servers' Domain List. You should see only the members of the Node Group specified as your Forwarding Filter.

Incidents associated with the specified Nodes are not forwarded to the Global Manager. *Each server maintains an independent group of incidents.*

Regional Manager administrators can make exceptions to this for SNMP traps and NNM 6.x/7.x events. The administrator must specifically configure forwarding to the Global Managers:

- "Configure Forward to Global Manager Settings for an SNMP Trap Incident (NNMi Advanced)" (on page 554)
- "Configure Forward to Global Managers Settings for a Remote 6.x/7.x Event Incident (NNMi Advanced)" (on page 696)

To identify these specifically forwarded SNMP traps and NNM 6.x/7.x events on the Global Manager, see Monitor Incidents in a Global Network Management Environment (NNMi Advanced).

Global Manager: Connect to a Regional Manager

(*NNMi Advanced - Global Network Management feature*) As administrator, you can set up this NNMi management server as a Global Manager that displays information from other NNMi management servers (Regional Managers).

Tip: If the group of nodes being managed by a Regional Manager overlaps with the nodes already being managed by the Global Manager, the duplicate information from the Regional Manager is not imported into the Global Manager's database. If two Regional Managers are managing the same node, only the first instance to be forwarded is added to the Global Manager's database.

To enable communication from this NNMi management server to another in your network:

1. Prerequisite:

   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, the UNIX (HP-UX / Linux / Solaris) tool Network Time Protocol Daemon (NTPD) or one of the available Windows operating system tools.

   Review the Global Network Management deployment choices in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

2. Complete the required steps described in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`, then navigate to the **Global Network Management** form.

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Select the **Global Network Management** form.

3. Select the **Regional Manager Connections** Tab.

4. Do one of the following:

   - To create a new configuration, click the New icon.
   - To edit a configuration, click the Open icon in the row representing the configuration you want to edit.
   - DO NOT delete a configuration (the Delete icon). See "Disconnect Communication with a Regional Manager" (on page 59) for more information.

5. In the **Regional Manager** form, provide the basic configuration settings (see basic settings table).

6. From the Connection tab, navigate to the **Regional Manager Connection** form (see "Global Manager: Configure the Regional Manager Connection" (on page 56) for more information). Do one of the following:

   - To create a new connection, click the New icon.
   - To edit a connection, select a row, click the Open icon.
   - To delete a connection configuration, select a row and click the Delete icon.

7. Click **Save and Close** to return to the Regional Manager form.

8. Click **Save and Close** to return to the Global Network Management form.

9. Click **Save and Close**. NNMi establishes communication with the specified Regional Manager. That NNMi management server now forwards information about discovery and monitoring results to this NNMi management server.

Tip: To verify that the connection is working, see "Determine the State of the Connection to a Regional Manager" (on page 62).

Basic Settings for this Regional Manager (NNMi Management Server)

Attribute	Description
Name	Type a meaningful name for this configuration record about the Regional NNMi management server. For example: • The name your team uses to refer to the Regional NNMi management server. • The company site being managed by the Regional Manager. • The geographic area (Japan or Germany) being managed by the Regional Manager. The text you type appears in the Node view and NNMi Management Server view. This text string also appears in the Nodes by Management Server view's drop-down filter. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. No spaces are allowed. **Note**: Communicate this Name attribute value to your team so they understand the relationship between this name and the NNMi management server's DNS name (used to log onto that NNMi management server).
Description	*Optional*. Provide any description that would be useful for communication purposes within your team. Type a maximum of 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Global Manager: Configure the Regional Manager Connection

(*NNMi Advanced - Global Network Management feature*) As administrator, you configure how this NNMi management server communicates with another NNMi management server in your network environment (the Regional Manager).

Tip: If the group of nodes being managed by a Regional Manager overlaps with the nodes already being managed by the Global Manager, the duplicate information from the Regional Manager is not imported into the Global Manager's database. If two Regional Managers are managing the same node, only the first instance to be forwarded is added to the Global Manager's database.

To configure the communication connection to another NNMi management server:

1. Prerequisite:

   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, the UNIX (HP-UX / Linux / Solaris) tool Network Time Protocol Daemon (NTPD) or one of the available Windows operating system tools.

Review the Global Network Management deployment choices in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

2. Navigate to the **Regional Manager Connection** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Global Network Management** form
   c. Select the **Regional Manager Connections** tab.
   d. Do one of the following:
      - To create a new configuration, click the New icon.
      - To edit a configuration, click the Open icon in the row representing the configuration you want to edit.
      - DO NOT delete a configuration (the Delete icon). See "Disconnect Communication with a Regional Manager" (on page 59) for more information.
   e. In the **Regional Manager** form, navigate to the Connections tab. Do one of the following:
      - To create a new connection, click the New icon.
      - To edit a connection, select a row, click the Open icon.
      - To delete a connection configuration, select a row and click the Delete icon.
3. Provide the connection configuration settings (see connection configuration settings table).

   Note: If the Regional Manager participates in a high-availability (HA) environment, enter configuration settings for each server in the high-availability group (application fail-over).
4. Click **Save and Close** to return to the Regional Manager form.
5. Click **Save and Close** to return to the Global Network Management form.
6. Click **Save and Close**. NNMi establishes communication with the Regional NNMi management server. The Regional Manager forwards information about discovery and monitoring results.

Tip: To verify that the connection is working, see "Determine the State of the Connection to a Regional Manager" (on page 62).

Connection Configuration Settings for a Regional NNMi Management Server

Attribute	Description
Hostname	The official *fully-qualified-domain-name* of the Regional Manager (the NNMi management server). To verify the correct value, do one of the following: • Log onto the Regional Manager, select **Help** → **System Information**, and navigate to the **Server** tab. Use the value displayed in the Official Fully Qualified Domain Name (FQDN) field. • Use the nnmofficialfqdn.ovpl command. **Note**: If you want NNMi to use secure sockets layer encryption (HTTPS) to access this Regional NNMi management server, the value must match the hostname as specified in that server's SSL Certificate. For information about establishing the required trust relationship, see the "Global Network Management" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at:

Attribute	Description
	`http://h20230.www2.hp.com/selfsolve/manuals`. NNMi uses this hostname for communication with the Regional NNMi management server and to construct URL Actions. See "Authentication Requirements for URLs Access". See also "Actions Provided by NNMi" (on page 30) and read about these actions: • **Actions → Regional Manager Console** (opens the NNMi console) • **Actions → Open from Regional Manager** (opens the Node form)
Use Encryption	If ☐ disabled, NNMi uses hypertext transfer protocol (HTTP) and plain sockets to access this Regional NNMi management server. If ☑ enabled, NNMi uses secure sockets layer encryption (HTTPS/SSL) to access this Regional NNMi management server.
HTTP(S) Port	The Global Manager initiates all communication sockets. The Global Manager needs access to the following default TCP ports on each Regional Manager: **Non-Encrypted** • jboss.http.port = 80 • jboss.bisocket.port = 4457 • jboss.jmsControl.port = 4458 **Encrypted** • jboss.https.port = 443 • jboss.sslbisocket.port = 4459 • jboss.ssljmsControl.port = 4460 To determine the current port number configuration or change port settings, access the Regional Manager and look in the nms-local.properties file. See the nnm.ports Reference Page for more information. If ☐ Use Encryption is disabled (previous attribute), enter the port number for HTTP access to the NNMi console on the Regional NNMi management server. For example `http://<serverName>:<portNumber>/nnm/` If ☑ Use Encryption is enabled (previous attribute), enter the port number for HTTPS access to the NNMi console on the Regional NNMi management server. For example `https://<serverName>:<portNumber>/nnm/`
User Name	Type the user name required for NNMi sign-in for the **system** account on this Regional NNMi management server.

Attribute	Description
User Password	Type the password for the NNMi **system** account on this Regional NNMi management server. **Note**: NNMi encrypts the password and displays asterisks for this attribute. If you want to change the password, first clear the asterisks displayed in the Password attribute and enter the new Password value.
Ordering	A numeric value. NNMi checks for configuration settings in the order you define (lowest number first). NNMi uses the first match found for each address. Provide a unique connection ordering number for each Regional Manager configuration. Any duplicate Ordering numbers are checked in random order, for example that group of Regional Manager Connections can be checked in any order during each discovery cycle. **Tip**: It is recommended that ordering numbers are incremented by 10s or 100s to provide flexibility over time.

Disconnect Communication with a Regional Manager

(*NNMi Advanced - Global Network Management feature*) As administrator, you can disconnect communication between a Global Manager (NNMi management server) and a Regional Manager (another NNMi management server within your network environment).

To disconnect communication with a Regional Manager:

1. On the Global Manager (NNMi management server), navigate to the **Global Network Management** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Global Network Management** form.
2. Select the **Regional Manager Connections** tab.
3. Click the Open icon in the row representing the configuration you want to edit.

   In the **Regional Manager** form, delete all Connection objects:
   a. Select the **Connections** tab.
   b. Select all Connection records, and click the Delete icon.
4. Click **Save and Close** to return to the Global Network Management form. NNMi disables communication from this Global Manager (NNMi management server) to that Regional Manager (NNMi management server).
5. In the **Regional Manager Connections** tab, note the **Name** attribute value for that connection configuration (case-sensitive). You need to type this text string to replace <*RegionalNNMiServerName*> in a later step.
6. Click **Save and Close**.
7. On the Global Manager (NNMi management server), at the command line, type the following command (see "Delete Nodes" (on page 172) and nnmnodedelete.ovpl for more information):

Note: The original *node records* on the Regional Manager (NNMi management server) are not affected. Only the *copy of the node records* will be deleted from the Global Manager's database.

If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access" (on page 275) for more information.

Windows:
`%NnmInstallDir%\bin\nnmnodedelete -rm <RegionalNNMiServerName> -u <NNMiadminUserName> -p <NNMiadminPassword>`

UNIX:
`/opt/OV/bin/nnmnodedelete -rm <RegionalNNMiServerName> -u <NNMiadminUserName> -p <NNMiadminPassword>`

NNMi searches the Global Manager's database for all nodes that this Regional Manager is responsible for monitoring in your network environment. NNMi removes the node records from the Global Manager's database (these node records represent information *forwarded from* the Regional Manager). NNMi removes all associated data:

- Any interface or IP address information belonging to a deleted node.
- Any discovery seeds that match the name or IP address of a deleted node (unless you use the `nnmnodedelete -keepSeed` option).

Each Incident associated with the deleted Node is modified in the following ways, but not deleted from the NNMi database: The **Status** attribute changes to **Closed**. The **Correlation Notes** indicate the deletion of the associated node, interface, or address. The **RCA State** attribute changes to **FALSE**. **Note**: Incidents generated from SNMP traps or NNM 6.x/7.x Events (received from the deleted Node) appear in the Incident views, but remain unresolved.

To remove the Incidents from your NNMi database, follow the instructions in "Archive and Delete Incidents" (on page 1037) to delete "Closed" Incidents. You will be deleting all "Closed" Incidents, not just the "Closed" Incidents associated with this Regional Manager.

8. On the Global Manager (NNMi management server), remove the configuration record for this Regional Manager.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Global Network Management** form.
   c. Select the **Regional Manager Connections** tab.
   d. Select the row that represents the Regional Manager (NNMi management server) that should no longer communicate with this NNMi management server (the Global Manager), and click the Delete icon.
   e. Click **Save and Close**.
9. NNMi no longer requests information about discovery and monitoring results from that Regional Manager.

Note: The NNMi management server that is no longer one of the Regional Managers is still fully-functioning, but communication between the two NNMi management servers is now disabled.

Note: Traps from that Regional Manager are still forwarded to the Global Manager if configured to do so, see "Configure Trap Forwarding Destinations" (on page 301). Disable any trap forwarding that you no longer need.

Troubleshoot Global Network Management

(*NNMi Advanced - Global Network Management feature*) The Global Manager and the Regional Manager maintain separate sets of data. Conclusions about each Node are derived from the available data and can sometimes be different. Regional Managers forward the results of each Auto-Discovery cycle to the Global Manager. The Regional Manager can have a Node Group filter configured to limit the amount of data that is forwarded to the Global Manager. Filters are usually unnecessary for Global Network Management. Do not filter out nodes that are important for connectivity in your network environment to ensure NNMi has the data needed for accurate root cause analysis.

- The Global Manager might know information about why a connection from one site to another is down, but the Regional Manager just knows that the router connected to that remote site has an interface that is down. Use **Actions** → **Regional Manager Console** to see the other perspective.
- When troubleshooting a Node on the Global Manager, you can use **Actions** → **Open from Regional Manager** to see the latest Node information on the Regional Manager.

(*NNMi Advanced - Global Network Management feature*) This group of help topics can help you troubleshoot any problems with Global Network Management:

- "Clock Synchronization Issues (SSO / Global Network Management)" (on page 61)
- "Determine the State of the Connection to a Regional Manager" (on page 62)
- "Check the Health of Global Managers and Regional Managers" (on page 63)

Watch for these Incidents (error messages):

- "Error Messages About Regional Managers (NNMi Advanced)" (on page 64)
- $hostName Message Queue Size Exceeded Limit
- Forwarded Incident Rate Exceeded Limit
- $queueName Queue Size Exceeded Limit

See also these topics in NNMi Help for Operators:

- Is the Global Network Management Feature Enabled?
- View the NNMi Management Servers' Domain List

Clock Synchronization Issues (SSO / Global Network Management)

(Single Sign-On and *NNMi Advanced - Global Network Management feature*)

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, the UNIX (HP-UX / Linux / Solaris) tool Network Time Protocol Daemon (NTPD) or one of the available Windows operating system tools.

Review the Global Network Management deployment choices in the *HP Network Node Manager i Software Deployment Reference*, which is available at:
`http://h20230.www2.hp.com/selfsolve/manuals`.

- For clock issues when creating Regional Manager Connections, click here.

  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 `nnm.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>.
  ```

- If Regional Manager Connections break after running successfully, click here.

  Perhaps the clocks have drifted apart. Check the `nnm.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.
  ```

Determine the State of the Connection to a Regional Manager

(*NNMi Advanced - Global Network Management feature*) NNMi provides the **Connection State** attribute to help you track the health of communication connections between Global Managers and Regional Managers in your network environment. The table below describes each possible Connection State value.

To verify the state of the communication connection between NNMi management servers:

1. Open the NNMi console on the Global Manager (NNMi management server).
2. Navigate to the **Global Network Management** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Global Network Management** form.
3. Select the **Regional Management Servers** tab.
4. Locate the **Connection State** column in this view.
5. Check the Connection State value for each Regional NNMi management server.

   Tip: To verify the list of Nodes being managed by each NNMi management server, see View the NNMi Management Servers' Domain List.

Possible States for Connections to Regional NNMi Management Servers

Connection State	Description
Not Established	The connection configuration was recently saved, and NNMi is attempting to establish the connection.
Partial Connection	The connection state is transitioning between states due to a recent change in your network environment or a change in NNMi configuration settings.
Connected	Communication between the two NNMi management servers is working properly.

Connection State	Description
Not Connected	An error occurred and the connection failed. Check the Regional Management Server configuration settings. Perhaps one of the designated port numbers is not correct? See "Global Manager: Connect to a Regional Manager" (on page 55). Perhaps the Regional NNMi management server is currently down? See "Troubleshoot Global Network Management" (on page 61).

Check the Health of Global Managers and Regional Managers

Do one of the following to check the health of the Global Network Management feature:

- Log onto the Global Manager as an NNMi administrator, and open the NNMi console on the Global Manager (NNMi management server). Click here for more information.

  Do one of the following:

  - Click the **Tools** → **NNMi System Health Report** menu option to view detailed information about the health of Global Network Management.
  - Click the **Help** → **System Information**.

    i. Click the **Global Network Management** tab.

    ii. In the **Regional Managers Reporting to this Global Manager** section, review the list of all Regional Managers that report to this Global Manager:

    - **Name**: The current value of the Name attribute for this Regional NNMi management server (as specified in the Remote Manager Connection form).
    - **Connection State**: The current state of communication between the Global Manager and Regional Manager. There are four possible values:
      - **Not Established** — A new Regional Manager Connection is not yet fully functional. This state is brief unless NNMi encounters a problem.
      - **Connected** — Data is flowing between the Global Manager and the Remote Manager.
      - **Not connected** — A previously established connection is no longer working. See "Clock Synchronization Issues (SSO / Global Network Management)" (on page 61).

        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, the UNIX (HP-UX / Linux / Solaris) tool Network Time Protocol Daemon (NTPD) or one of the available Windows operating system tools.

        Review the Global Network Management deployment choices in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

- **Node Count**: The number of nodes in the Global Manager's database that are being managed by this Regional Manager.

- Log onto the Regional Manager as an NNMi administrator, and open the NNMi console on the Regional Manager (NNMi management server).click here for more information..

  Do one of the following:

  - Click the **Tools → NNMi System Health Report** menu option to view detailed information about the health of Global Network Management.
  - Click the **Help → System Information**.

    i. Click the **Global Network Management** tab.

    ii. Scroll down to the **Reporting to Global Managers** section, and review the list of all Global Managers that receive data from this Regional Manager:

    - **Name**: The fully-qualified DNS hostname of the Global Manager (NNMi management server).

      Note: If you see something other than a fully-qualified DNS hostname in the Name column, the Global Manager is down and has been down since this Regional Manager was last restarted (see "Stop or Start an NNMi Process" (on page 45) or "Stop or Start NNMi Services" (on page 49) for more information).

    - **Messages Currently in Queue**: The current number of messages that need to be sent to the Global Manager.

      Messages are automatically sent to the Global Manager. If the number of messages in the queue is continually increasing and never decreasing or if the number is consistently over 10,000 then there may be a problem.

      Note: If the Global Manager is down for maintenance for a few hours, the queue size naturally increases until the Global Manager is back online.

      Queue size over 100,000 indicates a serious issue. Consider disconnecting that global manager and removing the subscription until the issue can be resolved..

- NNMi administrators can use the command line on any NNMi management server to generate a report about NNMi health. See the nnmhealth.ovpl Reference Page for more information.

There are two ways to log onto a Regional Manager:

- Directly log onto the Regional Manager (NNMi management server).
- From the Global Manager, select any Node being managed by the Regional Manager and click **Actions → Regional Manager Console**. See "Actions Provided by NNMi" (on page 30).

Error Messages About Regional Managers (*NNMi Advanced*)

(*NNMi Advanced - Global Network Management feature*) A special set of incidents keeps the Global Manager informed of any problems with the Regional Manager:

- Licensing issues
  - License Expired
  - License Mismatch
  - License Node Count Exceeded
- Application fail-over health issues

  - Nnm Cluster Failover
  - Nnm Cluster Lost Standby
  - Nnm Cluster Startup
  - Nnm Cluster Transfer
- Traffic volume issues
  - Snmp Trap Limit Critical
  - Snmp Trap Limit Major
  - Snmp Trap Limit Warning
  - Trap Storm

These incidents are generated on the Regional Manager (NNMi management server). The Regional Manager forwards a copy of these incidents to the Global Manager. NNMi dynamically closes these incidents on the Regional Manager when the issue is resolved. The NNMi administrator for the Global Manager (NNMi management server) must manually close the forwarded copy.

From any Incident view, you can identify the forwarding server or servers (`cia.remotemgr`). Use the Custom Incident Attribute tab on the Incident form for the selected incident. NNMi uses Custom Incident Attributes (CIAs) to attach additional information to incidents.

Configuring Communication Protocol

NNMi uses the following protocols to discover your network and monitor the health of your network environment:

- Simple Network Management Protocol (SNMPv1 and SNMPv2c)
  - Read-only queries, also known as "Get" commands.

    SNMPv1 and SNMPv2c require the use of a read community string to authenticate messages that are sent between NNMi and SNMP agents. NNMi cannot discover information about the SNMPv1 and SNMPv2c devices in your network environment until you provide the appropriate read community strings. During discovery and monitoring, NNMi uses the read community strings you provide in the Communication Configuration workspace. When a device is first discovered, NNMi tries all appropriate read community strings and makes a record of the first read community string that works. To keep network traffic to a minimum, from then on NNMi uses the recorded read community string when communicating with that device using SNMP. If at some point the device no longer responds to the recorded read community string, NNMi tries all appropriate read community strings and makes a record of the first read community string that now works.
  - Write commands, also known as "Set" commands.

    SNMPv1 and SNMPv2c require the use of a write community string to authenticate messages that are sent between the nnmsnmpset.ovpl command and SNMP agents. To keep network traffic to a minimum, NNMi tries each of the three appropriate write community strings in this order: Specific Node, Region, Default. NNMi makes a record of the first write community string that now works.
- SNMPv3 requires the use of user-based security model (USM) user names instead of *SNMPv1/SNMPv2c community strings* to authenticate messages that are sent between NNMi and SNMP agents. NNMi cannot discover information about the SNMPv3 devices in your network environment until you provide the appropriate user name and authentication. During discovery and monitoring, NNMi uses the *SNMPv3 User Name* attribute value and authentication that the NNMi administrator provides in the Communication Configuration workspace. When a device is first discovered, NNMi tries all appropriate USM user names and makes a record of the first USM user name that works. To keep network traffic to a minimum, from then on NNMi uses the recorded *SNMPv3 User Name* attribute value when communicating with that device using SNMP. If at some point the device no longer responds to the recorded *SNMPv3 User Name* attribute value, NNMi tries all appropriate USM user names and makes a record of the one that now works.
- Internet Control Message Protocol (ICMP) ping commands

Note: If NNMi discovers a device for which no SNMP authentication was provided in the Communication Configuration workspace, that device is treated as a non-SNMP device.

You control the amount of traffic NNMi generates on your network. You can modify the settings to meet your needs.

To configure the way NNMi uses ICMP and SNMP protocols, do the following:

1. Navigate to the **Communication Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Communication Configuration**.
2. Make your configuration choices. Click here for a list of choices .
3. Click **Save and Close** to apply your changes.

Note: You control the amount of network traffic generated by NNMi by designating the **Rediscovery Interval** setting (see "Adjust the Rediscovery Interval" (on page 138) for more information) and making choices when you "Configure Monitoring Behavior" (on page 216).

Configure Default SNMP and ICMP Protocol Settings

NNMi generates network traffic using ICMP and SNMP protocols to discover and monitor your network environment. Default settings for the use of these protocols are provided, for example timeout and retry behavior settings.

To configure the default communication protocol settings for your environment:

1. Navigate to the **Communication Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Communication Configuration**.
2. Locate the **Default Settings** groups.
3. Make your configuration choices (see the Default SNMP Settings table and Default ICMP Settings table).

   For an explanation of how NNMi implements timeout and retry configurations, see "Timeout / Retry Behavior Example for SNMP" (on page 72) and "Timeout / Retry Behavior Example for ICMP" (on page 73).
4. Click **Save and Close** to apply your changes.

Default SNMP Settings Attributes

Attribute	Description
Enable SNMP Address Rediscovery	**Note**: The NNMi administrator can over-ride this setting for a Region or on a per-node basis. See "Communication Region Form" (on page 80)and "Specific Node Settings Form (Communication Settings)" (on page 96). If ☑ enabled, NNMi automatically identifies which management address (SNMP agent) to use for each device. If the initially configured address becomes unreachable, NNMi automatically locates another SNMP agent, if possible, and changes the management address attribute value. Click here for more information. When NNMi first discovers a node, the *seed address* (provided by the NNMi administrator) or discovered address (for non-seeded nodes) becomes the initial Management Address of the node. After NNMi builds an inventory of all IP addresses

Attribute	Description
	associated with the node (see "How Spiral Discovery Works" (on page 114)), NNMi follows a set of rules to determine which address is the best choice for each node's Management Address: **Note**: With NNMi Advanced, the NNMi administrator specifies whether NNMi prefers IPv4 or IPv6 addresses when selecting the Management Address. See the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals` 1. NNMi ignores the following addresses when determining which Management Address is most appropriate: ▪ Any address of an administratively-down interface. ▪ Any address that is virtual (HSRP/VRRP). ▪ Any IPv4 **Anycast Rendezvous Point IP Address**[1] or IPv6 Anycast address. ▪ Any address in the reserved loopback network range. IPv4 uses 127/24 (`127.*.*.*`) and IPv6 uses `::1`. ▪ Any IPv6 **link-local address**[2]. 2. NNMi prefers loopback interfaces for management communication. ▪ If a node has only one **loopback address**[3], and the associated SNMP agent responds to NNMi, that address becomes the Management Address. ▪ If a node supports multiple loopback addresses, NNMi queries each loopback addresses, starting with the lowest number. NNMi uses the loopback address with the lowest number from which the SNMP agent responds (for example, 10.16.42.197 is a lower number than 10.16.197.42). 3. If no loopback address responds, NNMi queries the last-known Management Address (if any). 4. If no response, NNMi queries up to 10 of any remaining IP addresses in the node's IP address inventory, starting with the lowest number. NNMi uses the address with the lowest number from which the SNMP agent responds. 5. If no response, NNMi might be configured to repeat the sequence using SNMPv1, SNMPv2c, or SNMPv3 in the order specified by the NNMi administrator (Communication Configurations *SNMP Minimum Security Level* settings). 6. When all else fails, NNMi retains the last known Management Address (if any) and automatically changes the State of that SNMP Agent object to Critical.

[1]Rendezvous Point addresses are loopback addresses used for routers in multi-cast network configurations.
[2]A non-routable IPv6 unicast address only used for communication with other nodes on the same link (LAN or VLAN). Link local addresses cannot be used for communication that must be forwarded through a router. IPv6 auto-configuration automatically assigns a unique link local address in the fe80::/10 address space to each IPv6-enabled interface on a system.
[3]The address associated with the loopback interface. The loopback interface is a virtual interface on a device that provides a route for internal communication. Many vendors provide a specially configured loopback for management purposes. Exact details of how loopbacks are configured varies by vendor and model. See each device's documentation for details. NNMi identifies these loopback addresses by using IfType 24, softwareloopback from the IANA ifType-MIB.

Attribute	Description
	This process is repeated during each Auto-Discovery cycle, and the Management Address can change. For example, NNMi's inventory of addresses for the node expands, or the current Management Address does not respond to SNMP queries due to network problems or node reconfiguration. The NNMi administrator can prevent changes to the management address using the Communication Configurations *Enable SNMP Address Rediscovery* or *Preferred Management Address* setting. If ☐ disabled, when the current management address (SNMP agent) becomes unreachable, NNMi reclassifies the node as a non-SNMP node until the previously configured management address is available again.

Attribute	Description
Enable SNMP GetBulk	*Applies only to SNMPv2 or higher.* Some devices do not respond well to multiple SNMP OIDs requests at one time. If you have devices in your network environment that have trouble responding to `GetBulk` commands, you can instruct NNMi to use `Get` or `GetNext` instead of `GetBulk`. If ☑ enabled, NNMi uses the SNMPv2 `GetBulk` command to gather information from devices in your network environment. If ☐ disabled, NNMi uses the SNMP `Get` or `GetNext` command to gather information from devices in your network environment (requesting responses for one SNMP OID at a time).
SNMP Timeout	(Seconds:Milliseconds) Maximum 1 millisecond less than a minute: 59 seconds 999 milliseconds. Time that NNMi waits for a response to an SNMP query before reissuing the request. Both the Discovery Process and the State Poller Service use this setting. For an explanation of how NNMi implements timeout and retry configurations, see "Timeout / Retry Behavior Example for SNMP" (on page 72).
SNMP Retries Count	Maximum number of retries that NNMi issues for an SNMP query before determining the query result to be "unresponsive". Zero means no retries. Both the Discovery Process and the State Poller Service use this setting.
SNMP Port	Default is 161. Specifies the NNMi management server's port that NNMi uses when generating SNMP traffic. Both the Discovery Process and the State Poller Service use this setting.
SNMP Proxy Address	*Optional*. IP address of the your SNMP Proxy Server (for example, a proxy that gathers data from non-SNMP devices and can use that data to respond to NNMi SNMP requests). To enable a proxy, you must also provide the port number of your SNMP Proxy Server. See SNMP Proxy Port (next attribute).
SNMP Proxy Port	*Optional*. Port number of the SNMP Proxy Server. To enable a proxy, you must also provide the IP address of your SNMP Proxy Server. See SNMP Proxy Address (previous attribute).

Attribute	Description
SNMP Minimum Security Level	For SNMPv1 or SNMPv2c, configure NNMi to use Community Strings in your network environment: • Community Only (SNMPv1) NNMi tries only SNMPv1 settings. • Community Only (SNMPv1 or v2c) NNMi first tries to use SNMPv2c settings, and, if that fails, NNMi tries SNMPv1 settings. • Community NNMi first tries to use SNMPv2c settings, and, if that fails, NNMi tries SNMPv1 settings. If both SNMPv2c and SNMPv1 fail, NNMi tries SNMPv3 settings if any are available. For SNMPv3, configure NNMi to use the User-based Security Module (USM) level of security required in your network environment (if your environment also uses SNMPv1/SNMPv2c, select Community): • No Authentication, No Privacy • Authentication, No Privacy • Authentication, Privacy See "Timeout / Retry Behavior Example for SNMP" (on page 72) for an explanation of NNMi behavior with each of these choices.

Note: NNMi needs to know which SNMPv1 or SNMPv2c community strings (read/write) are used in your environment (see "Configure Default Community Strings (SNMPv1 or SNMPv2c)" (on page 73)) and which SNMPv3 USM settings are used in your environment (see "Configure Default SNMPv3 Settings" (on page 76)).

Default ICMP Settings

Attribute	Description
ICMP Timeout	(Seconds:Milliseconds) Maximum 1 millisecond less than a minute: 59 seconds 999 milliseconds. Time that NNMi waits for a response to an ICMP query before reissuing the request. For an explanation of how NNMi implements timeout and retry configurations, see "Timeout / Retry Behavior Example for ICMP" (on page 73).
ICMP Retries Count	Maximum number of retries that NNMi issues for an ICMP query before logging an error. Zero means no retries.

Related Topics:

"Configure Default Community Strings (SNMPv1 or SNMPv2c)" (on page 73)

"Configure Regions (Communication Settings)" (on page 79)

"Configure Specific Nodes (Communication Settings)" (on page 95).

Timeout / Retry Behavior Example for SNMP

When NNMi attempts to contact a device, your configuration settings for Timeout and Retry influence NNMi behavior.

NNMi attempts to obtain information about a hostname/IP-address using SNMP, then waits the configured timeout interval for a response. If not successful, NNMi increments the timeout interval before trying again. This process repeats until one of the following is true:

- The device responds to SNMP.
- The maximum configured number of SNMP Retries fails. For example, if your timeout is 2 seconds and your retry is 4:
  - NNMi attempts to communicate with a device and waits 2 seconds for a response.
  - If unsuccessful, NNMi tries again and waits 4 seconds for a response.
  - If unsuccessful, NNMi tries again and waits 6 seconds for a response.
  - If unsuccessful, NNMi tries again and waits 8 seconds for a response.

  If no response, NNMi repeats this process using the next configured SNMP level.
- NNMi exhausts all possibilities. NNMi considers the hostname/IP-address to be a *non-SNMP* device until the next Discovery or Monitoring cycle.

Tip: It is best to use the same timeout/retry numbers for both ICMP and SNMP.

Your choice of SNMP Minimum Security Level determines the range of possibilities:

- If your SNMP Minimum Security Level is ***Community Only (SNMPv1)***, NNMi uses only SNMPv1 to locate SNMP agents.
- If your SNMP Minimum Security Level is ***Community Only (SNMPv1 or v2c)***, NNMi cycles through the following until successful:

  SNMPv2c

  SNMPv1
- If your SNMP Minimum Security Level is ***Community***, NNMi cycles through the following until successful:

  SNMPv2c

  SNMPv1

  SNMPv3 *No Authentication, No Privacy* settings (if any matching configurations, otherwise skip).

  SNMPv3 *Authentication, No Privacy* settings (if any matching configurations, otherwise skip).

  SNMPv3 *Authentication, Privacy* settings (if any matching configurations).
- If your SNMP Minimum Security Level is ***No Authentication, No Privacy***, NNMi cycles through the following until successful:

  SNMPv3 *No Authentication, No Privacy* settings (if any matching configurations at this, otherwise skip)

  SNMPv3 *Authentication, No Privacy* settings (if any matching configurations, otherwise skip).

  SNMPv3 *Authentication, Privacy* settings (if any matching configurations).
- If your SNMP Minimum Security Level is ***Authentication, No Privacy***, NNMi cycles through the

following until successful:

SNMPv3 *Authentication, No Privacy* settings (if any matching configurations, otherwise skip).

SNMPv3 *Authentication, Privacy* settings (if any matching configurations).

- If your SNMP Minimum Security Level is **Authentication, Privacy**, NNMi cycles through the following until successful:

  SNMPv3 *Authentication, Privacy* settings (if any matching configurations).

Timeout / Retry Behavior Example for ICMP

When NNMi attempts to contact a device, your configuration settings for Timeout and Retry influence NNMi behavior.

NNMi attempts to contact the device using ICMP, then waits the configured timeout interval for a response. If not successful, NNMi increments the timeout interval before trying again. This process repeats until one of the following is true:

- The device responds to ICMP.
- The maximum configured number of ICMP Retries fails. NNMi considers the device unreachable through ICMP until the next Discovery or Monitoring cycle. For example, if your timeout is 2 seconds and your retry is 4:
  - NNMi attempts to communicate with a device and waits 2 seconds for a response.
  - If unsuccessful, NNMi tries again and waits 4 seconds for a response.
  - If unsuccessful, NNMi tries again and waits 6 seconds for a response.
  - If unsuccessful, NNMi tries again and waits 8 seconds for a response.

Tip: It is best to use the same timeout/retry numbers for both ICMP and SNMP.

Configure Default Community Strings (SNMPv1 or SNMPv2c)

Use the Default Community Strings tab to provide default SNMPv1 and SNMPv2c passwords. For each address, NNMi checks the communication configuration settings in this order: communication protocols for Specific Nodes, communication protocols for Network Regions, and if no match is found, NNMi tries these default community strings. If NNMi discovers a device for which no SNMP settings are provided, that device is treated as a Non-SNMP device.

During initial discovery, NNMi tries many community strings until a match is found. After a match is identified for a node, the information is recorded to prevent future authentication errors.

Note: If you provide a read community string for a specific device, NNMi honors your choice and does not try any Region or Default community strings for that device.

NNMi uses SNMP read-only queries (Get commands) for ongoing discovery and monitoring of your network environment. SNMP read community strings are the validation passwords used to authenticate messages sent from NNMi to an SNMP agent. NNMi uses SNMP to gather useful information about the devices in your network environment. After receiving an SNMP request, an SNMP agent compares the read community string in the request to the read community strings that are configured for that SNMP agent. The SNMP agent responds to the request only when the request is accompanied by a valid community string.

During NNMi installation, any community strings that were provided are automatically stored in the table on the Default Community Strings tab.

Provide any number of additional community strings that are used broadly in your environment (for example, by default). The order in which your read community string settings appear in the table does not matter. NNMi checks all Default read community strings in parallel.

Tip: Having a large number of default community strings can negatively impact discovery performance. Instead of entering many default community strings, consider fine tuning the community string configuration for particular areas of your network by using the Regions or Specific Nodes settings.

NNMi uses the SNMPv2 settings to discover the SNMPv2 information about your network. This also determines whether NNMi *receives or discards incoming* SNMPv2 traps. Click here for more information.

- If the *incoming* trap's Source Node object or Source Object (such as card or interface) has not yet been discovered by NNMi, see "Handle Unresolved Incoming Traps" (on page 432) for additional information. See also "Configure Network Devices to Send SNMP Notifications to NNMi" (on page 428).
- If the Source Node or Source Object of the *incoming* trap has been discovered by NNMi using SNMPv2c or SNMPv1, NNMi accepts *incoming* traps from SNMPv2c or SNMPv1, but NNMi discards *incoming* traps from SNMPv3.
- NNMi discards traps that have no Incident Configuration or with an Incident Configuration set to Disabled.
- If either the Source Node or Source Object has *Management Mode* set to **Not Managed** or **Out of Service** in the NNMi database, NNMi always discards the incoming trap. See "Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " (on page 257).

  NNMi provides the Management Mode workspace so that you can quickly view lists of all nodes, interfaces, cards, addresses, or node components that NNMi is not currently discovering or monitoring. For information about these views:

Note: If you want the NNMi management server to *forward* SNMPv2 traps to other machines in your network environment, see "Configuring Trap Forwarding" (on page 297) for additional configuration steps.

To configure default SNMPv1 or SNMPv2c community strings for your environment:

1. Navigate to the **Communication Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Communication Configuration**.
2. Locate the **Default SNMPv1/v2 Community Strings** tab.
3. To provide a default *read community string*, navigate to the **Read Community Strings** table and do one of the following:
   - To establish a community string setting, click the New icon. In the Default Read Community String form, provide the required information (see table).
   - To edit a community string setting, click the Open icon in the row representing the community string setting you want to edit. In the Default Read Community String form, provide the required information (see table).
   - To delete a community string setting, select a row and click the Delete icon.
4. To provide a default *write community string*, navigate to **the Write Community String** attribute (see table).

5. Click **Save and Close** to return to the Communication Configuration form.
6. Click **Save and Close** to apply your changes.

Default SNMPv1 or SNMPv2c Community Strings

Attribute	Description
Read Community String	The SNMPv1 or SNMPv2c "get" (read-only) community string that is used for this region (case-sensitive). Many proxy vendors use the *read community string* for specifying remote target information. NNMi supports substitution parameters within read community strings for SNMPv1 or SNMPv2 proxy environments. Click here for more information. Copy and paste these codes at the end of your read community string to provide the values required by your proxy environment. NNMi substitutes the actual attribute values from the NNMi database at runtime: ${contextName} = Used for specifying VLAN context for switches (VLAN associated with the remote target node) ${managementAddress} = Node form, Management Address attribute value (the remote target node) ${snmpPort} = SNMP Agent form, UDP Port attribute value (SNMP agent associated with the remote target node)
Write Community String	The SNMPv1 or SNMPv2c "set" (write) community string that is used for this region (case-sensitive). *Optional*. For use with the nnmsnmpset.ovpl command line tool. SNMPv1 and SNMPv2c require that you know the SNMP agent's *write community string* before you can change settings on any device. The nnmsnmpset.ovpl command can use the value you provide here, rather than requiring that you type the write community string each time you invoke the command.

Default Read Community String Form

For each IP address, NNMi checks the communication configuration settings in this order: communication protocols for Specific Devices, communication protocols for Network Regions, and if no match is found, NNMi tries the default community strings. If NNMi discovers a device for which no community string is provided, that device is treated as a Non-SNMP device.

To provide a default community string for your environment:

1. Navigate to the **Default Read Community String** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select the **Communication Configuration**.
    c. Navigate to the **Default SNMPv1/v2 Community Strings** tab.
    d. Navigate to the **Read Community Strings** table.
    e. Do one of the following:

- To establish a community string setting, click the New icon, and continue.
- To edit a community string setting, select a row, click the Open icon in the row representing the configuration you want to edit, and continue.

2. Provide the read community string (see table).

   Provide any number of additional SNMPv1 or SNMPv2c read community strings that are used broadly in your environment (for example, by default). The order in which your read community string settings appear in this table does not matter. NNMi checks all Default read community strings in parallel.

3. Click either:
   - **Save and Close** to return to the Communication Configuration form.
   - Save and New to add another community string.
4. Click **Save and Close** to apply your changes.

Default Read Community String

Attribute	Description
Read Community String	The SNMP "get" (read-only) community string that is used in your network environment (case-sensitive). Many proxy vendors use the *read community string* for specifying remote target information. NNMi supports substitution parameters within read community strings for SNMPv1 or SNMPv2 proxy environments. Click here for more information. Copy and paste these codes at the end of your read community string to provide the values required by your proxy environment. NNMi substitutes the actual attribute values from the NNMi database at runtime: ${contextName} = Used for specifying VLAN context for switches (VLAN associated with the remote target node) ${managementAddress} = Node form, Management Address attribute value (the remote target node) ${snmpPort} = SNMP Agent form, UDP Port attribute value (SNMP agent associated with the remote target node)

Configure Default SNMPv3 Settings

Use the Default SNMPv3 Settings tab to provide default SNMPv3 user-based security model (USM) settings. For each address, NNMi checks the communication configuration settings in this order: communication protocols for Specific Nodes, communication protocols for Network Regions, and if no match is found, NNMi tries these default user-based security model (USM) settings. If NNMi discovers a device for which no SNMP settings are provided, that device is treated as a Non-SNMP device.

During initial discovery, NNMi tries many SNMP configuration settings until a match is found. After a match is identified for a Node, the information is recorded to prevent future authentication errors.

Note: If you provide SNMPv3 user-based security model (USM) settings for a specific device, NNMi honors your choice and does not try any Region or Default settings for that device.

NNMi uses SNMP queries for ongoing discovery and monitoring of your network environment. SNMPv3 user-based security model (USM) settings are used to authenticate messages sent from NNMi to an SNMP agent. NNMi uses SNMP to gather useful information about the devices in your network environment. After receiving an SNMP request, an SNMP agent compares the SNMPv3 user-based security model (USM) settings in the request to the SNMPv3 user-based security model (USM) settings that are configured for that SNMP agent. The SNMP agent responds to the request only when the request is accompanied by valid SNMPv3 user-based security model (USM) settings.

Provide any number of additional SNMPv3 user-based security model (USM) settings that are used broadly in your environment (for example, by default). The order in which your SNMPv3 user-based security model (USM) settings appear in this table does not matter. NNMi checks all Default SNMPv3 Settings at a particular security level in parallel.

NNMi uses Default SNMPv3 user-based security model (USM) settings to access devices.

To view the current list of default SNMPv3 USM settings:

1. Navigate to the **Default SNMPv3 Settings** tab.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Communication Configuration**.
    c. Navigate to the **Default SNMPv3 Settings** tab.
2. The displayed table lists the Unique Name of each default SNMPv3 USM setting.

    Note: NNMi tries to use the Specific Node SNMPv3 Settings. If none match, NNMi tries the Region SNMPv3 Settings. If none match, NNMi tries the default SMNPv3 settings provided here.
3. You can do the following:
    - To establish a new setting, click the New icon. See "Default SNMPv3 Settings form" (on page 77).

      Click **Save and Close** to return to the Default SNMPv3 Settings form.
    - To edit an existing setting, select a row, click the Open icon. See "Default SNMPv3 Settings form" (on page 77).

      Click **Save and Close** to return to the Default SNMPv3 Settings form.
    - To delete an existing setting from the Default list, select a row and click the Delete icon.

      Note: The record remains in the database for possible use elsewhere and is simply removed from the Default list.
4. Click **Save and Close** to return to the Communication Configuration form.
5. Click **Save and Close** to apply your changes.

Default SNMPv3 Settings form

NNMi can use SNMPv3 user-based security model (USM) settings to access devices.

NNMi tries to use the current SNMPv3 Settings attribute value from Specific Node Settings. If none match, NNMi tries the Region SNMPv3 Settings. If none match, NNMi tries the default SMNPv3 settings provided here.

To configure a Default SNMPv3 Setting:

1. Navigate to the **Default SNMPv3 Settings** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select the **Communication Configuration**.
    c. Navigate to the **Default SNMPv3 Settings** tab.
    d. Do one of the following:
        - To create default SNMPv3 Setting definition, click the New icon.
        - To edit a default SNMPv3 Setting, select a row, click the Open icon.
2. Click the SNMPv3 Settings Lookup icon and select one of the options from the drop-down menu:
    - Quick View to display summary information for the currently configured (selected) SNMPv3 Setting name.
    - Quick Find to view and select from the list of all existing SNMPv3 Settings (for more information see "Use the Quick Find Window" (on page 28)).
    - Open to display the details of the currently configured (selected) SNMPv3 Setting (see "SNMPv3 Settings Form" for more information).
    - New to create a new SNMPv3 Setting (see "SNMPv3 Settings Form" for more information).
3. Click **Save and Close** to return to the Default SNMPv3 Settings form.
4. Click **Save and Close** to return to the Communication Configuration form.
5. Click **Save and Close** to apply your changes.

Configure the Default Device Credentials (*NNM iSPI NET*)

HP Network Node Manager iSPI Network Engineering Toolset Software uses the Default Credentials setting to access devices when running Diagnostics either automatically or when the **Actions → Run Diagnostics (iSPI NET only)** option is used. (See "Configure Diagnostics for an Incident (NNM iSPI NET)" (on page 421) and Node Form: Diagnostics Tab for more information.)

NNM iSPI NET uses Secure Shell (SSH) to establish a secure connection with devices in your network environment, using the credentials provided here. If the SSH attempt fails, If the SSH attempt fails, NNMi uses Telnet protocol as the communication method.

To provide the default credentials setting:

1. Navigate to the **Default Device Credentials** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Communication Configuration**.
    c. Navigate to the **Default Device Credentials** tab.
    d. Do one of the following:
        - To establish a credential setting, click the New icon, and continue.
        - To edit a credential setting, select a row, click the Open icon, and continue.

- To delete a credential setting, select a row and click the Delete icon

2. Provide *one* setting for the default attribute values (see table).

   Caution: Populate only one row in this table.

   Note: NNMi tries to use the Specific Node Device Credentials. If none match, NNMi tries the Region Device Credentials. If none match, NNMi tries the default credential settings provided here.

3. Click **Save and Close** to return to the Communication Configuration form.
4. Click **Save and Close** to apply your changes.

Default Device Credential Attributes

Attribute	Description
User Name	Type the user name that you want NNMi to use for logging into devices by default (when no Region or Specific Node settings work).
Password	Type the password that you want NNMi to use for logging into devices by default (when no Region or Specific Node settings work). **Note**: NNMi encrypts the password and displays asterisks for this attribute. If you want to change the password, first clear the asterisks displayed in the Password attribute and enter the new Password value.

Configure Regions (Communication Settings)

Configuring communication protocols for regions is optional.

Note: If you provide an SNMPv1 or SNMPv2c *read community string* or an SNMPv3 USM Setting for a specific device, NNMi honors your choice and does not try any Region or Default settings for that device.

Use the Regions tab to fine tune communication protocol usage and settings for particular regions of your network (for example, buildings, floors within those buildings, workgroups within a particular floor, or **private IP addresses**[1]). When you leave a field blank in a region definition, NNMi uses the next applicable configuration setting in the following order:

- The value for each field as defined in the first Region definition that matches, Regions are checked according to the Ordering number. The match with the lowest Ordering number applies.
- If no Region definition provides a value for an attribute, the default value is used.

Note: NNMi enables you to set up one or more SNMP Proxy Servers when an SNMP node is otherwise unreachable (for example, when a node you want to manage is behind a firewall). To enable NNMi to use the SNMP Proxy Server, when you configure communication protocols for network regions, you must include the IP address and port number on the SNMP Proxy Server. See "Communication Region Form" (on page 80) for more information.

If your communication protocol usage is too complex for Region definitions, see "Configure Specific Nodes (Communication Settings)" (on page 95).

[1]These are IPv4 addresses that can be reused in home and office local area networks (LANs). Following the standards set by RFC 1918 and RFC 4193 (10.*.*.*, 169.254.*.*, 172.16-31.*.*, and 192.168.*.*)

To configure communication protocols for a particular region of your network:

1. Navigate to the **Communication Region** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select the **Communication Configuration**.
    c. Navigate to the **Regions** tab.
    d. Do one of the following:
        - To establish a region definition, click the New icon, and continue.
        - To edit a region definition, select a row, click the Open icon, and continue.
        - To delete a region definition, select a row and click the Delete icon.
2. Provide the required information. Define the regions with wildcard address, wildcard device names, or literal addresses and names . See "Communication Region Form" (on page 80).
3. Click **Save and Close** to return to the Communication Configuration form.
4. Click **Save and Close** to apply your changes.

Related Topics:

"Configure Default SNMP and ICMP Protocol Settings" (on page 67)

"Configure Default Community Strings (SNMPv1 or SNMPv2c)" (on page 73)

"Configure Specific Nodes (Communication Settings)" (on page 95)

Communication Region Form

To configure communication regions:

1. Navigate to the **Communication Region** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select the **Communication Configuration**.
    c. Navigate to the **Regions** tab.
    d. Do one of the following:
        - To establish a region definition, click the New icon, and continue.
        - To edit a region definition, select a row, click the Open icon, and continue.
2. Provide the basic communication region definition (see the Regional Basic Settings table, Regional SNMP Settings table, and Regional ICMP Settings table).
3. Make your configuration choices. Click here for a list of choices .
4. Click **Save and Close** to return to the Communication Configuration form.
5. Click **Save and Close** to apply your changes.

Regional Basic Settings

Attribute	Description
Name	A name for this region.
Ordering	A numeric value. NNMi checks for configuration settings in the order you define (lowest number first). NNMi uses the first match found for each address. No duplicate Ordering numbers are allowed. Each Communication Region ordering number must be unique. **Tip**: It is recommended that ordering numbers are incremented by 10s or 100s to provide flexibility when adding new regions over time.
Description	*Optional*. Provide any description that would be useful for communication purposes within your team. Type a maximum of 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Regional SNMP Settings

Attribute	Description
Enable SNMP Communication	If ☑ enabled, the Discovery Process and State Poller Service generate network traffic with SNMP protocol to discover and monitor your network devices in this region. If ☐ disabled, NNMi does not generate any SNMP traffic on your network in this region. **Caution**: At least one IP Address in each node must have SNMP enabled, otherwise no SNMP data is collected from that Node. With no SNMP data, Spiral Discovery interprets each IP Address as a separate node, Causal Engine calculates Status based only on IP address State, previously discovered Interfaces show a State attribute value of "Not Polled" and a Status attribute value of "No Status" with the Interface map-symbol color set to beige, and no new Interfaces are discovered. **Note**: See "Monitoring Network Health" (on page 215) for information about enabling/disabling SNMP communication specifically for the State Poller Service.
Enable SNMP Address Rediscovery	**Note**: The NNMi administrator can over-ride this setting on a per-node basis. See "Specific Node Settings Form (Communication Settings)" (on page 96).

Attribute	Description
	If ☑ enabled, NNMi automatically identifies which management address (SNMP agent) to use for each device. If the initially configured address becomes unreachable, NNMi automatically locates another SNMP agent, if possible, and changes the management address attribute value. Click here for more information. When NNMi first discovers a node, the *seed address* (provided by the NNMi administrator) or discovered address (for non-seeded nodes) becomes the initial Management Address of the node. After NNMi builds an inventory of all IP addresses associated with the node (see "How Spiral Discovery Works" (on page 114)), NNMi follows a set of rules to determine which address is the best choice for each node's Management Address: **Note**: With NNMi Advanced, the NNMi administrator specifies whether NNMi prefers IPv4 or IPv6 addresses when selecting the Management Address. See the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals` 1. NNMi ignores the following addresses when determining which Management Address is most appropriate: ■ Any address of an administratively-down interface. ■ Any address that is virtual (HSRP/VRRP). ■ Any IPv4 **Anycast Rendezvous Point IP Address**[1] or IPv6 Anycast address. ■ Any address in the reserved loopback network range. IPv4 uses 127/24 (`127.*.*.*`) and IPv6 uses `::1`. ■ Any IPv6 **link-local address**[2]. 2. NNMi prefers loopback interfaces for management communication. ■ If a node has only one **loopback address**[3], and the associated SNMP agent responds to NNMi, that address becomes the Management Address. ■ If a node supports multiple loopback addresses, NNMi queries each loopback addresses, starting with the lowest number. NNMi uses the loopback address with the lowest number from which the SNMP agent responds (for example, 10.16.42.197 is a lower number than 10.16.197.42). 3. If no loopback address responds, NNMi queries the last-known Management Address (if any). 4. If no response, NNMi queries up to 10 of any remaining IP addresses in the

[1]Rendezvous Point addresses are loopback addresses used for routers in multi-cast network configurations.
[2]A non-routable IPv6 unicast address only used for communication with other nodes on the same link (LAN or VLAN). Link local addresses cannot be used for communication that must be forwarded through a router. IPv6 auto-configuration automatically assigns a unique link local address in the fe80::/10 address space to each IPv6-enabled interface on a system.
[3]The address associated with the loopback interface. The loopback interface is a virtual interface on a device that provides a route for internal communication. Many vendors provide a specially configured loopback for management purposes. Exact details of how loopbacks are configured varies by vendor and model. See each device's documentation for details. NNMi identifies these loopback addresses by using IfType 24, softwareloopback from the IANA ifType-MIB.

Attribute	Description
	node's IP address inventory, starting with the lowest number. NNMi uses the address with the lowest number from which the SNMP agent responds. 5. If no response, NNMi might be configured to repeat the sequence using SNMPv1, SNMPv2c, or SNMPv3 in the order specified by the NNMi administrator (Communication Configurations *SNMP Minimum Security Level* settings). 6. When all else fails, NNMi retains the last known Management Address (if any) and automatically changes the State of that SNMP Agent object to Critical. This process is repeated during each Auto-Discovery cycle, and the Management Address can change. For example, NNMi's inventory of addresses for the node expands, or the current Management Address does not respond to SNMP queries due to network problems or node reconfiguration. The NNMi administrator can prevent changes to the management address using the Communication Configurations *Enable SNMP Address Rediscovery* or *Preferred Management Address* setting. If ☐ disabled, when the current management address (SNMP agent) becomes unreachable, NNMi reclassifies the node as a non-SNMP node until the previously configured management address is available again.

Attribute	Description
Enable SNMP GetBulk	*Applies only to SNMPv2 or higher.* Some devices do not respond well to multiple SNMP OIDs requests at one time. If you have devices in your network environment that have trouble responding to `GetBulk` commands, you can instruct NNMi to use `Get` or `GetNext` instead of `GetBulk`. If ☑ enabled, NNMi uses the SNMPv2 `GetBulk` command to gather information from devices in this Region of your network environment. If ☐ disabled, NNMi uses the SNMP `Get` or `GetNext` command to gather information from devices in this Region of your network environment (requesting responses for one SNMP OID at a time).
SNMP Timeout	(Seconds:Milliseconds) Maximum 1 millisecond less than a minute: 59 seconds 999 milliseconds. Time that NNMi waits for a response to an SNMP query before reissuing the request. Both the Discovery Process and the State Poller Service use this setting in this region. For an explanation of how NNMi implements timeout and retry configurations, see "Timeout / Retry Behavior Example for SNMP" (on page 72).
SNMP Retries Count	Maximum number of retries that NNMi issues for an SNMP query before determining the query result to be "unresponsive". Zero means no retries. Both the Discovery Process and the State Poller Service use this setting in this region.
SNMP Port	Default is 161. Specifies the management server's port that NNMi uses when generating SNMP traffic. Both the Discovery Process and the State Poller Service use this setting in this region.
SNMP Proxy Address	*Optional*. IP address of the your SNMP Proxy Server (for example, a proxy that gathers data from non-SNMP devices and can use that data to respond to NNMi SNMP requests). To enable a proxy, you must also provide the port number of your SNMP Proxy Server. See SNMP Proxy Port (next attribute).
SNMP Proxy Port	*Optional*. Port number of the SNMP Proxy Server. To enable a proxy, you must also provide the IP address of your SNMP Proxy Server. See SNMP Proxy Address (previous attribute).

Attribute	Description
SNMP Minimum Security Level	For SNMPv1 or SNMPv2c, configure NNMi to use Community Strings in your network environment: • Community Only (SNMPv1) NNMi tries only SNMPv1 settings. • Community Only (SNMPv1 or v2c) NNMi first tries to use SNMPv2c settings, and, if that fails, NNMi tries SNMPv1 settings. • Community NNMi first tries to use SNMPv2c settings, and, if that fails, NNMi tries SNMPv1 settings. If both SNMPv2c and SNMPv1 fail, NNMi tries SNMPv3 settings if any are available. For SNMPv3, configure NNMi to use the User-based Security Module (USM) level of security required in your network environment (if your environment also uses SNMPv1/SNMPv2c, select Community): • No Authentication, No Privacy • Authentication, No Privacy • Authentication, Privacy See "Timeout / Retry Behavior Example for SNMP" (on page 72) for an explanation of NNMi behavior with each of these choices.

Regional ICMP Settings

Attribute	Description
Enable ICMP Communication	If ☑ enabled, NNMi generates network traffic with ICMP protocol in this region. If ☐ disabled, NNMi does not generate any ICMP traffic on your network in this region: • Addresses in this Region (both previously discovered and newly discovered) have a State attribute value of "Not Polled" and a Status attribute value of "No Status" with the IP Address map-symbol color set to beige. • Nodes with all IP addresses and interfaces showing a Status attribute value of "No Status" have a map-symbol background shape color set to beige. However, it is possible for a node to have IP addresses in multiple regions with multiple Status values. **Note**: See "Monitoring Network Health" (on page 215) for information about enabling/disabling ICMP communication specifically for the State Poller Service.
ICMP Timeout	(Seconds:Milliseconds) Maximum 1 millisecond less than a minute: 59 seconds 999 milliseconds. Time that NNMi waits for a response to an ICMP query before reissuing the request in this region. For an explanation of how NNMi implements timeout and retry configurations, see "Timeout / Retry Behavior Example for ICMP" (on page 73).
ICMP Retries Count	Maximum number of retries that NNMi issues for an ICMP query in this region before logging an error. Zero means no retries.

Configure Address Ranges for Regions

To configure an address range for this region:

1. Navigate to the **Region Included Address Range** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Communication Configuration**.
   c. Navigate to the **Regions** tab.
   d. Do one of the following:
      - To establish a region definition, click the New icon, and continue.
      - To edit a region definition, select a row, click the Open icon, and continue.
   e. In the **Communication Region** form, navigate to the **Included Address Regions** tab.
   f. Do one of the following:
      - To establish an address range setting, click the New icon, and continue.
      - To edit an address range setting, select a row, click the Open icon, and continue.
      - To delete an address range setting, select a row and click the Delete icon.
2. Provide address range definition (see table).

   If you provide multiple IP address ranges for a region, each device must pass at least one to meet the criteria.

   Tip: If you provide both IP address ranges and hostname wildcards, each device must pass at least one in either category (not both) to meet the criteria.
3. Click **Save and Close** to return to the Communication Region form.
4. Click **Save and Close** to return to the Communication Configuration form.
5. Click **Save and Close** to apply your changes.

Address Range Definition Attribute

Attribute	Description
IP Range	To specify a range of IP addresses for this Communications Region, use one of the following. Pick one address notation style, combinations of wildcards and CIDR notation are not allowed within one address range. You can provide multiple address range settings: • **IPv4 address wildcard notation**. An IPv4 Address range is a modified dotted-notation where each octet is one of the following: ▪ A specific octet value between 0 and 255 ▪ A low-high range specification for the octet value (for example, "112-119") ▪ An asterisk (*) wildcard character which is equivalent to the range expression "0-255"

Attribute	Description

Note: The following two IPv4 addresses are considered invalid: `0.0.0.0` and `127.0.0.0`

Examples of valid IPv4 address wildcards include:

```
10.1.1.*
10.*.*.*
10.1.1.1-99
10.10.50-55.*
10.22.*.4
10.1-9.1-9.1-9
```

- **IPv4 Classless Inter-Domain Routing (CIDR) notation**.

  The CIDR notation specifies the number of consecutive bits in the IPv4 address that must match.

  For example, `10.2.120.0/21`

  Note: NNMi does not support CIDR subnet mask notation such as, `10.2.120.0/255.255.248.0`

Example IPv4 Prefix Length Values	Number of Usable IPv4 Addresses
28	14 (16-2=14)*
29	6 (8-2=6)*
30	2 (4-2=2)*
31	2

* Two IPv4 addresses are reserved in each subnet. The first IPv4 address is used for the network itself and the last IPv4 address is reserved for broadcast.

- **IPv6 address wildcard notation**

  Separate each 16-bit value of the IPv6 address with a colon. The 16-bit value can be any of the following:

  - A specific hexadecimal value between `0` and `FFFF` (case insensitive).
  - A low-high range specification of the hexadecimal value (for example, `1-1fe`).
  - An asterisk (`*`) wildcard character (equivalent to the range expression `0-ffff`).

  Note: The standard IPv6 short-hand notation (`::`) is allowed to express one or more 16-bit elements of zero (`0`) values. However, the mixed IPv6/IPv4 dot-notation (for example, `2001:d88::1.2.3.4`) is not allowed as an IPv6 address range.

  Valid examples of ranges in modified IPv6 address notation include the following:

  ```
  2001:D88:0:A00-AFF:*:*:*:*
  2001:D88:1:*:*:*:*:*
  2001:D88:2:0:a07:ffff:0a01:3200-37ff
  ```

- **IPv6 Classless Inter-Domain Routing (CIDR) notation**

  The CIDR notation specifies the number of consecutive bits in the IPv6 address that must match.

  `1080:0:a00::/44` (equivalent to modified IPv6 address notation `1080:0:a00-`

Attribute	Description
	`a0f:*:*:*:*:*`) For example, valid IPv6 address ranges in CIDR notation include the following: `2001:d88:0:a00::/56` (equivalent to modified IPv6 address notation `2001:D88:0:A00-AFF:*:*:*:*`) `2001:d88:1::/48` (equivalent to modified IPv6 address notation `2001:D88:1:*:*:*:*:*`)

Attribute	Description

Configure Hostname Filters for Regions

Define the Communication Region with hostname patterns.

To establish a Hostname Filter setting:

1. Navigate to the **Region Hostname Filter** form.
   - From the workspace navigation panel, select the **Configuration** workspace.
   - Select the **Communication Configuration**.
   - Navigate to the **Regions** tab.
   - Do one of the following:
     - To create a region definition, click the New icon.
     - To edit a region definition, select a row, click the Open icon.
   - In the **Communication Region** form, access the **Hostname Filters** tab.
   - Do one of the following:
     - To create a hostname wildcard definition, click the New icon.
     - To edit a hostname wildcard definition, select a row, click the Open icon.
     - To delete a hostname wildcard setting, select a row and click the Delete icon.
2. Type an appropriate hostname filter (see table).

   If you provide multiple hostname wildcard expressions for a region, each device must pass at least one to meet the criteria for the Region.

   Tip: If you provide both hostname wildcards and IP address ranges, each device must pass at least one in either category (not both) to meet the criteria for the Region.
3. Click **Save and Close** to return to the Communication Region form.
4. Click **Save and Close** to return to the Communication Configuration form.
5. Click **Save and Close** to apply your changes. See "Discovering Your Network" (on page 113) and Verify Device Configuration Details.

Node Hostname Filter Definition

Attribute	Description
Hostname Filter	Enter a wildcard expression using the following characters as wildcards: • ? = one character • * = multiple characters Wildcard expressions are *not case-sensitive*. So a wildcard of ABC* would match devices with hostnames beginning with ABC*, abc*, and Abc* **Caution**: The Hostname attribute value on the Node form of the discovered node must match (not case-sensitive) what is entered here. NNMi follows a set of rules to dynamically generate the value stored in the NNMi database for each Node's Hostname. Click here for details.

Attribute	Description
	Note: The actual Hostname *might be converted* to all uppercase or all lowercase before it is added to the NNMi database (depending on how the NNMi administrator configured settings in the `nms-topology.properties` file). See the information about the `nms-topology.properties` file in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`. • If the Node supports SNMP, NNMi requests the Hostname using the IP Address of the associated SNMP agent (the Management Address attribute value on the Node form). If the NNMi administrator chooses ☑ **Enable SNMP Address Rediscovery** in the Communication Configuration: ▪ If the SNMP Agent does not respond, NNMi checks for another Management Address to request the Hostname, and the Hostname could change. ▪ If the SNMP Agent associated with the node changes, the Management Address and Hostname could change. If the NNMi administrator disables ☐ **Enable SNMP Address Rediscovery** in the Communication Configuration: ▪ If the SNMP Agent does not respond, NNMi uses the previously gathered Management Address attribute value to request the Hostname. ▪ If the SNMP Agent associated with the node changes, NNMi uses the previously gathered Management Address attribute value to request the Hostname. • If the Node does not support SNMP, no Management Address is available. NNMi requests a Hostname starting with the lowest IP Address associated with the node (a Discovery Seed value or an IP address value gathered from a neighboring device). NNMi uses the first Hostname provided. The Hostname might change during a future discovery cycle.

Configure SNMPv1/v2 Community Strings for Regions

If more than one SNMPv1 or SNMPv2c "get" community string is used within this region, repeat this step any number of times. Order does not matter because all community strings defined for this Region are checked in parallel.

During initial discovery, NNMi tries many community strings until a match is found. After a match is identified for a Node, the information is recorded to prevent future authentication errors.

NNMi uses the SNMPv2 settings to discover the SNMPv2 information about your network. This also determines whether NNMi *receives or discards incoming* SNMPv2 traps. Click here for more information.

- If the *incoming* trap's Source Node object or Source Object (such as card or interface) has not yet been discovered by NNMi, see "Handle Unresolved Incoming Traps" (on page 432) for additional information. See also "Configure Network Devices to Send SNMP Notifications to NNMi" (on page 428).
- If the Source Node or Source Object of the *incoming* trap has been discovered by NNMi using SNMPv2c or SNMPv1, NNMi accepts *incoming* traps from SNMPv2c or SNMPv1, but NNMi discards *incoming* traps from SNMPv3.

- NNMi discards traps that have no Incident Configuration or with an Incident Configuration set to Disabled.
- If either the Source Node or Source Object has *Management Mode* set to **Not Managed** or **Out of Service** in the NNMi database, NNMi always discards the incoming trap. See "Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " (on page 257).

  NNMi provides the Management Mode workspace so that you can quickly view lists of all nodes, interfaces, cards, addresses, or node components that NNMi is not currently discovering or monitoring. For information about these views:

Note: If you want the NNMi management server to *forward* SNMPv2 traps to other machines in your network environment, see "Configuring Trap Forwarding" (on page 297) for additional configuration steps.

To provide a community string for this region:

1. Navigate to the **Communication Region** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Communication Configuration**.
   c. Navigate to the **Regions** tab.
   d. Do one of the following:
      - To establish a region definition, click the New icon, and continue.
      - To edit a region definition, select a row, click the Open icon, and continue.
2. In the **Communication Region** form, navigate to the **SNMPv1/v2 Community Strings** tab.
3. To provide a *read community string*, navigate to the **Read Community Strings** table and do one of the following:

   Note: If you do not provide any community strings, NNMi uses the Default Community Strings.

   - To establish a community string setting, click the New icon, and provide the required information (see table).
   - To edit a community string setting, select a row, click the Open icon, and provide the required information (see table)
   - To delete a community string setting, select a row and click the Delete icon
4. To provide a *write community string* for this region, navigate to **the Write Community String** attribute (see table).

   Note: If you do not provide any community strings, NNMi uses the Default Community Strings.
5. Click **Save and Close** to return to the Communication Region form.
6. Click **Save and Close** to return to the Communication Configuration form.
7. Click **Save and Close** to apply your changes.

SNMPv1 or SNMPv2c Community String for this Region

Attribute	Description
Read Community String	The SNMPv1 or SNMPv2c "get" (read-only) community string that is used for this region (case-sensitive). **Tip**: If no values appear in this table, the default settings are used (see "Configure Default Community Strings (SNMPv1 or SNMPv2c)" (on page 73)). Many proxy vendors use the *read community string* for specifying remote target information. NNMi supports substitution parameters within read community strings for SNMPv1 or SNMPv2 proxy environments. Click here for more information. Copy and paste these codes at the end of your read community string to provide the values required by your proxy environment. NNMi substitutes the actual attribute values from the NNMi database at runtime: ${contextName} = Used for specifying VLAN context for switches (VLAN associated with the remote target node) ${managementAddress} = Node form, Management Address attribute value (the remote target node) ${snmpPort} = SNMP Agent form, UDP Port attribute value (SNMP agent associated with the remote target node)
Write Community String	*Optional*. For use with the nnmsnmpset.ovpl command line tool. SNMPv1 and SNMPv2c require that you know the SNMP agent's *write community string* before you can change settings on any device. The nnmsnmpset.ovpl command can use the value you provide here, rather than requiring that you type the write community string each time you invoke the command. **Tip**: If no value is provided here, the default settings are used (see "Configure Default Community Strings (SNMPv1 or SNMPv2c)" (on page 73)).

Configure SNMPv3 Settings for Regions

NNMi can use SNMPv3 user-based security model (USM) settings to access devices.

To view the current list of SNMPv3 USM settings for a Region:

1. Navigate to the **SNMPv3 Settings** tab on the Communication Region form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Communication Configuration**.
   c. Navigate to the **Regions** tab.
   d. Do one of the following:
      - To create a region definition, click the New icon.
      - To edit a region definition, select a row, click the Open icon.
   e. In the **Communication Region** form, access the **SNMPv3 Settings** tab.
2. The displayed table lists the Unique Name of each SNMPv3 USM setting for this region.

Note: NNMi tries to use the Specific Node SNMPv3 Settings. If none match, NNMi tries the Region SNMPv3 Settings provided here. If none match, NNMi tries the default SMNPv3 settings.

3. You can also do the following:
   - To establish a new setting, click the New icon. See "Communication Region SNMPv3 Settings form" (on page 93).

     Click **Save and Close** to return to the Communication Region form.
   - To edit an existing setting, select a row, click the Open icon. See "Communication Region SNMPv3 Settings form" (on page 93).

     Click **Save and Close** to return to the Communication Region form.
   - To delete a setting from the Region's list, select a row and click the Delete icon.

     Note:The record remains in the database for possible use elsewhere and is simply removed from this Communication Region's list.
4. Click **Save and Close** to return to the Communication Configuration form.
5. Click **Save and Close** to apply your changes.

Communication Region SNMPv3 Settings form

NNMi can use SNMPv3 user-based security model (USM) settings to access devices.

NNMi tries to use the current SNMPv3 Settings attribute value from Specific Node Settings. If none match, NNMi tries the Region SNMPv3 Settings provided here. If none match, NNMi tries the default SMNPv3 settings.

To configure an SNMPv3 Setting for a Region:

1. Navigate to the **Communication Region SNMPv3 Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Communication Configuration**.
   c. Navigate to the **Regions** tab.
   d. Do one of the following:
      - To create a region definition, click the New icon.
      - To edit a region definition, select a row, click the Open icon.
   e. In the **Communication Region** form, navigate to the **SNMPv3 Settings** tab.
   f. Do one of the following:
      - To create an SNMPv3 Setting definition, click the New icon.
      - To edit an SNMPv3 Setting, select a row, click the Open icon.
      - To remove an SNMPv3 Setting from this Region, select a row, click the Delete icon.

        Note: The record remains in the database for possible use elsewhere and is simply removed from this Communication Region's list.
2. Click the SNMPv3 Settings Lookup icon and select one of the options from the drop-down

menu:

- Quick View to display summary information for the currently configured (selected) SNMPv3 Setting name.
- Quick Find to view and select from the list of all existing SNMPv3 Settings (for more information see "Use the Quick Find Window" (on page 28)).
- Open to display the details of the currently configured (selected) SNMPv3 Setting (see "SNMPv3 Settings Form" for more information).
- New to create a new SNMPv3 Setting (see "SNMPv3 Settings Form" for more information).

3. Click **Save and Close** to return to the Communication Region SNMPv3 Settings form.
4. Click **Save and Close** to return to the Communication Region form.
5. Click **Save and Close** to return to the Communication Configuration form.
6. Click **Save and Close** to apply your changes.

Configure Credential Settings for Regions (*NNM iSPI NET*)

The HP Network Node Manager iSPI Network Engineering Toolset Software uses Default Credential Settings to access devices when running Diagnostics either automatically or when the **Actions → Run Diagnostics** option is used. (See "Configure Diagnostics for an Incident (NNM iSPI NET)" (on page 421) and Node Form: Diagnostics Tab for more information.)

NNM iSPI NET uses Secure Shell (SSH) to establish a secure connection with devices in your network environment, using the credentials provided here. If the SSH attempt fails, NNMi uses Telnet protocol as the communication method.

To provide credential settings for this region:

1. Navigate to the **Region Device Credentials** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Communication Configuration**.
   c. Navigate to the **Regions** tab.
   d. Do one of the following:
      - To establish a region definition, click the New icon, and continue.
      - To edit a region definition, select a row, click the Open icon, and continue.
   e. In the **Communication Region** form, navigate to the **Device Credentials** tab.
   f. Do one of the following:
      - To establish a credential setting, click the New icon, and continue.
      - To edit a credential setting, select a row, click the Open icon, and continue.
      - To delete a credential setting, select a row and click the Delete icon
2. Provide the attribute values of credentials for this region (see table).

Note: NNMi tries to use the Specific Node Device Credentials. If none match, NNMi tries the Region Device Credential settings provided here. If none match, NNMi tries the Default Device Credentials.

3. Click **Save and Close** to return to the Communication Region form.
4. Click **Save and Close** to return to the Communication Configuration form.
5. Click **Save and Close** to apply your changes.

Device Credential Attributes for this Region

Attribute	Description
User Name	Type the user name that you want NNMi to use for logging into devices in this Communication Region.
Password	Type the password that you want NNMi to use for logging into devices in this Communication Region. **Note**: NNMi encrypts the password and displays asterisks for this attribute. If you want to change the password, first clear the asterisks displayed in the Password attribute and enter the new Password value.

Configure Specific Nodes (Communication Settings)

Configuring communication protocols for specific devices is optional.

Use the Specific Node Settings tab to fine tune communication protocol usage and settings for a particular device within your environment. For example, provide settings for your most important devices, or disable communication with the least important devices.

When you leave a field blank, NNMi uses the next applicable configuration setting for that field in the following order:

- The value configured for a Region that includes this device. If multiple Region definitions include this device (for example, buildings, floors within those buildings, or workgroups within a particular floor), the first match applies (the matching region with the lowest Ordering number) . See "Configure Regions (Communication Settings)" (on page 79).
- The default value for this field (see "Configure Default SNMP and ICMP Protocol Settings" (on page 67), "Configure Default Community Strings (SNMPv1 or SNMPv2c)" (on page 73), "Configure the Default Device Credentials (NNM iSPI NET)" (on page 78), and "Configure Default SNMPv3 Settings" (on page 76).

Note: NNMi enables you to set up one or more SNMP Proxy Servers in the cases where an SNMP node is otherwise unreachable (for example, when a node you want to manage is behind a firewall). To enable NNMi to use the SNMP Proxy Server, when you configure communication protocols for specific devices, you must include the IP address and port number on the SNMP Proxy Server. See "Specific Node Settings Form (Communication Settings)" (on page 96) for more information.

To configure specific devices, you have two choices:

- "Specific Node Settings Form (Communication Settings)" (on page 96).
- "Load Specific Node Settings from a File" (on page 106)

Specific Node Settings Form (Communication Settings)

Create specific node settings to control the way NNMi monitors your most important devices or least important devices.

Tip: If no value is provided for an attribute in the Communication Node form, NNMi uses the applicable Region settings and if none match, NNMi uses the default settings.

If configuring Specific Node Settings, see the *HP Network Node Manager i Software Deployment Reference* for additional considerations: http://h20230.www2.hp.com/selfsolve/manuals.

To configure communication protocol settings for a specific node:

1. Access the **Specific Node Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Communication Configuration**.
   c. Navigate to the **Specific Node Settings** tab.
   d. Do one of the following:
      - To establish settings for a node, click the New icon, and continue.
      - To edit settings for a node, select a row, click the Open icons, and continue.
      - To delete settings for a node, select a row and click the Delete icon.
2. Provide the communication protocol settings for the node (see the Basic Settings table, SNMP Settings table, and ICMP Settings table).

   For an explanation of how NNMi implements timeout and retry configurations, see "Timeout / Retry Behavior Example for SNMP" (on page 72) and "Timeout / Retry Behavior Example for ICMP" (on page 73).
3. *Optional*. Make additional configuration choices. Click here for a list of choices .
4. Click **Save and Close** to return to the Communication Configuration form.
5. Click **Save and Close** to apply your changes.

Basic Settings for this Device

Attribute	Description
Target Hostname	The *case-sensitive* Hostname attribute value from the Node form of the discovered node must match what is entered here. NNMi follows a set of rules to dynamically generate the value stored in the NNMi database for each Node's Hostname. Click here for details. **Note**: The actual Hostname *might be converted* to all uppercase or all lowercase before it is added to the NNMi database (depending on how the NNMi administrator configured settings in the `nms-topology.properties` file). See the information about the `nms-topology.properties` file in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`. • If the Node supports SNMP, NNMi requests the Hostname using the IP Address of the associated SNMP agent (the Management Address attribute value on the Node form).

Attribute	Description
	If the NNMi administrator chooses ☑ **Enable SNMP Address Rediscovery** in the Communication Configuration: ▪ If the SNMP Agent does not respond, NNMi checks for another Management Address to request the Hostname, and the Hostname could change. ▪ If the SNMP Agent associated with the node changes, the Management Address and Hostname could change. If the NNMi administrator disables ☐ **Enable SNMP Address Rediscovery** in the Communication Configuration: ▪ If the SNMP Agent does not respond, NNMi uses the previously gathered Management Address attribute value to request the Hostname. ▪ If the SNMP Agent associated with the node changes, NNMi uses the previously gathered Management Address attribute value to request the Hostname. • If the Node does not support SNMP, no Management Address is available. NNMi requests a Hostname starting with the lowest IP Address associated with the node (a Discovery Seed value or an IP address value gathered from a neighboring device). NNMi uses the first Hostname provided. The Hostname might change during a future discovery cycle.
Preferred Management Address	Do one of the following: • Specify the address you want NNMi to use for SNMP communications with this device. If you enter an invalid or unreachable address, the device is not discovered or monitored. • Leave this attribute empty. NNMi dynamically selects the management address, based on responses from the device's SNMP agent. **Note**: The NNMi administrator can over-ride this setting. See the Enable SNMP Communication attribute and the Enable SNMP Address Rediscovery attribute settings.

SNMP Settings for this Device

Attribute	Description
Enable SNMP Communication	If ☑ enabled, the Discovery Process and State Poller Service generate network traffic with SNMP protocol to discover and monitor this device.

Attribute	Description
	Note: Your choice might be overridden if Monitoring Configuration settings disable SNMP usage for the State Poller Service, see "Set Global Monitoring" (on page 217) or "Configure Monitoring Behavior" (on page 216). If ☐ disabled, NNMi does not generate any SNMP traffic to this device. **Caution**: With no SNMP data, Spiral Discovery interprets each IP Address as a separate node, Causal Engine calculates Status based only on IP address State, previously discovered Interfaces show a State attribute value of "Not Polled" and a Status attribute value of "No Status" with the Interface map-symbol color set to beige, and no new Interfaces are discovered.

Attribute	Description
Enable SNMP Address Rediscovery	**Note**: The NNMi administrator can over-ride this setting. See the Enable SNMP Communication and the Preferred Management Address attributes. If ☑ enabled, NNMi automatically identifies which management address (SNMP agent) to use for each device. If the initially configured address becomes unreachable, NNMi automatically locates another SNMP agent, if possible, and changes the management address attribute value. Click here for more information. When NNMi first discovers a node, the *seed address* (provided by the NNMi administrator) or discovered address (for non-seeded nodes) becomes the initial Management Address of the node. After NNMi builds an inventory of all IP addresses associated with the node (see "How Spiral Discovery Works" (on page 114)), NNMi follows a set of rules to determine which address is the best choice for each node's Management Address: **Note**: With NNMi Advanced, the NNMi administrator specifies whether NNMi prefers IPv4 or IPv6 addresses when selecting the Management Address. See the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals` 1. NNMi ignores the following addresses when determining which Management Address is most appropriate: ■ Any address of an administratively-down interface. ■ Any address that is virtual (HSRP/VRRP). ■ Any IPv4 **Anycast Rendezvous Point IP Address**[1] or IPv6 Anycast address. ■ Any address in the reserved loopback network range. IPv4 uses 127/24 (`127.*.*.*`) and IPv6 uses `::1`. ■ Any IPv6 **link-local address**[2]. 2. NNMi prefers loopback interfaces for management communication. ■ If a node has only one **loopback address**[3], and the associated SNMP agent responds to NNMi, that address becomes the Management Address. ■ If a node supports multiple loopback addresses, NNMi queries each loopback addresses, starting with the lowest number. NNMi uses the loopback address with the lowest number from which the SNMP agent responds (for example, 10.16.42.197 is a lower number than 10.16.197.42). 3. If no loopback address responds, NNMi queries the last-known Management Address (if any).

[1]Rendezvous Point addresses are loopback addresses used for routers in multi-cast network configurations.
[2]A non-routable IPv6 unicast address only used for communication with other nodes on the same link (LAN or VLAN). Link local addresses cannot be used for communication that must be forwarded through a router. IPv6 auto-configuration automatically assigns a unique link local address in the fe80::/10 address space to each IPv6-enabled interface on a system.
[3]The address associated with the loopback interface. The loopback interface is a virtual interface on a device that provides a route for internal communication. Many vendors provide a specially configured loopback for management purposes. Exact details of how loopbacks are configured varies by vendor and model. See each device's documentation for details. NNMi identifies these loopback addresses by using IfType 24, softwareloopback from the IANA ifType-MIB.

Attribute	Description
	4. If no response, NNMi queries up to 10 of any remaining IP addresses in the node's IP address inventory, starting with the lowest number. NNMi uses the address with the lowest number from which the SNMP agent responds. 5. If no response, NNMi might be configured to repeat the sequence using SNMPv1, SNMPv2c, or SNMPv3 in the order specified by the NNMi administrator (Communication Configurations *SNMP Minimum Security Level* settings). 6. When all else fails, NNMi retains the last known Management Address (if any) and automatically changes the State of that SNMP Agent object to Critical. This process is repeated during each Auto-Discovery cycle, and the Management Address can change. For example, NNMi's inventory of addresses for the node expands, or the current Management Address does not respond to SNMP queries due to network problems or node reconfiguration. The NNMi administrator can prevent changes to the management address using the Communication Configurations *Enable SNMP Address Rediscovery* or *Preferred Management Address* setting. If ☐ disabled, when the current management address (SNMP agent) becomes unreachable, NNMi reclassifies the node as a non-SNMP node until the previously configured management address is available again.

Attribute	Description
Enable SNMP GetBulk	*Applies only to SNMPv2 or higher.* Some devices do not respond well to multiple SNMP OIDs requests at one time. If you have devices in your network environment that have trouble responding to `GetBulk` commands, you can instruct NNMi to use `Get` or `GetNext` instead of `GetBulk`. If ☑ enabled, NNMi uses the SNMPv2 `GetBulk` command to gather information from this device. If ☐ disabled, NNMi uses the SNMP `Get` or `GetNext` command to gather information from this device (requesting responses for one SNMP OID at a time).
SNMP Timeout	(Seconds:Milliseconds) Maximum 1 millisecond less than a minute: 59 seconds 999 milliseconds. Time that NNMi waits for a response to an SNMP query before reissuing the request. Both the Discovery Process and the State Poller Service use this setting for this device. For an explanation of how NNMi implements timeout and retry configurations, see "Timeout / Retry Behavior Example for SNMP" (on page 72).
SNMP Retries Count	Maximum number of retries that NNMi issues for an SNMP query before determining the query result to be "unresponsive". Zero means no retries. Both the Discovery Process and the State Poller Service use this setting for this device.
SNMP Port	Default is 161. Specifies the management server's port that NNMi uses when generating SNMP traffic. Both the Discovery Process and the State Poller Service use this setting for this device.
SNMP Proxy Address	*Optional*. IP address of the your SNMP Proxy Server (for example, a proxy that gathers data from non-SNMP devices and can use that data to respond to NNMi SNMP requests). To enable a proxy, you must also provide the port number of your SNMP Proxy Server. See SNMP Proxy Port (next attribute).
SNMP Proxy Port	*Optional*. Port number of the SNMP Proxy Server. To enable a proxy, you must also provide the IP address of your SNMP Proxy Server. See SNMP Proxy Address (previous attribute).

Attribute	Description
SNMP Preferred Version	Specifies the SNMP version that NNMi should use when communicating with a device. Select one of the following options: 1 Indicates you want NNMi to try only SNMPv1 settings. **Tip**: Use this option when you do not want NNMi to use `GetBulk` commands on the device. 2 Indicates you want NNMi to use SNMPv2c settings, and, if that fails, try SNMPv1 settings. If both SNMPv2c and SNMPv1 fail, NNMi tries SNMPv3 settings if any are available. 3 Indicates you want NNMi to use SNMPv3 settings for this device. NNMi uses the **SNMPv3 Settings** configuration to determine which of the following User-based Security Module (USM) levels of security to provide: • No Authentication, No Privacy • Authentication, No Privacy • Authentication, Privacy See "Configure Default SNMP and ICMP Protocol Settings" (on page 67) for more information.

Note: The SNMP Minimum Security Level is determined by the settings on the Communication Configurations' Specific Node Settings form, **SNMPv3 Settings** tab where SNMPv3 Settings for this Node are established.

ICMP Settings for this Device

Attribute	Description
Enable ICMP Communication	If ☑ enabled, NNMi generates network traffic with ICMP protocol to this device. If ☐ disabled, NNMi does not generate any ICMP traffic to this device: • Addresses in this Node (both previously discovered and newly discovered) have a State attribute value of "Not Polled" and a Status attribute value of "No Status" with the IP Address map-symbol color set to beige. • If both ICMP and SNMP are disabled, the Node has a Status attribute value of "No Status" have a map-symbol background shape color set to beige. **Note**: Your choice might be overridden if Monitoring Configuration settings disable ICMP usage for the State Poller Service, see "Set Global Monitoring" (on page 217) or "Configure Monitoring Behavior" (on page 216).
ICMP Timeout	(Seconds:Milliseconds) Maximum 1 millisecond less than a minute: 59 seconds 999 milliseconds. Time that NNMi waits for a response to an ICMP query before reissuing the request to this device. For an explanation of how NNMi implements timeout and retry configurations, see "Timeout / Retry Behavior Example for ICMP" (on page 73).
ICMP Retries	Maximum number of retries that NNMi issues for an ICMP query to this device before

Attribute	Description
Count	logging an error. Zero means no retries.

Related Topics:

"Configure Default SNMP and ICMP Protocol Settings" (on page 67)

"Configure Default Community Strings (SNMPv1 or SNMPv2c)" (on page 73)

"Configure Regions (Communication Settings)" (on page 79)

Configure SNMPv1/v2 Community Strings for a Specific Node

Optional. Configure the SNMPv1 or SNMPv2c community strings for each node.

NNMi uses the SNMPv2 settings to discover the SNMPv2 information about your network. This also determines whether NNMi *receives or discards incoming* SNMPv2 traps. Click here for more information.

- If the *incoming* trap's Source Node object or Source Object (such as card or interface) has not yet been discovered by NNMi, see "Handle Unresolved Incoming Traps" (on page 432) for additional information. See also "Configure Network Devices to Send SNMP Notifications to NNMi" (on page 428).
- If the Source Node or Source Object of the *incoming* trap has been discovered by NNMi using SNMPv2c or SNMPv1, NNMi accepts *incoming* traps from SNMPv2c or SNMPv1, but NNMi discards *incoming* traps from SNMPv3.
- NNMi discards traps that have no Incident Configuration or with an Incident Configuration set to Disabled.
- If either the Source Node or Source Object has *Management Mode* set to **Not Managed** or **Out of Service** in the NNMi database, NNMi always discards the incoming trap. See "Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " (on page 257).

  NNMi provides the Management Mode workspace so that you can quickly view lists of all nodes, interfaces, cards, addresses, or node components that NNMi is not currently discovering or monitoring. For information about these views:

Note: If you want the NNMi management server to *forward* SNMPv2 traps to other machines in your network environment, see "Configuring Trap Forwarding" (on page 297) for additional configuration steps.

To provide SNMPv1/v2 community strings for a specific device:

1. Navigate to the **Specific Node Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Communication Configuration**.
   c. Navigate to the **Specific Node Settings** tab.
   d. Do one of the following:
      - To establish a node definition, click the New icon, and continue.
      - To edit a node definition, select a row, click the Open icon, and continue.
2. Navigate to the **SNMPv1/v2 Community Strings** tab.

3. To provide a *read community string*, navigate to the **Read Community String** attribute and provide the appropriate string (see table).

   Tip: If you do not provide any read community string, NNMi uses the applicable Region settings and if none match, NNMi uses the default settings .

4. To provide a *write community string*, navigate to **the Write Community String** attribute and provide the appropriate string (see table).

   Tip: If you do not provide any write community string, NNMi uses the applicable Region setting and if none match, NNMi uses the default setting .

5. Click **Save and Close** to return to the Communication Configuration form.

6. Click **Save and Close** to apply your changes.

SNMPv1 or SNMPv2c Community String for this Device

Attribute	Description
Read Community String	The SNMPv1 or SNMPv2c "get" (read-only) community string that is used for this device (case-sensitive). **Tip**: If you do not provide any read community string, NNMi uses the applicable Region settings and if none match, NNMi uses the default settings . Many proxy vendors use the *read community string* for specifying remote target information. NNMi supports substitution parameters within read community strings for SNMPv1 or SNMPv2 proxy environments. Click here for more information. Copy and paste these codes at the end of your read community string to provide the values required by your proxy environment. NNMi substitutes the actual attribute values from the NNMi database at runtime: ${contextName} = Used for specifying VLAN context for switches (VLAN associated with the remote target node) ${managementAddress} = Node form, Management Address attribute value (the remote target node) ${snmpPort} = SNMP Agent form, UDP Port attribute value (SNMP agent associated with the remote target node)
Write Community String	The SNMPv1 or SNMPv2c "set" (write) community string that is used for this device (case-sensitive). *Optional*. For use with the nnmsnmpset.ovpl command line tool. SNMPv1 and SNMPv2c require that you know the SNMP agent's *write community string* before you can change settings on any device. The nnmsnmpset.ovpl command can use the value you provide here, rather than requiring that you type the write community string each time you invoke the command. **Tip**: If you do not provide any write community string, NNMi uses the applicable Region setting and if none match, NNMi uses the default setting.

Configure SNMPv3 Settings for a Specific Node

NNMi can use SNMPv3 user-based security model (USM) settings to access devices.

NNMi uses the current SNMPv3 Settings provided a node, if available.

To configure an SNMPv3 Settings for a specific node:

1. Navigate to the **Specific Node Settings** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Communication Configuration**.
    c. Navigate to the **Specific Node Settings** tab.
    d. Do one of the following:
        - To establish a node definition, click the New icon, and continue.
        - To edit a node definition, select a row, click the Open icon, and continue.
2. Navigate to the **SNMPv3 Settings** tab.
3. Click the SNMPv3 Settings Lookup icon and select one of the options from the drop-down menu:
    - Quick View to display summary information for the currently configured (selected) SNMPv3 Setting name.
    - Quick Find to view and select from the list of all existing SNMPv3 Settings (for more information see "Use the Quick Find Window" (on page 28)).
    - Open to display the details of the currently configured (selected) SNMPv3 Setting (see "SNMPv3 Settings Form" for more information).
    - New to create a new SNMPv3 Setting (see "SNMPv3 Settings Form" for more information).
4. Click **Save and Close** to return to the Specific Node Settings form.
5. Click **Save and Close** to return to the Communication Configuration form.
6. Click **Save and Close** to apply your changes.

Configure Credential Settings for a Specific Node (*NNM iSPI NET*)

HP Network Node Manager iSPI Network Engineering Toolset Software uses Default Credential Settings to access devices when running Diagnostics either automatically or when the **Actions → Run Diagnostics (iSPI NET only)** option is used. (See "Configure Diagnostics for an Incident (NNM iSPI NET)" (on page 421) and Node Form: Diagnostics Tab for more information.)

NNM iSPI NET uses Secure Shell (SSH) to establish a secure connection with devices in your network environment, using the credentials provided here. If the SSH attempt fails, NNMi uses Telnet protocol as the communication method.

To provide credential settings for a specific node:

1. Navigate to the **Specific Node Device Credentials** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Communication Configuration**.
    c. Navigate to the **Specific Node Settings** tab.
    d. Do one of the following:

- To establish a definition, click the New icon, and continue.
- To edit a definition, click the Open icon in the row representing the configuration you want to edit, and continue.

e. In the **Specific Nodes Settings** form, navigate to the **Device Credentials** tab.

f. Do one of the following:

- To establish a credential setting, click the New icon, and continue.
- To edit a credential setting, click the Open icon in the row representing the configuration you want to edit, and continue.
- To delete a credential setting, select a row and click the Delete icon

2. Provide the attribute values of credentials for this node (see table).

   Note: NNMi tries to use the Specific Node Device Credentials provided here. If none match, NNMi tries the Region Device Credential settings. If none match, NNMi tries the Default Device Credentials.

3. Click **Save and Close** to return to the Specific Node Settings form.
4. Click **Save and Close** to return to the Communication Configuration form.
5. Click **Save and Close** to apply your changes.

Specific Node Device Credential Attributes

Attribute	Description
User Name	Type the user name that you want NNMi to use for logging into this device.
Password	Type the password that you want NNMi to use for logging into this device. **Note**: NNMi encrypts the password and displays asterisks for this attribute. If you want to change the password, first clear the asterisks displayed in the Password attribute and enter the new Password value.

Load Specific Node Settings from a File

Import a list of devices, using a command line command. You also have the option of importing the SNMPv1 or SNMPv2c community strings (read and write) or the SNMPv3 USM settings for each device. This is useful when your SNMP is managed by a change control mechanism. You can bulk insert the SNMP assignments into NNMi. Each assignment shows up as an individual entry in the table on the **Communication Configuration** form's **Specific Node Settings** tab.

If configuring Specific Node Settings, see the *HP Network Node Manager i Software Deployment Reference* for additional considerations: http://h20230.www2.hp.com/selfsolve/manuals.

To import SNMP assignments:

1. On the NNMi management server's hard drive, create a text file according to the specifications in the

nnmcommload.ovpl reference page. Create one line for each device. For more information, see nnmcommload.ovpl

To add comments to your file, place a # character at the beginning of each comment line.

Note: When you load this file, the data in the file overwrites any previously entered information about each Hostname (*case-sensitive*).

2. Use the following command line command to load the information into the NNMi database:

   If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access" (on page 275) for more information.

   Windows:
   `%NnmInstallDir%\bin\nnmcommload.ovpl -u <NNMiadminUserName> -p <NNMiadminPassword> -file <path/filename>`

   UNIX:
   `/opt/OV/bin/nnmcommload.ovpl -u <NNMiadminUserName> -p <NNMiadminPassword> -file <path/filename>`

3. Verify that the import worked properly:
   a. From the workspace navigation panel, select the Configuration workspace.
   b. Select Communication Configuration.
   c. Access the Specific Node Settings tab.
4. Review each entry in the table to verify that the import was successful.

To verify the SNMP configuration for an IP Address, at the command line, type:

Note: For more information, see nnmcommconf.ovpl

If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access" (on page 275) for more information.

Windows:
`%NnmInstallDir%\bin\nnmcommconf.ovpl -u <NNMiadminUsername> -p <NNMiadminPassword> -proto snmp -host <node IP address>`

UNIX:
`/opt/OV/bin/nnmcommconf.ovpl -u <NNMiadminUsername> -p <NNMiadminPassword> -proto snmp -host <node IP address>`

To verify the ICMP configuration for an IP Address, at the command line, type:

Note: For more information, see nnmcommconf.ovpl

If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access" (on page 275) for more information.

Windows:
`%NnmInstallDir%\bin\nnmcommconf.ovpl -u <NNMiadminUsername> -p <NNMiadminPassword> -proto icmp -host <node IP address>`

UNIX:
`/opt/OV/bin/nnmcommconf.ovpl -u <NNMiadminUsername> -p <NNMiadminPassword> -proto icmp -host <node IP address>`

Troubleshooting Communication Settings

After you configure your communication settings and wait until Auto-Discovery completes at least one cycle, you can verify your Communication Settings:

- "Verify That All Nodes Support SNMP" (on page 108)
- "Verify a Node's Communication Settings" (on page 109)
- "Identify Unexpected Overrides of Communication Settings" (on page 110)
- "Resolve Authentication Errors" (on page 111)

You can fine tune NNMi's SNMP/ICMP traffic in the following ways:

- Minimize timeouts and retries.

  When NNMi attempts to contact a node using ICMP / SNMP during an Auto-Discovery cycle, the Communication Configuration settings determine what information NNMi can gather. If the correct ICMP / SNMP settings are not provided or if NNMi discovers non-SNMP devices (see "Verify That All Nodes Support SNMP" (on page 108)), NNMi resorts to timeouts and retries.

  Large timeout values or a high number of retries can degrade overall performance of discovery. If your network environment contains nodes that you know respond slowly to ICMP / SNMP requests, consider using the Regions or Specific Nodes settings to fine tune the number of timeouts and retries NNMi uses during each Auto-Discovery cycle.

- Limit the number of *default* SNMPv1/SNMPv2c Community Strings to ensure efficient Auto-Discovery performance. See "Configure Default Community Strings (SNMPv1 or SNMPv2c)" (on page 73).
- Limit the number of *default* SNMPv3 user-based security model (USM) settings to ensure efficient Auto-Discovery performance. See "Configure Default SNMPv3 Settings" (on page 76).

Verify That All Nodes Support SNMP

After you configure your communication settings and wait until Auto-Discovery completes at least one cycle, check for any nodes that do not respond to SNMP:

1. From the workspace navigation panel, select the **Inventory** workspace.
2. Select the **Nodes** view.
3. Right-click the **Device Profile** column, and select **Create Filter**.
4. Select "contains", and type the following text into **Enter a string**: `No SNMP`.
5. NNMi displays a list of all nodes in your network environment that did not respond to SNMP during Auto-Discovery.
6. Verify that the resulting list is valid.
7. To troubleshoot unexpected results, see:

- "Verify a Node's Communication Settings" (on page 109)
- "Identify Unexpected Overrides of Communication Settings" (on page 110)
- "Resolve Authentication Errors" (on page 111)

Verify a Node's Communication Settings

After you configure your communication settings and wait until Auto-Discovery completes at least one cycle, you can check to determine what settings NNMi is using to communicate with a node of interest.

NNMi provides a report about the communication configuration information for a selected node, including the SNMP and ICMP configuration information.

To display a report of a node's current communication settings:

Note: You must have Administrator role to use this action.

1. Do one of the following:

   Navigate to a table view and select a node:

   a. From the workspace navigation panel, select the workspace of interest. For example, **Inventory**.

   b. Select the view that contains the node with communication settings you want to check. For example, **Nodes**.

   c. From the table view, select the ☑ check box that precedes the node with communication settings you want to check.

   Navigate to a map view and select a node:

   a. From the workspace navigation panel, select the workspace of interest; for example, **Topology Maps**.

   b. Click the view that contains the node with communication settings you want to check; for example **Network Overview**.

   c. From the map view, click the node with communication settings you want to check.

   Navigate to a Node form:

   - From a table view, click the Open icon that precedes the node of interest.
   - From a map view, click the node of interest on the map and click the Open icon.

2. Select **Actions** → **Communication Settings**.

Sometimes a device is temporarily not responding properly to SNMP during NNMi's initial discovery, so NNMi makes the wrong decision about which version of SNMP to use. Or perhaps you deployed upgrades to the SNMP agents in your network environment.

To update NNMi's choice of SNMP version used for a Node or Nodes:

Note: You must have Administrator role to use this action.

1. From the workspace navigation panel, select **Inventory**.
2. Select the **Custom Nodes** view.
3. Click the **Protocol Version** column heading to sort the view according to SNMP version currently being used by NNMi for communications with each SNMP agent in your network environment.

4. Select all rows that you want NNMi to check for SNMP upgrades or changes.
5. select **Actions** → **Configuration Poll**.

   NNMi reconfigures the SNMP Communication settings by verifying the highest SNMP version available to the SNMP Agent assigned to the node (according to your Communication Configuration settings).
6. Click the **Protocol Version** column heading to resort the view according to SNMP version.
7. Verify that NNMi made the expected changes.

   If still receiving unexpected results, see "Identify Unexpected Overrides of Communication Settings" (on page 110).

See "Configuring Communication Protocol" (on page 66) for information about configuring communication settings.

Related Topics

nnmcommconf.ovpl

Identify Unexpected Overrides of Communication Settings

Even if the communication settings enable ICMP / SNMP protocol traffic to an area of your network (see "Configuring Communication Protocol" (on page 66)), that type of traffic might be disabled in the monitoring settings (see "Monitoring Network Health" (on page 215)).

To determine whether your Communication Configuration settings are being overridden:

Note: You must have Administrator role to use this action.

1. Do one of the following:

   Navigate to a table view and select a node:

   a. From the workspace navigation panel, select the workspace of interest. For example, **Inventory**.

   b. Select the view that contains the node with communication settings you want to check. For example, **Nodes**.

   c. From the table view, select the ☑ check box that precedes the node with communication settings you want to check.

   Navigate to a map view and select a node:

   a. From the workspace navigation panel, select the workspace of interest; for example, **Topology Maps**.

   b. Click the view that contains the node with communication settings you want to check; for example **Network Overview**.

   c. From the map view, click the node with communication settings you want to check.

   Navigate to a Node form:

   - From a table view, click the Open icon that precedes the node of interest.
   - From a map view, click the node of interest on the map and click the Open icon.

2. Select **Actions** → **Monitoring Settings**.

   NNMi displays a report showing ICMP and SNMP communication configuration settings for this node.

   Tip: If either the Monitoring settings or the Communication settings disable ICMP or SNMP traffic to a node, NNMi does not use that protocol for communication with that node. See also "Verify a Node's Communication Settings" (on page 109).

   (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager:

   - Node managed by the Global Manager = Actions → Monitoring Settings displays a report, provided by the Global Manager (NNMi management server).
   - Node managed by a Regional Manager = Actions → Monitoring Settings accesses that Regional Manager (NNMi management server) and requests the report.

     Note: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.

Resolve Authentication Errors

To create a list of authentication errors:

1. From the workspace navigation panel, select the **Incident Browsing** workspace.
2. Select an Incident view.
3. Right-click the **Category** column, and select **Create Filter**.
4. Select "equals", and select **Security**.
5. NNMi displays a list of all incidents related to authentication errors; for example an SNMP authentication failure (see also Node Down).

If NNMi generates incidents related to *authentication failure* during discovery, there are several configuration settings that influence authentication errors:

- Communication Configuration.

  Each Node's Management Address is the address NNMi uses to communicate with the Node's SNMP agent. The NNMi administrator can control NNMi behavior:

  - Specify the Management Address for a node (in the Communications Configuration, Specific Nodes settings).
  - Otherwise, let NNMi choose an address from all IP addresses associated with each node. This NNMi behavior can be fine-tuned by the NNMi administrator in the Discovery configuration settings.

  Consider configuring smaller Regions with more focused lists of possible access credentials. Or configure Specific Nodes to avoid requiring NNMi to try multiple possible settings.

- Discovery Configuration.

  The following Discovery Configuration fields influence NNMi's use of SNMP (see "Configure Basic Settings for the Auto-Discovery Rule" (on page 143)):

  - **Discovery Any SNMP Device** field.

    If disabled, NNMi discovers only Routers and Switches that respond to SNMP.

    If enabled, NNMi discovers all devices that respond to SNMP.

- **Non-SNMP Devices** field.

  If disabled, when there is no SNMP response from the device, NNMi does not discover information about the device or add a record of that device to the NNMi database.

  If enabled, NNMi discovers devices that do not respond to SNMP and assigns the Device Profile named No SNMP as the basis of the database record.

NNMi's access to SNMP agents is also influenced by the set of rules for choosing management addresses and settings to exclude certain addresses.

- Device Profiles.

  The Device Profiles' **Force Device** attribute setting influences NNMi's use of SNMP (see Device Profile form).

- Monitoring Configuration.

  NNMi discovers and monitors devices in an ongoing basis (see "Monitoring Network Health" (on page 215)). To control management address discovery after the first NNMi discovery cycle, use the **Enable SNMP Address Rediscovery** field to control NNMi behavior when previously discovered SNMP agents quit responding (for example, when you reconfigure the device's SNMP agent):

  - If disabled, NNMi reports the device as Node Down and does not attempt to find another Communication Configuration setting that works.
  - If enabled, NNMi retries any configured values in search of one that works.

Discovering Your Network

Configure NNMi to discover only the nodes that are important to you and your team.

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

This dynamic discovery process continues over time. When things change in your network management domain, Spiral Discovery automatically updates information according to a schedule that you set. The topology maps always reflect accurate timely information about any changes in your network.

Note: Review and complete the prerequisites before configuring discovery, "Prerequisites for Discovery" (on page 124).

You decide which nodes are discovered and how often NNMi checks for new devices in your network (see "Determine Your Approach to Discovery" (on page 127) for ideas). The steps required depend on what you want to do:

- "Adjust the Rediscovery Interval" (on page 138) – *Optional*. The time NNMi waits between the discovery cycles that keep your network information current. By default, NNMi updates information about devices and connections every 24 hours.
- "Configure the Node Name Strategy" (on page 140) – *Optional*. Choose the node naming strategy for NNMi to use for the map icons and in the Name column of the table views.
- "Configure Auto-Discovery Rules" (on page 141) – *Optional*. Specify whether you want NNMi to automatically discover groups of network devices (identified by IP address ranges and MIB II sysObjectIDs). NNMi extends discovery by using requests for Address Resolution Protocol (ARP) cache information about neighbors. NNMi uses a variety of protocols to gather information from all neighbor devices. See "Auto-Discovery Rules" (on page 120) for more details.

  Specify whether NNMi uses Ping Sweep (ICMP ping) or your Discovery Seeds as starting points for gathering information about neighboring devices. Note that with NNMi Advanced, Ping Sweep works only with IPv4 addresses, not IPv6 addresses.

  NNMi discovers any devices that comply with your rule configurations, and creates a record of each device in the NNMi database. If the device supports SNMP, all addresses for that device are combined into one Node object. If the device does not support SNMP, NNMi queries DNS to determines the hostname. If this hostname matches another non-SNMP node, NNMi merges the information to create only one node with multiple associated addresses.
- "Configure an Excluded IP Addresses Filter" (on page 155) – *Optional*. Specify addresses that you do not want NNMi to discover.
- "Specify Discovery Seeds: Initial Routers or Specific Nodes to be Discovered" (on page 158) – *Optional*. Use Discovery Seeds to accomplish either of the following purposes:
  - Limit Spiral Discovery to only the seeds that you specify.
  - Provide seeds as starting points for your Auto-Discovery Rules.

For details about how Spiral Discovery works:

For a list of the types of things NNMi can discover, see About Map Symbols.

From the information collected, NNMi constructs a model of your network configuration in the database, and displays this information in the map views. See View Maps of Network Connectivity for more information about the available map views.

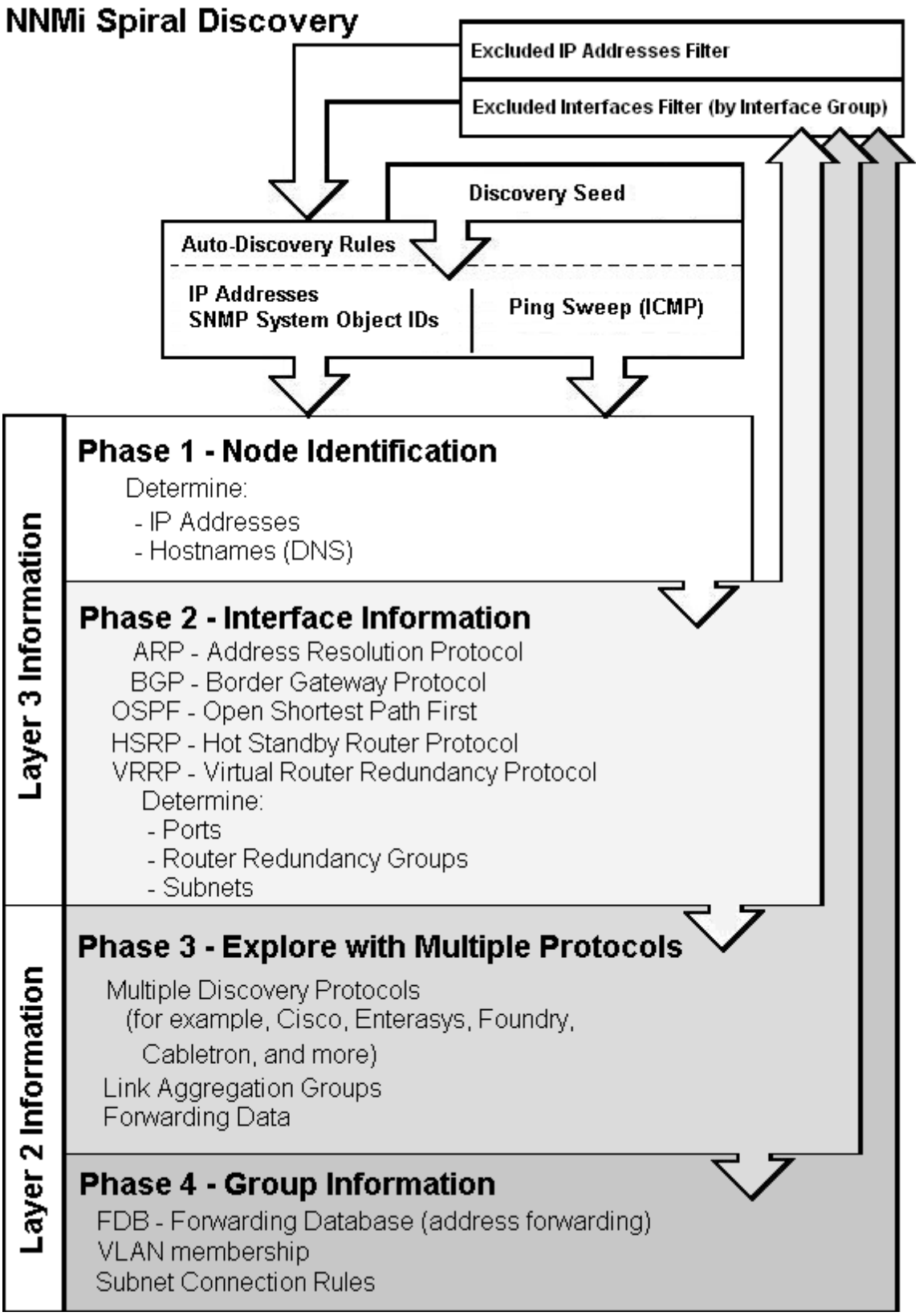
How Spiral Discovery Works

NNMi uses a variety of network protocols (read-only queries) within your defined network management domain to gather information about each discovered device. You see the real-time accumulation of information as it is collected, rather than waiting until NNMi discovers your entire network environment.

Spiral Discovery dynamically gathers two categories of information from each discovered node:

1. Information about the node — NNMi gathers detailed information about each device. You can review this data on the device's Node form. Examples of configuration details include IP address, subnet information, sysObjectID, number of interfaces, and version of SNMP supported.
2. Conectivity details — NNMi gathers information about how devices are connected to each other on **Layer 2**[1] and **Layer 3**[2] of your network.

[1]Refers to the Data Link layer of the multilayered communication model, Open Systems Interconnection (OSI). The Data Link layer moves data across the physical links in the network. The switches and bridges are devices that redirect data messages at the layer 2 level, using the destination Media Access Control (MAC) address to determine where to direct the message.
[2]Refers to the Network layer of the multilayered communication model, Open Systems Interconnection (OSI). The Network layer is concerned with knowing the address of the neighboring nodes in the network, selecting routes and quality of service, and recognizing and forwarding incoming messages to local host domains. The router and switch-router are the devices that redirect data messages at the Layer 3 level. Everything in a subnet is connected at the Layer 3 (IP) level.


NNMi Spiral Discovery
Excluded IP Addresses Filter
Excluded Interfaces Filter (by Interface Group)
Discovery Seed
Auto-Discovery Rules
IP Addresses
SNMP System Object IDs
Ping Sweep (ICMP)
Layer 3 Information
Phase 1 - Node Identification
Determine:
- IP Addresses
- Hostnames (DNS)
Phase 2 - Interface Information
ARP - Address Resolution Protocol
BGP - Border Gateway Protocol
OSPF - Open Shortest Path First
HSRP - Hot Standby Router Protocol
VRRP - Virtual Router Redundancy Protocol
Determine:
- Ports
- Router Redundancy Groups
- Subnets
Layer 2 Information
Phase 3 - Explore with Multiple Protocols
Multiple Discovery Protocols
(for example, Cisco, Enterasys, Foundry,
Cabletron, and more)
Link Aggregation Groups
Forwarding Data
Phase 4 - Group Information
FDB - Forwarding Database (address forwarding)
VLAN membership
Subnet Connection Rules

Spiral Discovery checks for changes according to a schedule determined by the Rediscovery Interval. NNMi administrators can set the schedule to meet any service-level agreement (SLA) commitments.

After NNMi completes initial discovery of your network, ongoing discovery takes over according to the Rediscovery Interval:

- If a new node is added to your defined network management domain, NNMi dynamically updates the topology database and maps. The node form provides details of the new node's configuration. The maps reflect the new connectivity information.
- If configuration settings change on an existing node, NNMi dynamically updates the database and maps to reflect the changes.

  The only exception is when non-SNMP addresses that had the same DNS hostname are changed to have separate DNS hostnames, NNMi must completely rediscover the non-SNMP nodes to correctly update the database objects (node, interface, address, connection, and incidents). The NNMi administrator must delete the old non-SNMP node object and force NNMi to rediscover the new node configurations. See "Delete Nodes" (on page 172).

Tip: At any time, you can initiate an on-demand discovery poll to gather the most current information about a previously discovered device. Select a node and click the **Actions** → **Configuration Poll** command, or use the nnmconfigpoll.ovpl command.

A number of NNMi tools let NNMi administrators control how Spiral Discovery works.

For details about how Spiral Discovery works:

Rediscovery Intervals

Specify how often your entire network is checked for the latest information.

This interval controls how often NNMi generates network traffic to gather the following information:

- Information about the nodes, addresses, and interfaces you configure for discovery.
- Information about Level 2 connectivity between interfaces and VLANs in your network.
- Information about Level 3 connectivity between addresses in your network.

Make sure the interval value provides plenty of time so Spiral Discovery cycles do not overlap. The larger your network environment, the longer the time required to complete a Spiral Discovery cycle.

See "Adjust the Rediscovery Interval" (on page 138) to learn how to set the rediscovery interval.

For details about how Spiral Discovery works:

Discovery Node Name Choices

Control how the **Name** attribute on node forms is populated during discovery. This Name value is used to identify the object in NNMi maps and table views. You specify a hierarchy for discovery to use. You configure three levels in the hierarchy. See "Node Name Decision Tree" (on page 118).

You can designate any of the following for each level of the node Name decision hierarchy:

- **DNS Names**. Discovery uses the results of hostname resolution.

  NNMi follows a set of rules to dynamically generate the value stored in the NNMi database for each Node's Hostname. Click here for details.

  Note: The actual Hostname *might be converted* to all uppercase or all lowercase before it is added to

the NNMi database (depending on how the NNMi administrator configured settings in the `nms-topology.properties` file). See the information about the `nms-topology.properties` file in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

- If the Node supports SNMP, NNMi requests the Hostname using the IP Address of the associated SNMP agent (the Management Address attribute value on the Node form).

  If the NNMi administrator chooses ☑ **Enable SNMP Address Rediscovery** in the Communication Configuration:

  - If the SNMP Agent does not respond, NNMi checks for another Management Address to request the Hostname, and the Hostname could change.
  - If the SNMP Agent associated with the node changes, the Management Address and Hostname could change.

  If the NNMi administrator disables ☐ **Enable SNMP Address Rediscovery** in the Communication Configuration:

  - If the SNMP Agent does not respond, NNMi uses the previously gathered Management Address attribute value to request the Hostname.
  - If the SNMP Agent associated with the node changes, NNMi uses the previously gathered Management Address attribute value to request the Hostname.

- If the Node does not support SNMP, no Management Address is available. NNMi requests a Hostname starting with the lowest IP Address associated with the node (a Discovery Seed value or an IP address value gathered from a neighboring device). NNMi uses the first Hostname provided. The Hostname might change during a future discovery cycle.

- **MIB II sysName Values**. Device administrators set the sysName. Discovery avoids populating the NNMi database with multiple devices having the same manufacturer's default sysName. If a sysName matches or starts with the manufacturer's default factory setting (case-sensitive), discovery ignores sysName as a choice for the Name attribute of the node. NNMi ships with a Device Profile for each device type. The Device Profile includes a record of the manufacturer's default sysName.

  Caution: You can override this choice using the Device Profile's Advanced settings, Never Use sysName attribute. See "Configure Device Profiles" (on page 135) for more information.

- **IP addresses**. The addresses are gathered from discovery seed addresses that you provided, ping sweep configurations, or neighbor addresses gathered using Auto-DiscoveryRules. Discovery avoids potential confusion when a device has multiple IP addresses by following these rules:

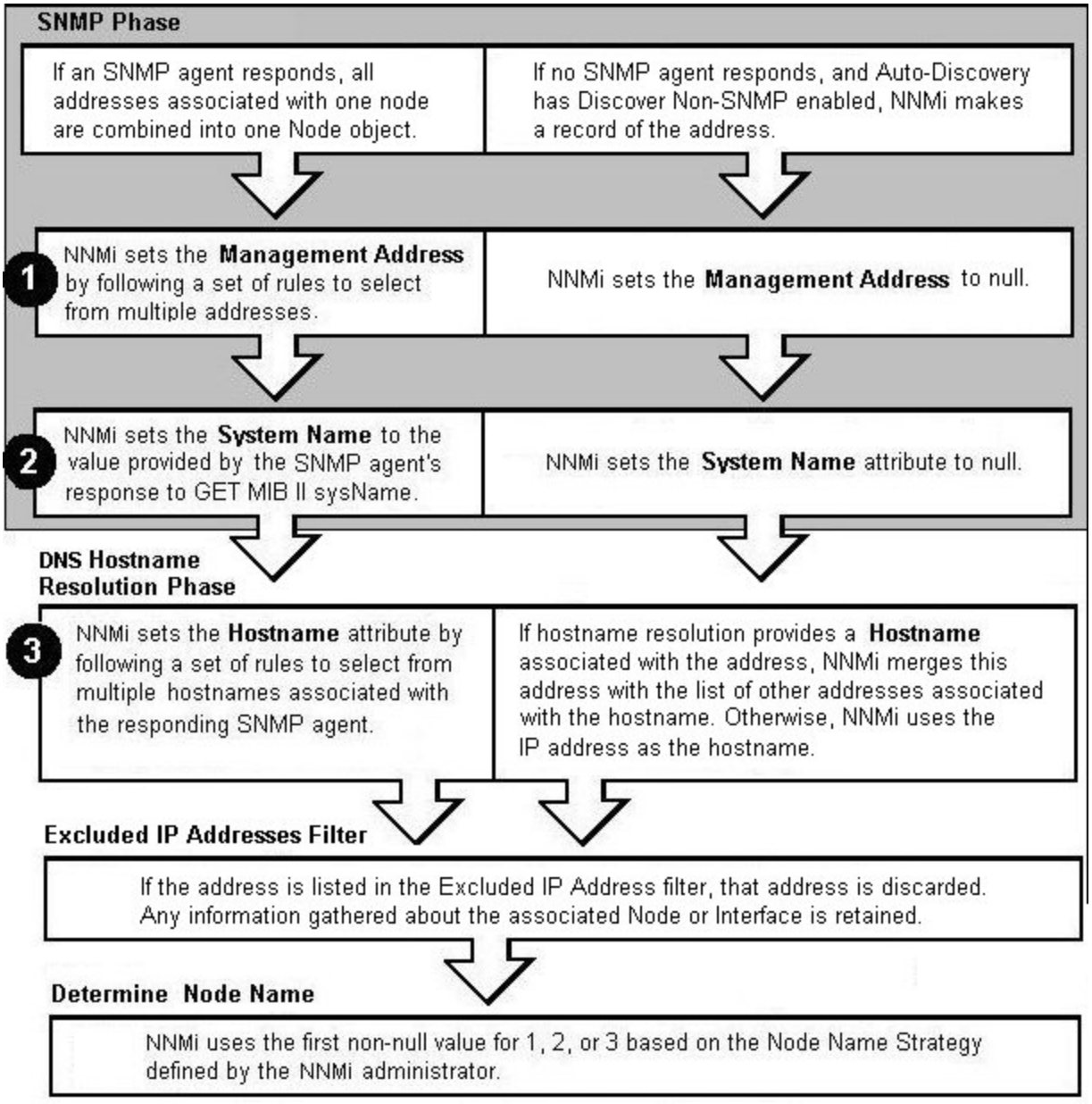
  - If the device supports SNMP, the address of the responding SNMP agent is recorded (the Management Address) and the other addresses are associated with the node. See "Specific Node Settings Form (Communication Settings)" (on page 96) for more information about configuring the management address.
  - If the device does not support SNMP, NNMi queries DNS to determine the hostname. If this hostname matches another non-SNMP node, NNMi merges the information to create only one node with multiple associated addresses.

See "Configure the Node Name Strategy" (on page 140) to learn how to configure the NNMi node name strategy.

For details about how Spiral Discovery works:

Node Name Decision Tree

For each discovered address, NNMi gathers multiple attributes that are used to implement your Node Name strategy. NNMi chooses the node Name based on the Management Address, System Name, and Hostname collected during discovery. The following diagram shows how NNMi determines values for these attributes.



For details about how Spiral Discovery works:

Discovery Seeds (as a starting point)

An optional discovery seed is a specific node that you want NNMi to discover. For example, a discovery seed might be a core router in your management environment.

Each discovery seed is identified by hostname (*not case-sensitive*) or IP address. When you add an optional discovery seed, NNMi immediately tries to discover that device (without waiting until the next regularly scheduled discovery interval). If discovery is not successful, NNMi tries again 10 minutes later, and continues trying. The time between each attempt is doubled until the time reaches 1 week or equals your current discovery interval.

NNMi discovers seed addresses regardless of how you configure Auto-Discovery Rule definitions or the Excluded IP Addresses filter.

Note: Nodes configured as discovery seeds are always discovered and added to the topology database. If you change your mind and delete a discovery seed configuration, the node is not automatically deleted from the topology database. See "Delete Nodes" (on page 172).

If you configure one or more Auto-Discovery Rules, note the following:

- If ☑ **Discover Included Nodes** is enabled for an Auto-Discovery Rule, NNMi uses each discovery seed as a starting point to gather information about neighboring devices to expand discovery.

  Note: You can use the Ping Sweep option in your Auto-Discovery Rules in addition to or instead of Discovery Seeds. Note that with NNMi Advanced, Ping Sweep works only with IPv4 addresses, not IPv6 addresses.

- If ☐ **Discover Included Nodes** is disabled for an Auto-Discovery Rule, no devices matching that rule's criteria are discovered and added to the topology database unless:

  - The device's address is a discovery seed.

    See "Specify Discovery Seeds: Initial Routers or Specific Nodes to be Discovered" (on page 158) to learn how to establish discovery seeds.

  - The device's address is reported as a neighbor to another discovered address.

    If you want to ensure that an address is never added to the NNMi database, use the "Configure an Excluded IP Addresses Filter" (on page 155) or "Configure an Excluded Interfaces Filter" (on page 157) settings.

For details about how Spiral Discovery works:

Ping Sweep (as a starting point)

You have two choices for Auto-Discovery starting points. Use either or both to best advantage in your network environment:

- Discovery Seeds
  You designate specific hostnames (*not case-sensitive*) or IP addresses where Auto-Discovery starts gathering neighbor information.
- Ping Sweep
  NNMi issues ICMP pings to certain addresses gathered from neighbor information.

  Note: NNMi Advanced- Ping Sweep works only with IPv4 addresses, not IPv6 addresses.

Ping Sweep sends ICMP ping commands to IP addresses in the ranges defined in your Auto-Discovery rules. Ping Sweep enforces the following limits to the ICMP pings:

- For each specific IP address range, NNMi issues pings across a maximum of the last two octets in the IPv4 address range. This is equivalent to a /16 subnet
- ICMP pings are limited to 500 at one time. This avoids flooding your network or tripping spam detection tools.

Ping Sweep is useful in wide area networks such as ATM, Frame Relay, and Point-to-Point that do not contain an Address Resolution Protocol (ARP) cache.

You configure the Ping Sweep feature at two levels:

- "Configure Ping Sweep Global Settings" (on page 139)
- "IP Address Ranges for Auto-Discovery" (on page 146) (Ping Sweep configuration for each rule)

For details about how Spiral Discovery works:

Auto-Discovery Rules

Auto-Discovery Rules control the extent of automatic discovery. You choose the starting points for automatic discovery (either Discovery Seeds or Ping Sweep, or both).

- If ☐ **Discover Included Nodes** is disabled for a particular Auto-Discovery Rule, nodes that match the Rule criteria are affected as follows:
  - *IP Address* ranges are not used for gathering neighbor information, see "Limit Sources of Neighbor Information" (on page 133).
  - *System Object ID* ranges are excluded from discovery. For examples, see "Specific System Object IDs Not Discovered" (on page 134) .
- If ☑ **Discover Included Nodes** is enabled for a particular Auto-Discovery Rule, a variety of protocols are used to gather information about the neighbors adjacent to each discovered device. Spiral Discovery then requests neighbor information from each new neighbor. This sequence continues until the boundaries identified by your rule definition are reached.

See "Configure Auto-Discovery Rules" (on page 141) to learn how to establish the rules that control automatic discovery.

When defining Auto-Discovery Rules, you must provide *at least one* Auto-Discovery Rule that includes an IP address range to define the limits of your management domain. By default NNMi discovers routers and switches. You can expand the number of device types that NNMi discovers by enabling ☑**Discover Any SNMP Device** and including one or more System Object ID Ranges (based on MIB II sysObjectID values). Your address ranges and system object ID ranges determine which discovered addresses are added to the NNMi database.

Note: NNMi also uses the source IP address from SNMP traps as hints to discovery.

NNMi gathers information about neighboring devices using ARP cache, DNS, and the following protocols:

- **BGP** — Border Gateway Protocol
- **CDP** — Cisco Discovery Protocol
- **EIGRP** — Cisco Enhanced Interior Gateway Routing Protocol
- **ENDP** — Enterasys Discovery Protocol (also known as CDP - Cabletron Discovery Protocol)

- **FDP** — Foundry Discovery Protocol
- **OSPF** — Open Shortest Path First

In Wide Area Networks (WANs) such as ATM, Frame Relay, and Point-to-Point (where ARP cache is not available), the optional Ping Sweep (IPv4-only) feature locates nodes for Auto-Discovery to use when gathering neighbor information and evaluating subnet connection rules (IPv4-only).

For details about how Spiral Discovery works:

Filters to Exclude Certain IP Addresses from Discovery

When configuring Spiral Discovery in NNMi, sometimes it is useful to exclude certain addresses or ranges of addresses from discovery and monitoring. For example:

- There are multiple Nortel switches in your environment. They each have a non-routable IP address of 192.168.168.168 that is defined by the manufacturer. This special address is used to establish the default VLAN for the switch. However, NNMi discovers this duplicate address and establishes a lot of unnecessary connections on the Layer 3 Neighbor View map.
- Your service provider forbids the generation of ICMP or SNMP traffic from your NNMi installation. That range of addresses can easily be excluded to prevent violating your contractual agreement with the vendor.
- The Provider Edge (PE) routers have addresses that NNMi ICMP ping commands cannot reach or have addresses that you want to exclude from Subnet views.

Note: The node and interface associated with any address identified in your Excluded IP Address filter shows up in the topology database and maps.

Carefully select the addresses for your Excluded IP Addresses filter. Do not populate the Excluded IP Addresses filter with the addresses associated with SNMPv1/SNMPv2c agents or SNMPv3 engines (the Management Addresses). See "Configure an Excluded IP Addresses Filter" (on page 155) to learn how to exclude an address or range of addresses from discovery.

For details about how Spiral Discovery works:

Filters to Exclude Certain Interfaces from Discovery

When configuring Spiral Discovery in NNMi, sometimes it is useful to exclude certain interfaces or interface types from discovery and monitoring.

Once configured as an excluded interface:

- The interface's membership status in groups is removed:
  - Connection
  - Router Redundancy Group (*NNMi Advanced*)
  - Port Aggregation (*NNMi Advanced*)
  - Link Aggregation
  - VLAN
- The interface's relationship to other objects is canceled:
  - Node
  - Address

  - VLAN Port
- During the next discovery cycle, NNMi automatically removes any previously discovered data associated with an excluded interface.

Note: The node associated with any interface identified in your Excluded Interface filter still shows up in the topology database and maps.

See "Configure an Excluded Interfaces Filter" (on page 157) to learn how to exclude certain interfaces or interface types from discovery.

For details about how Spiral Discovery works:

Subnet Connection Rules

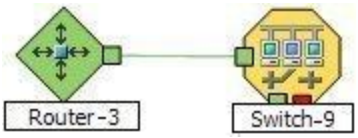
The NNMi Subnet Connection Rules work only with IPv4 subnets.

Sometimes it is useful to monitor connections in the following categories:

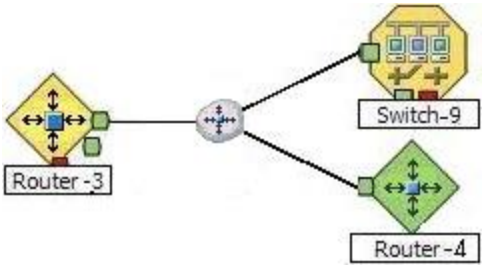
- Virtual IPv4 tunnel connections within your management domain.
- Connections to remote sites (across a Service Provider's network or a WAN).

NNMi accomplishes this by following special rules for subnets with prefix lengths between 28 and 31. These special rules are called Subnet Connection Rules.

These Subnet Connections Rules enable NNMi to draw arbitrary connections on maps where none would otherwise be detected. If the connection is between two nodes, NNMi draws a standard line on maps. For example:



If the connection is between more than two nodes, NNMi displays an icon:



If you double-click the line or the icon, the Layer 2 Connection form displays and the **Topology Source** value is `SUBNETCONNECTION`.

NNMiprovides a group of predefined Subnet Connection Rules (see "Subnet Connection Rules Provided by NNMi" (on page 154)). You can edit an existing Subnet Connection Rule or create your own (see "Configure Subnet Connection Rules" (on page 152)).

If you limit Spiral Discovery to only your Discovery Seeds, NNMi uses the Subnet Connection Rules to detect connections among those devices.

If you use Auto-Discovery rules to configure Spiral Discovery, when NNMi detects a subnet prefix between 28 and 31, NNMi uses the Subnet Connection Rules:

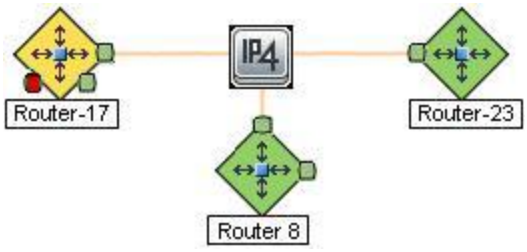
1. NNMi checks for an applicable Subnet Connection Rule (see "Subnet Connection Rules Provided by NNMi" (on page 154)).
2. If a match is found, Spiral Discovery checks the topology database for existing data about each IPv4 address in the subnet. If no data is found for a particular IPv4 address, NNMi issues an SNMP query to the new IPv4 address. The number of available IPv4 addresses for each valid prefix length is described in the following table:

   Valid Minimum Prefix Length Values (Subnet Mask Length)

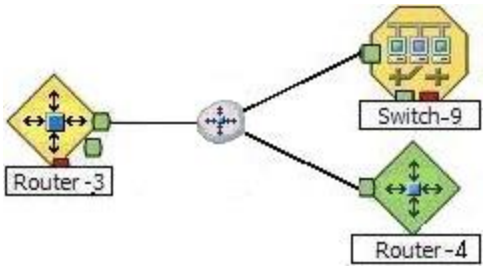
Valid Minimum IPv4 Prefix Length Values	Number of Usable IPv4 Addresses
28	14 (16-2=14)*
29	6 (8-2=6)*
30	2 (4-2=2)*
31	2

   * Two IPv4 addresses are reserved in each subnet. The first IPv4 address is used for the network itself and the last IPv4 address is reserved for broadcast.
3. NNMi checks the Excluded IP Addresses list. Any addresses in the list are dropped. For details, see "Filters to Exclude Certain IP Addresses from Discovery" (on page 121).
4. New IPv4 addresses that respond to SNMP are added to the topology database and available for monitoring purposes. New IPv4 addresses that do not respond to SNMP are ignored.
5. If the IPv4 address on each end of a connection has an associated interface, NNMi uses the subnet connection rule to display the connection on map views.

   In a Layer 3 Neighbor View map, if NNMi discovers an interface that is connected to more than one interface, the results of your subnet connection rule look like the following:



In a Layer 2 Neighbor View map, if NNMi discovers an interface that is connected to more than one interface, the results of your subnet connection rule look like the following:



See "Configure Subnet Connection Rules" (on page 152) to learn how to configure Subnet Connection Rules.

For details about how Spiral Discovery works:

Device Profiles and Discovery

You can modify the settings in the Device Profiles to fine-tune Spiral Discovery and the device symbols on the maps.

You can also use the **Configuration** → **Device Profiles** view to see the list of all known system object IDs (MIBII sysObjectIDs) at the time NNMi released. This list of system object IDs is useful if you want to expand the range of devices that NNMi discovers. By default, NNMi discovers only routers and switches (see "SNMP System Object ID Ranges for Discovery" (on page 150)).

See the Advanced Settings section of "Configure Device Profiles" (on page 135) for more information.

For details about how Spiral Discovery works:

Prerequisites for Discovery

NNMi uses SNMP and DNS while discovering and monitoring devices in your network environment. To ensure accurate network topology information, verify that these prerequisites are working properly:

- "SNMP Prerequisites" (on page 124)
- "Well-Configured DNS Prerequisite" (on page 125)
- "IPv6 Addresses Prerequisite (NNMi Advanced)" (on page 126)

SNMP Prerequisites

Spiral Discovery uses SNMP while detecting devices and connections among the devices in your network environment. NNMi also uses SNMP as part of monitoring and reporting on the health of devices in your network environment.

NNMi supports the following SNMP versions:

- SNMPv1
- SNMPv2c
- SNMPv3

NNMi uses information gathered from Routers to establish subnet membership for Layer 3 connections. Make sure that important Routers in your network environment are SNMP enabled.

NNMi uses either of the following criteria to identify a Router:

- Each Router responds with appropriate values for sysServices (1.3.6.1.2.1.1.7) and ipForwarding (1.3.6.1.2.1.4.1). See RFC 1213-MIB for details.
- Identify Routers with the Device Profile configuration settings for a particular device type.

You must provide the appropriate SNMP Community Strings to NNMi. See "Configuring Communication Protocol" (on page 66).

Before configuring NNMi discovery, complete the following steps:

1. Enable SNMP communication on important devices in your network (each device that you want NNMi to actively monitor).

   See the manufacturer's documentation for information about how to configure SNMP on each of your devices.

   - Establish *read community strings* for any SNMPv1 or SNMPv2c agents.
   - Establish the appropriate *User-based Security Module (USM) level of security for authentication and privacy* for any SNMPv3 agents.
2. Configure NNMi to use the appropriate *read community strings* or *USM settings* for your network environment. See "Configuring Communication Protocol" (on page 66).

Well-Configured DNS Prerequisite

NNMi uses Domain Name System (DNS) to determine relationships between hostnames and IP addresses. This can result in a large number of `nslookup` requests.

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

Use nslookup to Verify DNS Server Configurations

Verify that your DNS servers are well configured to prevent long delays when resolving `nslookup` requests. This means the DNS server responding to NNMi `nslookup` requests has these qualities:

- The DNS server is an authoritative server and does not forward DNS requests.
- The DNS server has consistent *hostname-to-IP address* mappings and *IP address-to-hostname* mappings.
- If your network uses multiple DNS servers, all respond consistently to any particular `nslookup` request.

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

On the NNMi management server, verify that the following configuration settings in your environment:

- **All operating systems**: Locate your `*/etc/hosts` file and ensure that the host file contains a minimum of two entries. When an `nslookup` command is not successful, this file takes over:

  `127.0.0.1` (loopback loghost) or `::1`
  `<NNMi_server_address>` (the IP address of the NNMi management server)

If your NNMi management server participates in a high availability (HA) environment, the virtual server name and IP-address is required in the `*/etc/hosts` file in addition to the physical server name and IP-address.

Windows: The following registry key determines the location of this file:

```
\HKEY_LOCAL_
   MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\DataBasePath
```

UNIX: This file is in the `/etc` directory.

- **Windows:** Use the Control Panel to navigate to your Network and Internet Connections configuration, Network Connections, Local Area Connections, Support tab, and click the Details button. Verify that all identified DNS servers provide consistent *hostname-to-IP address* mappings and *IP address-to-hostname* mappings.
- **UNIX:** Ensure that the `nslookup` search path resolves to the `nsswitch.conf` file. See the *nsswitch.conf(4)* manpage that was provided with your operating system. Verify that all identified DNS servers provide consistent *hostname-to-IP address* mappings and *IP address-to-hostname* mappings.

Exclude Problem Devices from nmlookup

You can populate two files that instruct `nslookup` to exclude certain addresses. The benefits of doing this are as follows:

- Speed up Spiral Discovery.
- Keep network traffic generated by NNMi to a minimum.

If you know there are problems with the DNS configuration in your network domain (hostnames or addresses that do not resolve properly), instruct NNMi to avoid `nslookup` requests for unimportant devices.

To identify problem devices, create the following two files prior to configuring NNMi discovery. NNMi never issues a DNS request for hostnames or IP addresses identified in these files:

- hostnolookup.conf — Enter fully-qualified hostnames or wildcards that identify groups of hostnames.
- ipnolookup.conf — Enter fully-qualified IP addresses or wildcards that identify groups of IP addresses.

Use an ASCII editor to populate the files. Place the files in the following location on the NNMi management server:

- **Windows**:
  `%NnmDataDir%\shared\nnm\conf\`
- **UNIX**:
  `/var/opt/OV/shared/nnm/conf/`

IPv6 Addresses Prerequisite (*NNMi Advanced*)

To discover and monitor Both IPv4 and IPv6 IP addresses, the settings in the `nms-jboss.properties` file must first be configured.

NNMi Advanced. If the NNMi administrator wants NNMi to access and monitor IPv6 addresses, NNMi must be configured to do so. For more information, see the "Configuring NNMi Advanced for IPv6" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`. One of the configuration steps explains how to make changes to the `nms-jboss.properties` file. Settings in the `nms-jboss.properties` file also partially control how NNMi determines the Management Address for each node:

- Prefer IPv4
- Prefer IPv6
- No Preference (either IPv4 or IPv6)

To check NNMi for the status of the IPv6 feature in your networking environment, click **Help** → **System Information** and navigate to the **Server** tab.

In the Management Server section, locate the following attributes and their current values:

- **IPv6 Address:** Displayed if the NNMi management server has an IPv6 address.
- **IPv6 Management**:
  - Enabled
  - Disabled (see the *HP Network Node Manager i Software Deployment Reference*)
  - Not Licensed (requires NNMi Advanced)
- **IPv6 Communication**:

  Displayed if NNMi IPv6 management is enabled.
  - Enabled
  - Disabled (see the *HP Network Node Manager i Software Deployment Reference*)
  - Not available (no IPv6 Address on management server)

Determine Your Approach to Discovery

Discover and monitor only the network devices that you and your team consider to be important. Take any approach that makes sense to you.

Tip: See the following examples for ideas. Print one or more of the following topics to use as a guide when you are configuring NNMi discovery.

Maintain absolute control over what is discovered.

- "Do Not Use Auto-Discovery Rules" (on page 128)

Configure Spiral Discovery to make decisions about what is discovered.

Create one or more Auto-Discovery Rules that define the boundaries of what is important to you and your team:

- "Routers and Switches Discovered" (on page 128) (Auto-Discovery Rules default behavior)
- "All SNMP Devices Discovered" (on page 129) (more than Routers and Switches)
- "Everything Discovered" (on page 130) (all SNMP enabled devices and all Non-SNMP devices)
- "All Devices from a Specific Vendor Discovered" (on page 132)

Fine tune Spiral Discovery behavior.

Identify the things your team is not interested in monitoring:

- "Limit Sources of Neighbor Information" (on page 133)
- "Exclude Problem IP Addresses from Discovery" (on page 134)

- "Exclude Problem Interfaces from Discovery" (on page 134)
- "Specific System Object IDs Not Discovered" (on page 134)

Do Not Use Auto-Discovery Rules

If you want NNMi to discover only what you specify, use these guidelines.

Note: After you set your configuration according to these guidelines, when a new device is added to your network, NNMi does not discover that device unless you configure another discovery seed to identify that device.

Configuration Steps to Discover Only What You Specify

Task	How
Do not include any **Auto-Discovery Rules**.	None are required for this strategy.
In the **Discovery Seeds** settings, designate the hostname (*not case-sensitive*) or IP address of each device you want NNMi to discover and configure NNMi to monitor your SNMP devices. See "Monitoring Network Health" (on page 215).	"In the Console, Configure Discovery Seeds " (on page 159)

Note: You control how often Spiral Discovery checks the discovered nodes based on a **Rediscovery Interval** setting. See "Adjust the Rediscovery Interval" (on page 138) for more information.

Routers and Switches Discovered

If you want Spiral Discovery to automatically find devices on your network, use these guidelines. By default, Auto-Discovery Rules apply only to routers and switches. If you want to discover more devices, see "All SNMP Devices Discovered" (on page 129) or "Everything Discovered" (on page 130).

Note: After you set your configuration according to these guidelines, when a new router or switch is added to your network, you do not need to do anything. NNMi discovers it during the next discovery cycle.

Configuration Steps to Discover Only Routers and Switches

Task	How
Create an **Auto-Discovery Rule**. Set the following attribute values: • Enter **Ordering** 500 It is recommended that you use **Ordering** number 500. This provides flexibility if you want to expand or limit Spiral Discovery behavior later by creating additional rules with higher or lower Ordering numbers. • Enable ☑ **Discover Included Nodes** • Disable ☐ **Discover Any SNMP Device** • Disable ☐ **Discover Non-SNMP Devices**	"Configure Auto-Discovery Rules" (on page 141)
Create one or more **IP Ranges** settings that identify the entire scope of the addresses in your network domain: Enter **IP Range** <IPv4/IPv6 range> (at least one)	"IP Address Ranges for Auto-Discovery" (on page 146)

Task	How
Set **Range Type** Include in rule	
Optional. NNMi can use **Enable Ping Sweep** (instead of or in addition to discovery seeds, see below) to gather neighbor information.	"Ping Sweep (as a starting point)" (on page 119) NNMi Advanced. Ping Sweep works only with IPv4 addresses, not IPv6 addresses.
If you want Spiral Discovery to find all routers and switches. Do not create any **System Object ID Ranges**. If you want to limit Spiral Discovery to only the vendor/make/model of routers and switches that you specify, create one or more **System Object ID Ranges**. Your list *must include everything* you want Spiral Discovery to find.	"SNMP System Object ID Ranges for Discovery" (on page 150) **Tip**: Navigate to the **Configuration** workspace, and select the **Device Profiles** view to see all known system object IDs at the time NNMi released.
Optional. In the **Discovery Seeds** settings, designate one or more hostnames (*not case-sensitive*) or IP addresses. Consider using devices with the largest neighbor data. For example, a good choice would be a core router. The discovery seeds become the starting points from which Spiral Discovery explores your network.	"In the Console, Configure Discovery Seeds " (on page 159)

Note: You control how often Spiral Discovery runs by designating the **Rediscovery Interval** setting. See "Adjust the Rediscovery Interval" (on page 138) for more information.

If you want to fine tune the Spiral Discovery results, see:

- "All Devices from a Specific Vendor Discovered" (on page 132) (more than routers and switches from the vendor)
- "Limit Sources of Neighbor Information" (on page 133) (less than all routers and switches)
- "Specific System Object IDs Not Discovered" (on page 134) (less than all routers and switches)
- "Exclude Problem IP Addresses from Discovery" (on page 134)

All SNMP Devices Discovered

If you want Spiral Discovery to find any device that has a working SNMP agent, use these guidelines. However, this strategy may cause you to reach your license limit very quickly. Consider defining additional Auto-Discovery Rules to limit this strategy. (See "Specific System Object IDs Not Discovered" (on page 134), or "Limit Sources of Neighbor Information" (on page 133). See also "Filters to Exclude Certain IP Addresses from Discovery" (on page 121)).

Note: When a new SNMP-supported device is added to your network, you do not need to do anything. NNMi discovers it during the next discovery cycle.

Configuration Steps to Discover All Devices that Have SNMP Agents

Task	How
Create an **Auto-Discovery Rule**. Set the following attribute values: • Enter **Ordering** 500 It is recommended that you use **Ordering** number 500. This provides flexibility if you want to expand or limit Spiral Discovery behavior later by creating additional rules with higher or lower Ordering numbers. • Enable ☑ **Discover Included Nodes** • Enable ☑ **Discover Any SNMP Device** • Disable ☐ **Discover Non-SNMP Devices** **Note**: This strategy may cause you to reach your licensed capacity very quickly. See "Extend a Licensed Capacity" (on page 1020).	"Configure Auto-Discovery Rules" (on page 141)
Create one or more **IP Range** settings that identify the entire scope of the addresses in your network domain: Enter **IP Range** <IPv4/IPv6 range> (at least one) Set **Range Type** Include in rule	"IP Address Ranges for Auto-Discovery" (on page 146)
Optional. NNMi can use ☑ **Enable Ping Sweep** (instead of or in addition to discovery seeds, see below) to gather neighbor information.	"Ping Sweep (as a starting point)" (on page 119) NNMi Advanced. Ping Sweep works only with IPv4 addresses, not IPv6 addresses.
Do not create any **System Object ID Ranges**. When Discover Any SNMP Device is enabled and no ranges are specified, *all SNMP devices* are discovered (every sysObjectID that responds to an SNMP query).	
Optional. In the **Discovery Seeds** settings, designate one or more hostnames (*not case-sensitive*) or IP addresses. Consider using devices with the largest neighbor data. For example, a good choice would be a core router. The discovery seeds become the starting points for Spiral Discovery.	"In the Console, Configure Discovery Seeds " (on page 159)

Note: You control how often Spiral Discovery runs by designating the **Rediscovery Interval** setting. "Adjust the Rediscovery Interval" (on page 138) for more information.

Everything Discovered

If you want Spiral Discovery to find all devices in your network, use these guidelines. However, this strategy may cause you to reach your licensed capacity very quickly. Consider defining additional Auto-Discovery Rules to limit this strategy. (See "All Devices from a Specific Vendor Discovered" (on page 132), "Specific System Object IDs Not Discovered" (on page 134), or "Limit Sources of Neighbor Information" (on page 133). See also "Filters to Exclude Certain IP Addresses from Discovery" (on page 121)).

If the device does not support SNMP, NNMi queries DNS to determine the hostname. If this hostname matches another non-SNMP node, NNMi merges the information to create only one node with multiple associated addresses to preserve licensed capacity limits for discovered nodes.

Note: After you set your configuration according to these guidelines, when a new device is added to your network, you do not need to do anything. NNMi discovers it during the next discovery cycle.

Configuration Steps to Discover Everything

Task	How
Create an **Auto-Discovery Rule**. Set the following attribute values: • Enter **Ordering** 500 It is recommended that you use **Ordering** number 500. This provides flexibility if you want to expand or limit Spiral Discovery behavior later by creating additional rules with higher or lower Ordering numbers. • Enable ☑ **Discover Included Nodes** • Enable ☑ **Discover Any SNMP Device** • Enable ☑ **Discover Non-SNMP Devices** **Note**: This strategy may cause you to reach your licensed capacity very quickly. See "Extend a Licensed Capacity" (on page 1020). Consider adding your non-SNMP devices using seeds instead of selecting the **Discover Non-SNMP Devices** option.	"Configure Auto-Discovery Rules" (on page 141)
Create one or more **IP Ranges** settings that identify the entire scope of the addresses in your network domain: Enter **IP Range** <IPv4/IPv6 range> (at least one is required) Set **Range Type** Include in rule	"IP Address Ranges for Auto-Discovery" (on page 146)
Optional. NNMi can use ☑ **Enable Ping Sweep** (instead of or in addition to discovery seeds, see below) to gather neighbor information.	"Ping Sweep (as a starting point)" (on page 119) NNMi Advanced. Ping Sweep works only with IPv4 addresses, not IPv6 addresses.
Do not include any **System Object ID Ranges**. When Discover All SNMP Devices is enabled and no ranges are specified, *all SNMP devices* are discovered (every sysObjectID that responds to an SNMP query).	
Optional. In the **Discovery Seeds** settings, designate one or more hostnames (*not case-sensitive*) or IP addresses. Consider using devices with the largest neighbor data. For example, a good choice would be a core router. The discovery seeds become the starting points for Spiral Discovery.	"In the Console, Configure Discovery Seeds " (on page 159)

Note: You control how often Spiral Discovery runs by designating the **Rediscovery Interval** setting. See "Adjust the Rediscovery Interval" (on page 138) for more information.

All Devices from a Specific Vendor Discovered

If you want to expand Spiral Discovery to all devices manufactured by a specific vendor (more than routers and switches), use these guidelines.

To implement this strategy, you must create two Auto-Discovery Rules (see prerequisite). The prerequisite rule configures Spiral Discovery to find any router or switch, regardless of vendor.

Note: After you set your configuration according to these guidelines, when a new device is added to your network, you do not need to do anything. NNMi discovers or skips devices according to your configuration choices.

Prerequisite: Create an Auto-Discovery Rule that uses IP address ranges to specify the outer limits of your network management domain. This rule instructs Spiral Discovery to find devices manufactured by *any* vendor. See "Routers and Switches Discovered" (on page 128), "All SNMP Devices Discovered" (on page 129) or "Everything Discovered" (on page 130).

Configuration Steps to Discover All Devices from Specific Vendors

Task	How
Create an **Auto-Discovery Rule**. Set the following attribute values: • Enter **Ordering** 400 It is recommended that you use **Ordering** number 400. This rule must have a lower number than the prerequisite rule. • Enable ☑ **Discover Included Nodes** • Enable ☑ **Discover Any SNMP Device** • Disable ☐ **Discover Non-SNMP Devices**	"Configure Auto-Discovery Rules" (on page 141)
Create one or more **IP Ranges** settings that identify the entire scope of the addresses in your network domain: Enter **IP Range** <IPv4/IPv6 range> (at least one) Set **Range Type** Include in rule	"IP Address Ranges for Auto-Discovery" (on page 146)
When Discover Any SNMP Devices is enabled and you specify one or more System Object ID Ranges, *only* the sysObjectIDs you specify are discovered. Create one or more **System Object ID Ranges** settings. Enter the SNMP sysObjectID prefix that identifies each vendor for the devices you want to discover. For example, to include all HP devices, use the following prefix: `1.3.6.1.4.1.11.` Enter **System Object ID Prefix** <sysObjectID> Set **Range Type** Include in rule	"SNMP System Object ID Ranges for Discovery" (on page 150) **Tip**: Navigate to the **Configuration** workspace, and select the **Device Profiles** view to see all known system object IDs at the time NNMi released.

Note:You control how often Spiral Discovery runs by designating the **Rediscovery Interval** setting. "Adjust the Rediscovery Interval" (on page 138) for more information.

Limit Sources of Neighbor Information

If you want Auto-Discovery to never request neighbor information from certain addresses within your management domain, use these guidelines. In other words, Auto-Discovery will not use these addresses as resources for further discovery.

Note: The addresses identified in your IP address Range might show up in the topology database if their neighbors provide information about them.

To implement this strategy, you must create two Auto-Discovery Rules (see prerequisite).

Prerequisite: Create an Auto-Discovery Rule that uses IP address ranges to specify the outer limits of your network management domain. See "Routers and Switches Discovered" (on page 128), "All SNMP Devices Discovered" (on page 129) or "Everything Discovered" (on page 130).

Configuration Steps to Exclude Some IP addresses from Providing Neighbor Data

Task	How To
Create an **Auto-Discovery Rule**. Set the following attribute values: • Enter **Ordering** 100 It is recommended that you use **Ordering** number 100. This rule must have a lower number than the prerequisite rule. • Disable ☐ **Discover Included Nodes** • Disable ☐ **Discover Any SNMP Device** • Disable ☐ **Discover Non-SNMP Devices** This strategy instructs NNMi to "not gather neighbor information " from certain addresses. No devices matching that rule's criteria are discovered and added to the topology database unless: • The device's address is a discovery seed. See "Specify Discovery Seeds: Initial Routers or Specific Nodes to be Discovered" (on page 158) to learn how to establish discovery seeds. • The device's address is reported as a neighbor to another discovered address. If you want to ensure that an address is never added to the NNMi database, use the "Configure an Excluded IP Addresses Filter" (on page 155) or "Configure an Excluded Interfaces Filter" (on page 157) settings.	"Configure Auto-Discovery Rules" (on page 141)
Create one or more **IP Ranges** settings that identify all addresses from which Auto-Discovery never requests neighbor information. Enter **IP Range** <IPv4/IPv6 range> (at least one) Set **Range Type** Include in rule	"IP Address Ranges for Auto-Discovery" (on page 146)
Create a list of IP addresses that NNMi should never discover.	"Configure an Excluded IP Addresses Filter" (on

Task	How To
	page 155)

Note: You control how often Spiral Discovery runs by designating the **Rediscovery Interval** setting. See "Adjust the Rediscovery Interval" (on page 138) for more information.

Exclude Problem IP Addresses from Discovery

If you want Spiral Discovery to never discover certain IP addresses, use these guidelines.

Note: The node and interface associated with any address identified in your Excluded IP Address filter are added to the topology database and maps.

Configuration Steps to Exclude Certain IP Addresses from Spiral Discovery

Task	How To
Create at least one **Excluded IP Addresses** filter.	"Configure an Excluded IP Addresses Filter" (on page 155)

Note: You control how often Spiral Discovery runs by designating the **Rediscovery Interval** setting. See "Adjust the Rediscovery Interval" (on page 138) for more information.

Exclude Problem Interfaces from Discovery

If you want Spiral Discovery to never discover certain interfaces, use these guidelines.

Note: The node associated with any interface identified in your Excluded Interfaces filter is added to the topology database and maps.

Configuration Steps to Exclude Certain Interfaces from Spiral Discovery

Task	How To
Create at least one **Excluded Interface** filter.	"Configure an Excluded Interfaces Filter" (on page 157)

Note: You control how often Spiral Discovery runs by designating the **Rediscovery Interval** setting. See "Adjust the Rediscovery Interval" (on page 138) for more information.

Specific System Object IDs Not Discovered

If you want to limit Spiral Discovery to never discover certain device makes/models, use these guidelines. You must be able to identify the devices using SNMP system object IDs.

To implement this strategy, you must create two Auto-Discovery Rules (see prerequisite).

Note: After you set your configuration according to these guidelines, when a new device is added to your network, you do not need to do anything. NNMi discovers or skips devices according to your configuration choices.

Prerequisite: Create an Auto-Discovery Rule that uses IP address ranges to specify the outer limits of your network management domain. See "Routers and Switches Discovered" (on page 128), "All SNMP Devices Discovered" (on page 129) or "Everything Discovered" (on page 130).

Configuration Steps to Exclude Some System Object IDs from Spiral Discovery

Task	How To
Create an **Auto-Discovery Rule**. Set the following attribute values: • Enter **Ordering** 200 It is recommended that you use **Ordering** number 200. This rule must have a lower number than the prerequisite rule. • Disable ☐ **Discover Included Nodes** • Disable ☐ **Discover Any SNMP Device** • Disable ☐ **Discover Non-SNMP Devices** **Note**: This strategy instructs NNMi to "not discover" certain things.	"Configure Auto-Discovery Rules" (on page 141)
Do not include any **IP Ranges**. When no IP address ranges are defined within an Auto-Discovery Rule, your system object ID ranges take precedence over all Auto-Discovery Rules that have higher Ordering numbers than this Auto-Discovery Rule. **Note**: NNMi automatically treats any Auto-Discovery Rules without any IP Range as if ☐ **Discover Included Nodes** were disabled.	
Create one or more **System Object ID Ranges** settings. Enter the SNMP sysObjectID that identifies the make/model of the SNMP device that you do not want to discover. Enter **System Object ID Prefix** <sysObjectID> Set **Range Type** Include in rule	"SNMP System Object ID Ranges for Discovery" (on page 150) **Tip**: Navigate to the **Configuration** workspace, and select the **Device Profiles** view to see all known system object IDs at the time NNMi released.

Note: You control how often Spiral Discovery runs by designating the **Rediscovery Interval** setting. See "Adjust the Rediscovery Interval" (on page 138) for more information.

Configure Device Profiles

According to industry standards (RFC 1213, MIB II), each combination of vendor, device type, and model number is assigned a unique SNMP system object ID (sysObjectID). For example, all Cisco 6500 series switches have the same sysObjectID prefix: `.1.3.6.1.4.1.9.*`

HP provides well over three thousand preconfigured Device Profiles, one for each known sysObjectID at the time NNMi released.

NNMi uses Device Profiles (which equate to sysObjectIDs) to control certain types of behavior:

- Spiral Discovery determines the closest matching device profile, and uses the device profile settings to control certain attribute values for the discovered device. The Device Profile also influences the following:
  - Auto-Discovery Rules can provide a list of sysObjectIDs that expand the default discovery behavior (beyond routers and switches) or prevent troublesome device types from being discovered.

- The Node Name value might be affected, depending on your choices, see "Configure the Node Name Strategy" (on page 140).

- When Node Groups are defined based on system object IDs, the State Poller Service monitors devices based on attribute values in the device profiles. Device Profile settings determine how State Poller detects renumbered interfaces.
- In Map views, the background shape of map icons is determined by the Device Category. See About Map Symbols for an example of each available shape. There is also a Force Device attribute that enables category overrides in troublesome situations.

Tip: To quickly locate the device profile settings for a particular network device, sort or filter the Device Profiles view by clicking the heading for the Device Vendor, Device Model, or Device Category columns.

To access the device profile definition for a particular device type:

1. Navigate to the **Device Profile** view.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Device Profiles** view.
2. Do one of the following:
   - To create a device profile, click the New icon.
   - To edit a device profile, click the Open icon in the row representing the configuration you want to edit.
3. Modify the settings as needed:

   Caution: When you make a change, NNMi must update all references to device profiles. This can take some time and slow down your system. Consider making this change during a slow time in your network environment.

   - The basic settings Device Category attribute value modifies NNMi behavior for Spiral Discovery and map symbols.

     Caution: If you make changes to a Menu Item provided by NNMi, those changes are at risk of being overwritten in the future. See Author form for important information.

   - The advanced settings control NNMi behavior for Spiral Discovery and Node name selection. For example, instruct NNMi to treat a certain device type as a Router.
4. Click **Save and Close**. NNMi applies your changes during the next regularly scheduled discovery cycle. To apply the changes immediately, use **Actions** → **Configuration Poll**. See Using Actions to Perform Tasks for more information.

Configure Discovery

NNMi uses Simple Network Management Protocol (SNMP read-only queries), and a variety of communication protocols to discover the devices within the network management domain that you define. See "How Spiral Discovery Works" (on page 114) for more information.

By default, NNMi does the following (and you can change these default settings):

- Drops all non-SNMP nodes from discovery.
- Discovers routers and switches.

If you want NNMi to discover non-SNMP devices or more than routers and switches, use Auto-Discovery Rules that configure discovery behavior to meet your needs. Or add the specific devices as discovery seeds.

To configure the NNMi Discovery Process, do the following:

1. Complete all prerequisites. See "Prerequisites for Discovery" (on page 124), .
2. Navigate to the **Discovery Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Discovery Configuration.**
3. Make your configuration choices (see table).
4. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval. To apply the changes immediately, use **Actions → Configuration Poll**. See Using Actions to Perform Tasks for more information.

Discovery Configuration Tasks

Task	How
"Determine Your Approach to Discovery" (on page 127)	Read these guidelines to understand how to configure discovery for the types of devices you want to discover, including the following: For strategies to discover devices: For strategies to prevent specific devices from being discovered:
"Adjust the Rediscovery Interval" (on page 138)	*Optional*. Use the **Discovery Configuration** workspace to modify the global discovery interval setting. The Global Control setting for Rediscovery Interval controls the frequency that NNMi uses for network discovery traffic.
"Configure Whether to Delete Unresponsive Nodes" (on page 138)	[*Optional*]. Use the **Discovery Configuration** workspace to determine whether to delete nodes from the NNMi database after a specified number of days in which the nodes are unresponsive. Note the following: • A value of zero (0) indicates that nodes should not be deleted. • Any nodes in the shadow of a deleted node are also deleted.
"Configure Ping Sweep Global Settings" (on page 139)	NNMi Advanced. Ping Sweep works only with IPv4 addresses, not IPv6 addresses. *Optional*. Use the **Discovery Configuration** workspace to configure the starting points for Auto-Discovery. The choices are Discovery Seeds or Ping Sweep within Auto-Discovery Rules (ICMP ping commands) or both. The Global Control settings for Spiral Discovery Ping Sweep Control provide control across all Auto-Discovery Rules.
"Configure the Node Name Strategy" (on page 140)	*Optional*. Use the **Discovery Configuration** workspace to specify a node naming strategy. The Global Control settings for Node Name enable you to choose the most meaningful name for devices in your environment.
"Configure Auto-Discovery Rules" (on page 141)	*Optional*. Use the **Auto-Discovery Rules** tab to specify any IP address ranges or MIB II sysObjectID ranges (or both) that you want NNMi to use for automatic discovery.

Task	How
	Within each Rule you can specify whether Ping Sweep is used as a starting point (in addition to or instead of discovery seeds). NNMi Advanced. Ping Sweep works only with IPv4 addresses, not IPv6 addresses.
"Configure an Excluded IP Addresses Filter" (on page 155)	*Optional*. Use the **Excluded IP Addresses** tab to provide a list of specific addresses or ranges of addresses that you want NNMi to never discover or monitor.
"Configure Subnet Connection Rules" (on page 152)	*Optional*. Use the **Subnet Connection Rules** tab to connect interfaces on devices that *do not respond* to Layer 2 Discovery protocols (for example, WAN edge devices).
"Specify Discovery Seeds: Initial Routers or Specific Nodes to be Discovered" (on page 158)	*Optional*. Use the **Discovery Seeds** tab to specify nodes to be discovered or to indicate which nodes are used as starting points for your Auto-Discovery Rules. **Tip**: Use the Discovery Seeds tab to verify that NNMi successfully located each Discovery Seed that you provided. See "Discovery Seed Results" (on page 167).

Adjust the Rediscovery Interval

When configuring Spiral Discovery, you determine how often network traffic is generated to gather and verify information about your network management domain. This time interval controls how frequently information is gathered about nodes, interfaces, IP addresses, subnets, VLANs, and connections in the network. See "Rediscovery Intervals" (on page 116) for more information.

To adjust the rediscovery cycle interval:

1. Navigate to the **Discovery Configuration** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Discovery Configuration**.
2. Locate the **Global Control** settings.
3. In the **Rediscovery Interval** attribute, set the time interval that Spiral Discovery waits between information gathering cycles.

    The default is 24 hours between cycles. The minimum is 1 hour. Maximum 99 days.

    Make sure the interval value provides plenty of time so Spiral Discovery cycles do not overlap. The larger your network environment, the longer the time required to complete a Spiral Discovery cycle.
4. Click **Save and Close** to apply your changes.

Configure Whether to Delete Unresponsive Nodes

When configuring Spiral Discovery, you determine whether to delete nodes that are unresponsive. NNMi determines whether a node is unresponsive using the following criteria:

- The node does not respond to SNMP requests.
- All of the node's IP Addresses do not respond to ICMP.

Possible Incident Conclusions that might be associated with the node include the following:

- NodeUnmanageable
- NonSNMPNodeUnmanageable
- NodeDown
- NodeOrConnectionDown
- NonSNMPNodeUnresponsive

Note the following:

- A value of zero (0) indicates that nodes should not be deleted.
- Any nodes in the shadow of a deleted node are also deleted.
- Spiral Discovery checks for and deletes unresponsive nodes once per hour.
- NNMi waits 24 hours after the initial discovery before deleting unresponsive nodes to ensure the nodes are polled.

Tip: Check the nnm log file to determine which nodes NNMi deleted. See "Verify that NNMi Services are Running" (on page 47)for more information.

To adjust the Unresponsive Interval:

1. Navigate to the **Discovery Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Discovery Configuration**.
2. Locate the **Global Control** settings.
3. In the **Unresponsive Interval** attribute, set the number of days in which a Node must be unresponsive before NNMi deletes the node and all nodes in its shadow from the NNMi database.

   The default is 0 to indicate that unresponsive nodes are not deleted from the NNMi database.
4. Click **Save and Close** to apply your changes.

Configure Ping Sweep Global Settings

You have two choices for Auto-Discovery starting points. Use either or both to best advantage in your network environment:

- Discovery Seeds: You designate specific hostnames (*not case-sensitive*) or IP addresses where Auto-Discovery starts gathering neighbor information.

  For details see "Discovery Seeds (as a starting point)" (on page 119). For information about creating Discovery Seeds, see "Specify Discovery Seeds: Initial Routers or Specific Nodes to be Discovered" (on page 158).
- Ping Sweep: NNMi issues ICMP pings to certain addresses to find new nodes. For details, see "Ping Sweep (as a starting point)" (on page 119). Note that with *NNMi Advanced*, Ping Sweep works only with IPv4 addresses, not IPv6 addresses.

Ping Sweep uses the current default ICMP interval and timeout settings from the Communications Configuration settings. See "Configure Default SNMP and ICMP Protocol Settings" (on page 67).

To configure the global Auto-Discovery setting for Ping Sweep:

1. Navigate to the **Discovery Configuration** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Discovery Configuration**.
2. Navigate to the **Global Control** settings.
3. Designate the global setting for **Ping Sweep**. Your choice determines how Spiral Discovery uses ICMP ping commands for the discovery process in your network environment:
    - **Each Rule (as configured)**— The instructions for Ping Sweep within each Auto-Discovery Rule configuration are followed exactly.

      To configure Ping Sweep for a specific Auto-Discovery Rule, see "IP Address Ranges for Auto-Discovery" (on page 146).
    - **All Rules**— Ping Sweep is applied for all of your current Auto-Discovery Rules. This overrides the Ping Sweep settings within each rule.
    - **None of the Rules**— Ping Sweep is not used for any of your current Auto-Discovery Rules. This overrides the Ping Sweep settings within each rule. This is useful to temporarily suspend issuing any ping commands within your network.

    Note: If things don't work as expected, check whether ICMP is allowed (see if "Communication Region Form" (on page 80)).
4. Designate the **Sweep Interval** (days/hours) that controls how often Spiral Discovery reissues ICMP Ping for each address. The minimum allowed Sweep Interval setting is 1 hour. Maximum 99 days.
5. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval.

Configure the Node Name Strategy

When configuring discovery in NNMi, you control how the Name attribute on the Node form is populated. For details see "Discovery Node Name Choices" (on page 116).

Note: NNMi can automatically convert the Node Name to all uppercase or all lowercase (if configured to do so by the NNMi administrator). See the information about the `nms-topology.properties` file in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

To resolve issues about choosing the Name value, NNMi follows a sequence of rules. If NNMi is unable to determine a Name based on your three choices, the node name is determined using the NNMi factory defaults for these three choices (see list in step 3).

The node Name shows up beneath the node symbol on the maps and in the Name column on table views.

To control how node names are determined for your network devices:

1. Navigate to the **Discovery Configuration** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Discovery Configuration**.
2. Locate the **Node Name Resolution** attributes on the left side of the form (see table).
3. Specify the three-level hierarchy for node naming decisions.

Short name and full name are related. The short name is everything before the first period in the full name. For example, full name `cisco5500.abc.example.com` and the short name `cisco5500`.

Select among the following choices. Use each choice only one time:

- **Short DNS Name** – (*first by default*) Use the group of characters prior to the first period in your in-house DNS naming standards. See "Discovery Node Name Choices" (on page 116) for possible issues with using DNS names.
- **Fully Qualified DNS Name** – Use the full in-house DNS naming standards.
- **Short sysName** – (*second by default*) Use the group of characters prior to the first period in the current MIB II sysName value established by the administrator for each SNMP enabled device. See "Discovery Node Name Choices" (on page 116) for possible issues with using sysName.
- **Full sysName** – Use the full current MIB II sysName value established by the administrator for each SNMP enabled device.
- **IP Address** – (*third by default*) Use the IP address. If the node responds to SNMP, the SNMP Management Address is used. For non-SNMP nodes, name is set to either a discovery seed address associated with this node or a neighbor address gathered by Spiral Discovery along the path to this node.

  Note: If you listed the address in your Excluded IP Address filter, Spiral Discovery skips that address. See "Exclude Problem IP Addresses from Discovery" (on page 134).

4. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval. To apply the changes immediately, use **Actions → Configuration Poll**. See Using Actions to Perform Tasks for more information.

Node Name Resolution Settings

Attribute	Description
First Choice	Click the drop-down list and choose the predefined node name strategy you want discovery to use first.
Second Choice	Click the drop-down list and choose the predefined node name strategy you want discovery to use if the first choice fails.
Third Choice	Click the drop-down list and choose the predefined node name strategy you want discovery to use if the second choice fails.

Configure Auto-Discovery Rules

Auto-Discovery Rule configuration settings control Spiral Discovery behavior (for details see "Auto-Discovery Rules" (on page 120)):

- Rules define the outer limits of discovery.
- Rules expand or reduce the types of devices that are discovered and added to the topology database.

Before you start, have a clear idea of what you want to accomplish, see "Determine Your Approach to Discovery" (on page 127).

If you do not configure any Auto-Discovery Rules, Spiral Discovery is limited to only discovery seeds. See "Discovery Seeds (as a starting point)" (on page 119) for more information.

To configure Auto-Discovery Rules, do the following:

1. Complete all prerequisites. See "Prerequisites for Discovery" (on page 124), .
2. Navigate to the **Auto-Discovery Rules** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Discovery Configuration**.
   c. Select the **Auto-Discovery Rules** tab.
3. Make your configuration choices (see table).
   - To establish a rule, click the New icon, and continue.
   - To edit a rule, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete a rule, select a row, and click the Delete icon.
   - To refresh the list, click the Refresh icon.
4. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval. To apply the changes immediately, use **Actions → Configuration Poll**. See Using Actions to Perform Tasks for more information.

Auto-Discovery Rule Configuration Tasks

Task	How
"Configure Basic Settings for the Auto-Discovery Rule" (on page 143)	Provide the basic requirements for an Auto-Discovery Rule configuration: • The name of the rule. • Specify the order in which Spiral Discovery applies this rule. • Specify how ICMP and SNMP protocols are used for this segment of discovery. • Designate whether devices identified by this rule are *Discovered* or *Rejected* during the Spiral Discovery process.

Task	How
Rule Criterion	"IP Address Ranges for Auto-Discovery" (on page 146) Use IP addresses with wildcards to specify the area you want Spiral Discovery to find in your network environment. You decide whether Ping Sweep is used for this segment of discovery. **Note**: NNMi also uses the source IP address from SNMP traps as hints to discovery. NNMi Advanced. Ping Sweep works only with IPv4 addresses, not IPv6 addresses.
	"SNMP System Object ID Ranges for Discovery" (on page 150) Use industry standard System Object IDs to control Spiral Discovery: • Expand Spiral Discovery to "include" additional device types to be discovered within the range of IP addresses you specify in this Auto-Discovery Rule. **Note**: requires that you also enable ☑ **Enable Any SNMP Device**. • Instruct Spiral Discovery to "ignore" (never discover, exclude) specific troublesome models of routers, switches, or other devices. **Tip**: To set this exclusion across all Auto-Discovery Rules, do not set any IP Address Ranges in the same Auto-Discovery Rule and use your lowest Ordering number for this Auto-Discovery Rule.

Configure Basic Settings for the Auto-Discovery Rule

The Auto-Discovery Rule settings determine which methods Spiral Discovery applies when discovering the part of your network defined in the rule.

To configure this Auto-Discovery Rule:

1. Navigate to the **Auto-Discovery Rule** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Discovery Configuration**.
   c. Locate the **Auto-Discovery Rules** tab.
   d. Do one of the following:
      - To establish a rule, click the New icon, and continue.
      - To edit a rule, click the Open icon in the row representing the configuration you want to edit, and continue.
      - To delete a rule, select a row, and click the Delete icon.
2. Provide the required basic settings (see the Basics for this Auto-Discovery Rule table).
3. There are many ways to implement discovery. Before you start this step, "Determine Your Approach to Discovery" (on page 127).

   Configure one or more ranges, to identify the devices you want to discover.

- "IP Address Ranges for Auto-Discovery" (on page 146)
- "SNMP System Object ID Ranges for Discovery" (on page 150)

4. *Optional*. Choose the Final Filter settings for this rule (see the Final Filter for this Auto-Discovery Rule table).
5. Click **Save and Close** to return to the **Discovery Configuration** form.
6. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval.
7. *Optional:* Open the **Discovery Configuration** workspace again and provide a discovery seed for each address range of this Auto-Discovery Rule. Core routers make the best seeds. See "Specify Discovery Seeds: Initial Routers or Specific Nodes to be Discovered" (on page 158).

Basics for this Auto-Discovery Rule

Task	How
Name	Give this Auto-Discovery Rule a meaningful name.
Ordering	Determine the order in which the Auto-Discovery Rules are applied. No Duplicate Ordering numbers are allowed. Each Auto-Discovery Rule ordering number must be unique. **Tip**: It is recommended that ordering numbers are incremented by 10s or 100s to provide flexibility when adding new rules over time. **IP address ranges**: If a device falls within two Auto-Discovery Rules, the Auto-Discovery Rule with the lowest ordering number applies. For example, if an Auto-Discovery Rule includes certain IP addresses, then no other Auto-Discovery Rules with higher ordering numbers apply to those addresses. **System Object ID ranges**: • If no IP address range is included in this Auto-Discovery Rule, then the system object ID settings take precedence over all Auto-Discovery Rules that have higher Ordering numbers than this Auto-Discovery Rule. • If an IP address range is included in this Auto-Discovery Rule, your system object ID range applies only within this Auto-Discovery Rule.
Discover Included Nodes	If ☑ enabled, Auto-Discovery gathers information about neighboring devices and adds devices to the NNMi database if they meet the rule's criteria. For more information see "Auto-Discovery Rules" (on page 120). **Note**: By default NNMi discovers routers and switches. You can expand the number of device types that NNMi discovers by enabling ☑**Discover Any SNMP Device** and including one or more System Object ID Ranges (based on MIB II sysObjectID values). Your address ranges and system object ID ranges determine which discovered addresses are added to the NNMi database. If ☐ disabled, Spiral Discovery ignores devices that match this rule unless: • The device's address is a discovery seed. See "Specify Discovery Seeds: Initial Routers or Specific Nodes to be Discovered" (on page 158) to learn how to establish discovery seeds. • The device's address is reported as a neighbor to another discovered address. If you want to ensure that an address is never added to the NNMi database, use the "Configure an Excluded IP Addresses Filter" (on page 155) or "Configure an Excluded

Task	How
	Interfaces Filter" (on page 157) settings.
Notes	Provide any additional useful information about this Auto-Discovery Rule. Type a maximum of 1024 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Final Filter for this Auto-Discovery Rule

Task	How
Discover Any SNMP Device	**Note**: This value is ignored if Discover Included Nodes is unchecked. However, if you configure any device as a discovery seed, discovery always adds that device to the database. If ☑ enabled, discovery gathers information about any device that responds to SNMP queries (in addition to routers or switches that are discovered by default). These nodes appear on maps and are monitored. If ☐ disabled, discovery ignores all device types except routers, switches, discovery seeds, and device types specified in your system object ID ranges. (Routers and switches are identified by the settings in the device profile.)
Discover Non-SNMP Devices	**Note**: This value is ignored if Discover Included Nodes is unchecked. However, if you configure any device as a discovery seed, discovery always adds that device to the database. Non-SNMP devices are those that do not respond to SNMP queries. If you enable **Discover Non-SNMP Devices**, note the following: • If you do not want NNMi to discover every node in your network, make sure your Auto-Discovery Rules correctly limit the scope of the discovery. See"Determine Your Approach to Discovery" (on page 127) for more information. • Selecting this option may cause you to reach your licensed capacity very quickly. See "Extend a Licensed Capacity" (on page 1020). • If NNMi determines that a non-SNMP node has a hostname matching another non-SNMP node, NNMi merges the information to create only one node and includes any additional IP address information under the same node. Non-SNMP nodes might be inaccurately represented under the following circumstances: ▪ One or more non-SNMP nodes in your network use the same hostname. ▪ The same non-SNMP node has multiple hostnames. ▪ A non-SNMP node name changes (see "Delete Nodes" (on page 172)). If ☑ enabled, addresses that do not respond to SNMP queries are added to the database. If ☐ disabled, discovery ignores any address that does not respond to SNMP queries.

IP Address Ranges for Auto-Discovery

Auto-Discovery IP address ranges determine the outer limits for the area controlled by the current Auto-Discovery Rule. You can create multiple IP ranges within one Auto-Discovery Rule. Before you start, have a clear idea of what you want to accomplish, see "Determine Your Approach to Discovery" (on page 127).

Note: NNMi also uses the source IP address from SNMP traps as hints to discovery.

If ☐ **Discover Included Nodes**, click here for additional information about IP address ranges when defining a rule with Discover Included Nodes disabled.

- Spiral Discovery *does not gather neighbor information* from the addresses identified in any IP address range included in this rule. The addresses, themselves, may still show up in the topology database.

  Note: Neighbor information is still gathered from IP addresses specifically identified in the discovery seeds configuration settings.

- IP address ranges are optional. However, when no IP address range is provided:
  - One or more system object ID (MIB II sysObjectIDs) ranges must be defined. This technique constricts or extends the types of devices affected by this rule. See "SNMP System Object ID Ranges for Discovery" (on page 150) for more information.
  - The system object ID range criteria applies to all Discovery Rules with higher Ordering numbers.

If ☑ **Discover Included Nodes**, click here for additional information about IP address ranges when defining a rule with Discover Included Nodes enabled.

- At least one IP address range is required to be designated as an **Include in rule** range type. Auto-Discovery *gathers neighbor information* from those addresses to extend discovery.
- *Optional*. You can configure NNMi to ignore subsets of those IP addresses (an **Ignored by rule** range, which means that those addresses are available for other Auto-Discovery Rules).
- *Optional*. Specify system object ID (MIB II sysObjectIDs) ranges to be included or ignored. This technique constricts or extends the types of devices affected by this rule. See "SNMP System Object ID Ranges for Discovery" (on page 150) for more information.

NNMi discovers any devices that comply with your rule configurations, and creates a record of each device in the NNMi database. If the device supports SNMP, all addresses for that device are combined into one Node object. If the device does not support SNMP, NNMi queries DNS to determine the hostname. If this hostname matches another non-SNMP node, NNMi merges the information to create only one node with multiple associated addresses.

To specify an Auto-Discovery Rule IP address range:

1. Complete all prerequisites. See "Prerequisites for Discovery" (on page 124), .
2. Navigate to the **Auto-Discovery** form.
   a. In the **Workspace** navigation panel, open the **Configuration** workspace.
   b. Select **Discovery Configuration**.
   c. Select the **Auto-Discovery Rule** tab, and do one of the following:
      - To establish an Auto-Discovery Rule, click the New icon.
      - To edit an Auto-Discovery Rule, click the Open icon in the row representing the configuration you want to edit.

3. Navigate to the **IP Ranges** tab.
4. *Optional*. Decide if you want to use Ping Sweep in this segment of network discovery.

   Note: NNMi Advanced - Ping Sweep works only with IPv4 addresses, not IPv6 addresses:

   - **Enable Ping Sweep** ☑
     Auto-Discovery issues a wide range of ICMP ping commands to determine starting points for Spiral Discovery. For details, see "Ping Sweep (as a starting point)" (on page 119). NNMi only uses Ping Sweep across a maximum of the last two octets (/16) of the network specified by each IPv4 IP address range.

     If things don't work as expected, check whether Ping Sweep is disabled (see "Configure Ping Sweep Global Settings" (on page 139)) and check whether ICMP is allowed (see if "Communication Region Form" (on page 80)).

   - **Enable Ping Sweep** ☐
     Auto-Discovery depends on Discovery Seeds as starting points for Spiral Discovery. For details, see "Discovery Seeds (as a starting point)" (on page 119) for important information.

5. *Optional*. To provide an IP address range for this Auto-Discovery Rule, do one of the following:

   - To create an IP range, click the New icon, and continue.
   - To edit an IP range, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an IP range, select a row, and click the Delete icon.

6. Provide the IP address range information for this Auto-Discovery Rule (see table).

   Note: If you choose to not include any IP address ranges in a particular Auto-Discovery Rule, then you must provide at least one system object ID range (see "SNMP System Object ID Ranges for Discovery" (on page 150)). And Auto-Discovery Rules without any IP Range must have **Discover Included Nodes** ☐ disabled in the Auto-Discovery Rule form.

7. Click **Save and Close** to return to the **Auto-Discovery Rule** form.
8. Click **Save and Close** to return to the **Discovery Configuration** form.
9. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval.

Discovery IP Range Form

Name	Description
IP Range	**Note**: If you enter an IP address value that represents only one IP address, Auto-Discovery gathers neighbor information only from the address you enter. (Discovery extends only one hop out from this address.) To specify a range of IP addresses for this Auto-Discovery Rule, use one of the following. Pick one address notation style, combinations of wildcards and CIDR notation are not allowed within one address range. You can provide multiple address range settings: • **IPv4 address wildcard notation**. An IPv4 Address range is a modified dotted-notation where each octet is one of the following: ▪ A specific octet value between 0 and 255 ▪ A low-high range specification for the octet value (for example, "112-119")

Name	Description

- An asterisk (*) wildcard character which is equivalent to the range expression "0-255"

Note: The following two IPv4 addresses are considered invalid: `0.0.0.0` and `127.0.0.0`

Examples of valid IPv4 address wildcards include:

```
10.1.1.*
10.*.*.*
10.1.1.1-99
10.10.50-55.*
10.22.*.4
10.1-9.1-9.1-9
```

- **IPv4 Classless Inter-Domain Routing (CIDR) notation**.

  The CIDR notation specifies the number of consecutive bits in the IPv4 address that must match.

  For example, `10.2.120.0/21`

  Note: NNMi does not support CIDR subnet mask notation such as, `10.2.120.0/255.255.248.0`

Example IPv4 Prefix Length Values	Number of Usable IPv4 Addresses
28	14 (16-2=14)*
29	6 (8-2=6)*
30	2 (4-2=2)*
31	2

* Two IPv4 addresses are reserved in each subnet. The first IPv4 address is used for the network itself and the last IPv4 address is reserved for broadcast.

- **IPv6 address wildcard notation**

  Separate each 16-bit value of the IPv6 address with a colon. The 16-bit value can be any of the following:

  - A specific hexadecimal value between `0` and `FFFF` (case insensitive).
  - A low-high range specification of the hexadecimal value (for example, `1-1fe`).
  - An asterisk (`*`) wildcard character (equivalent to the range expression `0-ffff`).

  Note: The standard IPv6 short-hand notation (`::`) is allowed to express one or more 16-bit elements of zero (`0`) values. However, the mixed IPv6/IPv4 dot-notation (for example, `2001:d88::1.2.3.4`) is not allowed as an IPv6 address range.

  Valid examples of ranges in modified IPv6 address notation include the following:

```
2001:D88:0:A00-AFF:*:*:*:*
2001:D88:1:*:*:*:*:*
2001:D88:2:0:a07:ffff:0a01:3200-37ff
```

- **IPv6 Classless Inter-Domain Routing (CIDR) notation**

  The CIDR notation specifies the number of consecutive bits in the IPv6 address that must match.

Name	Description
	`1080:0:a00::/44` (equivalent to modified IPv6 address notation `1080:0:a00-a0f:*:*:*:*:*`) For example, valid IPv6 address ranges in CIDR notation include the following: `2001:d88:0:a00::/56` (equivalent to modified IPv6 address notation `2001:D88:0:A00-AFF:*:*:*:*`) `2001:d88:1::/48` (equivalent to modified IPv6 address notation `2001:D88:1:*:*:*:*:*`)

Name	Description
Range Type	**Include in rule** - The current Auto-Discovery Rule settings apply to the addresses in this range. **Ignored by rule** - The current Auto-Discovery Rule settings do not apply to the addresses in this range. Use the **Ignored by rule** setting to identify a subset of addresses within a larger range. The addresses in the ignored range are available to conform to an Auto-Discovery Rule with a higher ordering number.

SNMP System Object ID Ranges for Discovery

Vendors are assigned a system object ID (RFC 1213 MIB II sysObjectID) for each type of network device that they manufacture. This system object ID number is unique for each combination of vendor, device type, and model number. For example, all Cisco 6509 routers have the same system object ID.

Tip: See "Configure Device Profiles" (on page 135) for more information about system object IDs. In the Device Profiles view (in the **Configuration** workspace), you can quickly and easily locate the system object IDs of devices in your network environment.

System object ID ranges are powerful tools for expanding or limiting discovery behavior. For example, expand discovery to include more than the default routers and switches, or limit discovery by excluding specific models of routers and switches. Before you start, have a clear idea of what you want to accomplish, see "Determine Your Approach to Discovery" (on page 127).

When using system object ID ranges, note the following:

- When one or more IP address ranges are defined within the Auto-Discovery Rule, your system object ID ranges apply only within the current Auto-Discovery Rule.
- When no IP address ranges are defined within the Auto-Discovery Rule, your system object ID ranges take precedence over all Auto-Discovery Rules that have higher Ordering numbers than this Auto-Discovery Rule.
- To enable Discover Included Nodes in this rule, at least one IP address range is required. Before any discovered node is added to the topology database, it must match both IP address range and system object ID range specifications.

The following table includes examples of how you might want to expand or limit your Spiral Discovery scope using System Object ID Ranges.

Controlling Spiral Discovery with System Object ID Ranges

Task	Related Topics
Expand Spiral Discovery to include device types in addition to routers and switches.	"All Devices from a Specific Vendor Discovered" (on page 132)
Globally exclude one or more specific device types from Spiral Discovery.	"Specific System Object IDs Not Discovered" (on page 134)

To specify a system object ID range:

1. Complete all prerequisites. See "Prerequisites for Discovery" (on page 124), .
2. Navigate to the **Discovery System Object ID Range** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Discovery Configuration**.

c. Select the **Auto-Discovery Rule** tab.

d. Do one of the following:

- To create an Auto-Discovery Rule, click the New icon.
- To edit an Auto-Discovery Rule, click the Open icon in the row representing the configuration you want to edit.

e. In the **Auto-Discovery Rule** form, select the **System Object ID Ranges** tab.

f. Do one of the following:

- To create a system object ID range, click the New icon, and continue.
- To edit a system object ID range, click the Open icon in the row representing the configuration you want to edit, and continue.
- To delete a system object ID range, click the Delete icon.

3. Provide one or more System Object ID ranges for this Auto-Discovery Rule (see the table).

   Use multiple System Object ID ranges to fine tune your discovery settings.

   Note: If you do not include any System Object ID ranges in a particular Auto-Discovery Rule, then you must provide at least one IP address range in that particular Auto-Discovery Rule (see "IP Address Ranges for Auto-Discovery" (on page 146)).

   Example 1. In an Auto-Discovery Rule with Discover Included Nodes enabled:

   - Create a definition that includes all HP devices. Use the System Object ID prefix 1.3.6.1.4.1.11 and set the Range Type to *Include in rule*.
   - Create a definition that excludes any HP Printers. Use the System Object ID prefix 1.3.6.1.4.1.11.2.3.9 and set the Range Type to *Ignored by rule*. (Order does not matter, now the printers are always ignored.)

   Example 2. In an Auto-Discovery Rule with Discover Included Nodes disabled:

   - Create a definition that excludes any HP Printers. Use the System Object ID prefix 1.3.6.1.4.1.11.2.3.9 and set the Range Type to *Include in rule*.
   - Skip step 4, below. HP printers are not discovered within the IP address range of any Auto-Discovery Rules with higher ordering numbers than this rule.

4. Click **Save and Close** to return to the **Auto-Discovery Rule** form.
5. Provide any IP ranges (see "IP Address Ranges for Auto-Discovery" (on page 146)):
   - Optional if Discover Included Nodes is disabled in the Auto-Discovery Rule form.
   - Required if Discover Included Nodes is enabled in the Auto-Discovery Rule form.

   Click **Save and Close** to return to the **Auto-Discovery Rule** form.
6. Click **Save and Close** to return to the **Discovery Configuration** form.
7. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval.

Discovery System Object ID Range Definition

Attribute	Description
System Object ID Prefix	Enter a prefix of an SNMP system object ID, or enter the entire SNMP system object ID. A partial entry becomes a wildcard. For example, if you enter 1.3.6.1.4.1.11, discovery finds all HP devices. If you enter 1.3.6.1.4.1.9, discovery finds all Cisco devices. **Note**: Do not use dashes or asterisks (*) in your system object ID value.
Range Type	**Include in rule** - Instructs Auto-Discovery to find devices matching this system object ID range. **Ignored by rule** - Instructs Auto-Discovery to ignore devices matching this system object ID range.
Notes	Add any information about this rule that would be useful to you and your team. Type a maximum of 1024 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Configure Subnet Connection Rules

The NNMi Subnet Connection Rules work only with IPv4 subnets.

For an explanation of how Subnet Connection Rules work, see "Subnet Connection Rules" (on page 122).

If important subnets in your network environment are not automatically connected by Spiral Discovery, edit a Subnet Connection Rule or create your own.

The following are typical situations that require Subnet Connection Rules:

- Point-to-point or point-to-multipoint connections between interfaces within subnets that have a prefix length ranging from 28-31.
- IPv4 tunnel or other virtual connections between interfaces within subnets that have a prefix length ranging from 28-31.

NNMi uses Subnet Connection Rules to detect connections between interfaces associated with IPv4 addresses that *do not respond* to Layer 2 Discovery protocols (see the list of Topology Source protocols in Layer 2 Connection Form). Subnet Connection Rules take priority over the Layer 2 Discovery protocol results. For special cases, you can override a Subnet Connection Rule by using the Connection Editor command line tool, see the nnmconnedit.ovpl Reference Page for more information.

When Spiral Discovery detects a subnet, NNMi uses the matching Subnet Connection Rule to request information about all possible IPv4 addresses (potentially detecting previously undiscovered IPv4 addresses). NNMi checks the Excluded IP Addresses list. Any addresses in the list are dropped (for details, see "Filters to Exclude Certain IP Addresses from Discovery" (on page 121)). Then NNMi creates connections among any interfaces associated with any newly discovered IPv4 addresses.

To configure Subnet Connection Rules:

1. Complete all prerequisites. See "Prerequisites for Discovery" (on page 124), .
2. Navigate to the **Subnet Connection Rule** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Discovery Configuration**.
    c. Select the **Subnet Connection Rules** tab.

d. Do one of the following:

- To establish a rule, click the New icon, and continue.
- To edit a rule, click the Open icon in the row representing the configuration you want to edit, and continue.
- To delete a rule, select a row, and click the Delete icon.

3. Provide the required basic settings (see Basics table).
4. Provide the Subnet Connection behavior settings for this rule (see Details table).
5. Click **Save and Close** to return to the **Discovery Configuration** form.
6. Click **Save and Close** to apply the configuration. Spiral Discovery implements your changes during the next regularly scheduled discovery interval. If more than two nodes are connected using this rule, NNMi uses the following icon to indicate this special connection on maps (see example in "Subnet Connection Rules" (on page 122)):

If you double-click the icon, the Layer 2 Connection form displays and the **Topology Source** value is `SUBNETCONNECTION`.

Basics for this Subnet Connection Rule

Task	How
Name	Type a meaningful name for this Subnet Connection Rule. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. No spaces are allowed. **Note**: This name is prepended to the Layer 2 connection name (when you request Quick View information about the connection on the Layer 2 Neighbor View map). If a subnet matches more than one rule, NNMi randomly chooses from among the matching rules.
Enable	If enabled ☑, NNMi uses the Subnet Connection Rule to create connections between interfaces associated with the IPv4 addresses within the specified subnets. If disabled ☐, NNMi ignores the Subnet Connection Rule.

Details for this Subnet Connection Rule

<table>
<tr><th>Task</th><th>How</th></tr>
<tr><td>Minimum IPv4 Prefix Length</td><td>Specify the minimum prefix length (subnet mask length) for the subnet where you want Spiral Discovery to create Layer 2 connections. Spiral Discovery creates connections between interfaces associated with IPv4 addresses that have subnet prefix lengths equal to or greater than the specified value and meet the other specified criteria.
<table>
<tr><th>Valid Minimum IPv4 Prefix Length Values</th><th>Number of Usable IPv4 Addresses</th></tr>
<tr><td>28</td><td>14 (16-2=14)*</td></tr>
<tr><td>29</td><td>6 (8-2=6)*</td></tr>
<tr><td>30</td><td>2 (4-2=2)*</td></tr>
<tr><td>31</td><td>2</td></tr>
</table>
* Two IPv4 addresses are reserved in each subnet. The first IPv4 address is used for the network itself and the last IPv4 address is reserved for broadcast.</td></tr>
<tr><td>ifType</td><td>Optional. Use this Interface MIB variable as an additional filter to identify the types of interfaces to include when creating the subnet connections. For example, if you want connections only between frameRelay interfaces, select <code>frameRelay</code> as the ifType.</td></tr>
<tr><td>ifName</td><td>Optional. Use this Interface MIB variable as an additional filter to identify the interfaces to include when creating the subnet connections. This attribute is useful if you have a naming convention that is used to identify a set of interfaces. For example, <code>lan0</code>.

Maximum 255 characters. The following wildcard characters are allowed: asterisk (*) represents any string, and question mark (?) represents a single character.</td></tr>
<tr><td>ifDescription</td><td>Optional. Use this Interface MIB variable as an additional filter to identify the interfaces to include when creating the subnet connections. For example, you might want to select a particular set of interfaces that have the same vendor description.

Maximum 255 characters. The following wildcard characters are allowed: asterisk (*) represents any string, and question mark (?) represents a single character.</td></tr>
<tr><td>ifAlias</td><td>Optional. Use this Interface MIB variable as an additional filter to identify the interfaces to include when creating the subnet connections. This attribute is useful if you have an alias naming convention that is used to identify a set of interfaces. For example, <code>Connection to remote store in Hawaii.</code>

Maximum 255 characters. The following wildcard characters are allowed: asterisk (*) represents any string, and question mark (?) represents a single character.</td></tr>
</table>

Related Topics

Intrepret Root Cause Messages

Subnet Connection Rules Provided by NNMi

The NNMi Subnet Connection Rules work only with IPv4 subnets.

NNMi provides the Subnet Connection Rules described in the following table (for more information, see "Subnet Connection Rules" (on page 122)).

The *Small Subnets* Rule ensures that NNMi detects IPv4 addresses within subnets of this size, regardless of the interface type. The remaining Subnet Connection Rules create connections based on interface type and the specified subnet size.

Tip: See IfTypes (Interface Types) Form for more information about interface types.

To create new Subnet Connection Rules (or modify the ones provided), see "Configure Subnet Connection Rules" (on page 152).

Subnet Connection Rules Provided by NNMi

Rule Name	Minimum IPv4 Prefix Length (Subnet Mask Length)	Interface Type (#)
Asynchronous Transfer Mode	28	atm (37)
Digital Signal 0	28	ds0 (81)
Digital Signal 1	28	ds1 (18)
Digital Signal 3	28	ds3 (30)
Digital Subscriber Loop over ISDN	28	idsl (154)
Frame Relay Interfaces	28	frameRelay (32)
Integrated Services Digital Network	28	isdn (63)
Multiprotocol Label Switching	28	mpls (166)
Point to Point	28	ppp (23)
Small Subnets	30	
Serial Line Internet Protocol	28	slip (28)
Serial Point to Point	28	propPointToPointSerial (22)
Synchronous Optical Networking	28	sonnet (39)

Configure an Excluded IP Addresses Filter

Sometimes there are IP addresses or ranges of IP addresses in your environment that you do not want NNMi to discover or monitor. For details and examples, see "Filters to Exclude Certain IP Addresses from Discovery" (on page 121).

Tip: If you have a large number of IP addresses that you want to exclude from Spiral Discovery, see the nnmdiscocfg.ovpl Reference Page.

To excluded specific IP addresses from the discovery process:

1. Complete all prerequisites. See "Prerequisites for Discovery" (on page 124), .
2. Navigate to the **Excluded IP Address** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Discovery Configuration**.
    c. Select the **Excluded IP Addresses** tab.

d. Do one of the following:

- To exclude an address or range of addresses from Spiral Discovery, click the New icon, and continue.
- To edit an excluded address setting, click the Open icon in the row representing the configuration you want to edit, and continue.
- To delete an excluded address setting, select a row, and click the Delete icon.

3. To specify a range of IP addresses for this Auto-Discovery Rule, use one of the following. Pick one address notation style, combinations of wildcards and CIDR notation are not allowed within one address range. You can provide multiple address range settings:

- **IPv4 address wildcard notation**.

  An IPv4 Address range is a modified dotted-notation where each octet is one of the following:

  - A specific octet value between 0 and 255
  - A low-high range specification for the octet value (for example, "112-119")
  - An asterisk (*) wildcard character which is equivalent to the range expression "0-255"

  Note: The following two IPv4 addresses are considered invalid: `0.0.0.0` and `127.0.0.0`

  Examples of valid IPv4 address wildcards include:

```
10.1.1.*
10.*.*.*
10.1.1.1-99
10.10.50-55.*
10.22.*.4
10.1-9.1-9.1-9
```

- **IPv4 Classless Inter-Domain Routing (CIDR) notation**.

  The CIDR notation specifies the number of consecutive bits in the IPv4 address that must match.

  For example, `10.2.120.0/21`

  Note: NNMi does not support CIDR subnet mask notation such as, `10.2.120.0/255.255.248.0`

Example IPv4 Prefix Length Values	Number of Usable IPv4 Addresses
28	14 (16-2=14)*
29	6 (8-2=6)*
30	2 (4-2=2)*
31	2

  * Two IPv4 addresses are reserved in each subnet. The first IPv4 address is used for the network itself and the last IPv4 address is reserved for broadcast.

- **IPv6 address wildcard notation**

  Separate each 16-bit value of the IPv6 address with a colon. The 16-bit value can be any of the following:

- A specific hexadecimal value between `0` and `FFFF` (case insensitive).
- A low-high range specification of the hexadecimal value (for example, `1-1fe`).
- An asterisk (`*`) wildcard character (equivalent to the range expression `0-ffff`).

Note: The standard IPv6 short-hand notation (`::`) is allowed to express one or more 16-bit elements of zero (`0`) values. However, the mixed IPv6/IPv4 dot-notation (for example, `2001:d88::1.2.3.4`) is not allowed as an IPv6 address range.

Valid examples of ranges in modified IPv6 address notation include the following:

```
2001:D88:0:A00-AFF:*:*:*:*
2001:D88:1:*:*:*:*:*
2001:D88:2:0:a07:ffff:0a01:3200-37ff
```

- **IPv6 Classless Inter-Domain Routing (CIDR) notation**

  The CIDR notation specifies the number of consecutive bits in the IPv6 address that must match.

  `1080:0:a00::/44` (equivalent to modified IPv6 address notation `1080:0:a00-a0f:*:*:*:*:*`)

  For example, valid IPv6 address ranges in CIDR notation include the following:

  `2001:d88:0:a00::/56` (equivalent to modified IPv6 address notation `2001:D88:0:A00-AFF:*:*:*:*`)

  `2001:d88:1::/48` (equivalent to modified IPv6 address notation `2001:D88:1:*:*:*:*:*`)

4. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval. To apply the changes immediately, use **Actions → Configuration Poll**. See Using Actions to Perform Tasks for more information.

Configure an Excluded Interfaces Filter

Sometimes there are certain types of interfaces in your environment that you do not want NNMi to discover or monitor. For details and examples, see "Filters to Exclude Certain Interfaces from Discovery" (on page 121).

To excluded specific types of interfaces from the discovery process:

1. Complete all prerequisites. See "Prerequisites for Discovery" (on page 124), .
2. Navigate to the **Excluded Interfaces** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Discovery Configuration**.
   c. Select the **Excluded Interfaces** tab.
3. Do one of the following:
   - To select an Interface Group to filter certain interfaces out of Spiral Discovery, click the New icon, and continue.
   - To edit an excluded interfaces setting, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an excluded interfaces setting, select a row, and click the Delete icon.

- To refresh the list of excluded interface settings, click the Refresh icon.
- To open the excluded interfaces settings list in a new window, click the New Window icon.

4. In the Interface Filter form, click the Lookup icon and select one of the options from the drop-down menu:
   - Quick View to display summary information for the currently selected Interface Group.
   - Quick Find to view and select from the list of all existing Interface Groups (for more information see "Use the Quick Find Window" (on page 28)).
   - Open to display the details of the currently selected Interface Group.
   - New to create a new Interface Group (see "Create Interface Groups" (on page 197) for more information).
5. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval. To apply the changes immediately, use **Actions** → **Configuration Poll**. See Using Actions to Perform Tasks for more information.

Specify Discovery Seeds: Initial Routers or Specific Nodes to be Discovered

Note: Discovery seeds are optional. Before you start this task, "Determine Your Approach to Discovery" (on page 127) and complete the prerequisites ("Prerequisites for Discovery" (on page 124)).

Nodes specified as discovery seeds are always discovered and added to the topology database. As soon as you enter one or more optional discovery seeds, discovery begins.

If you create Auto-Discovery Rules, NNMi uses neighbor information gathered from each discovery seed to extend discovery. See "Discovery Seeds (as a starting point)" (on page 119) for more information. NNMi can also use Ping Sweep (instead of or in addition to discovery seeds) to gather neighbor information. See "Ping Sweep (as a starting point)" (on page 119). Note that with NNMi Advanced, Ping Sweep works only with IPv4 addresses, not IPv6 addresses.

If you want Spiral Discovery to automatically find devices on your network, before you begin adding discovery seeds:

- Configure at least one Auto-Discovery Rule. See "Configure Auto-Discovery Rules" (on page 141).
- Configure any number of Auto-Discovery Rules to maintain fine control over the scope of Spiral Discovery.

A discovery seed is a hostname (*not case-sensitive*) or IP address. Consider devices with the largest neighbor data in your network environment. For example, a good choice would be a core router connected to a network you want to discover.

If you change your mind and delete a discovery seed from Discovery Configuration, the corresponding node is not deleted from the topology database. See "Delete Nodes" (on page 172) for information about removing the entire node record from the topology database.

To configure discovery seeds do one or more of the following:

- "In the Console, Configure Discovery Seeds " (on page 159)
- "With a Seed File, Add Multiple Discovery Seeds" (on page 162)
- "From the Command Line, Add Discovery Seeds" (on page 164)

Related Topics

"Discovery Seed Results" (on page 167)

"Delete Discovery Seeds" (on page 173)

In the Console, Configure Discovery Seeds

There are many ways to provide discovery seeds. Discovery seeds are optional. See "Specify Discovery Seeds: Initial Routers or Specific Nodes to be Discovered" (on page 158)for more information.

To add an optional discovery seed using the console:

1. Complete all prerequisites. See "Prerequisites for Discovery" (on page 124), .
2. Navigate to the **Discovery Seeds** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Discovery Configuration**.
   c. Locate the **Discovery Seeds** tab.
   d. Do one of the following:
      - To add a discovery seed, click the New icon.
      - To edit a discovery seed, click the Open icon in the row representing the discovery seed you want to edit.
      - To delete a discovery seed, select a row, and click the Delete icon (see "Delete Discovery Seeds" (on page 173) and "Delete Nodes" (on page 172) for more information).
3. Provide appropriate information (see table).

   NNMi uses information gathered from Routers to establish subnet membership for Layer 3 connections. Make sure that important Routers in your network environment are SNMP enabled.

   NNMi uses either of the following criteria to identify a Router:

   - Each Router responds with appropriate values for sysServices (1.3.6.1.2.1.1.7) and ipForwarding (1.3.6.1.2.1.4.1). See RFC 1213-MIB for details.
   - Identify Routers with the Device Profile configuration settings for a particular device type.

   You must provide the appropriate SNMP Community Strings to NNMi. See "Configuring Communication Protocol" (on page 66).
4. Click **Save and Close** to return to the Discovery Configuration form.

   Tip: Click the Save and New icon to continue to adding discovery seeds.
5. Click **Save and Close**. As soon as you enter one or more optional discovery seeds, discovery begins.

Discovery Seed Definition

Attribute	Definition

Hostname / IP Address	**Note**: NNMi does not validate your entry when you use this method to add discovery seeds. Use the nnmloadseeds.ovpl command to validate your discovery seed entries.
	
	To identify the node, enter one of the following:
	
	• **Fully-qualified hostname** of the discovery seed (*not case-sensitive*)
	• **IP address** of the discovery seed, specify a physical address
	
	When providing IPv4 addresses as discovery seeds, click here for more information.
	
	The following IPv4 addresses are considered invalid:
	
	• `255.255.255.255`
	• IP addresses that begin or end with 0 (zero)
	
	When providing IPv6 addresses as discovery seeds, use IPv6 notation as defined in RFC 2373. click here for more information.
	
	• 16-byte (128-bit) address, composed of eight groups of 2-byte (16-bit) hex values separated by colons (XXXX:XXXX: XXXX:XXXX: XXXX:XXXX: XXXX:XXXX)
	• Uppercase and lowercase (A-F/a-f) allowed for the hex digits.
	
	Note: NNMi displays IPv6 addresses as all lowercase.
	
	• *Optional*. Omit leading zeros in each 2-byte hex value.
	• `::` means a single contiguous sequence of all zero 2-byte hex values. However, `::` is allowed only one time per address. For example, the following three IPv6 address notations are equivalent:
	`1080:0000:0000:0000:0008:0800:200C:417A`
	`1080:0:0:0:8:800:200c:417a`
	`1080::8:800:200C:417a`
	
	• For the right-most 32-bits, IPv4 dotted-decimal notation can replace the pair of 2-byte hex values. For example, the following two IPv6 address notations are equivalent:
	`1080::5efe:10.7.150.201`
	`1080::5efe:a07:96c9`
	
	Types of IPv6 Addresses
	
	IPv6 Address Range — Explanation
	`2000::` to `3fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff` — **global unicast address**[1]
	`fc00::` to `fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff` — **unique local address**[2]
	`fe80::XXXX:XXXX:XXXX:XXXX` — **link local address**[3] (do not use as a seed)
	`ff00::` to `ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff` — **multicast address**[4] (do not use as a seed)

[1](2000:: to 3fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff) A publically routable IPv6 unicast address, used for communication between nodes anywhere on the internet. The first part of the address is a global routing prefix in the 2000::/3 address space for your organization (assigned by the Internet Service Providers). The complete host address may either be manually configured or automatically assigned using IPv6 auto-configuration and neighbor discovery.

[2](fd00:: to fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff) A privately routable IPv6 unicast address used only for communication between nodes within your organization. The unique local addresses cannot be routed to the public internet. Comprised of a routing prefix in the fd00::/8 address spaces assigned locally by your organization. And the full host address may be manually configured or automatically assigned using IPv6 auto-configuration and neighbor discovery.

[3]A non-routable IPv6 unicast address only used for communication with other nodes on the same link (LAN or VLAN). Link local addresses cannot be used for communication that must be forwarded through a router. IPv6 auto-configuration automatically assigns a unique link local address in the fe80::/10 address space to each IPv6-enabled interface on a system.

[4]Used to identify a group of hosts joined into a group, IPv4 multicast addresses are in the range 224.0.0.0 to 239.255.255.255 and IPv6 multicast addresses have the prefix ff00::/8

Attribute	Definition
Discovery Seed Results	An automatically generated value. The most recent discovery status for this discovery seed. See "Discovery Seed Results" (on page 167) for details.
Last Modified	The date and time of the last change in Discovery Seed Results.

Attribute	Definition
Notes	Provide any additional information about this discovery seed that would be useful to you or your team. Type a maximum of 1024 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

With a Seed File, Add Multiple Discovery Seeds

There are many ways to provide discovery seeds. Discovery seeds are optional. See "Specify Discovery Seeds: Initial Routers or Specific Nodes to be Discovered" (on page 158)for more information.

Use a seed file to simultaneously add large numbers of discovery seeds. Your seed file contains one line for each discovery seed. For example:

```
12.2.111.104# cisco5500
12.2.112.268# cisco6509
12.2.119.205# cisco5500
```

Note: Any comments included after the # in a seed file become Notes attribute values for the discovery seeds.

To identify a discovery seed, enter one of the following:

- **Fully-qualified hostname** of the discovery seed (*not case-sensitive*)
- **IP address** of the discovery seed, specify a physical address

When providing IPv4 addresses as discovery seeds, click here for more information.

The following IPv4 addresses are considered invalid:

- `255.255.255.255`
- IP addresses that begin or end with 0 (zero)

When providing IPv6 addresses as discovery seeds, use IPv6 notation as defined in RFC 2373. click here for more information.

- 16-byte (128-bit) address, composed of eight groups of 2-byte (16-bit) hex values separated by colons (XXXX:XXXX: XXXX:XXXX: XXXX:XXXX: XXXX:XXXX)
- Uppercase and lowercase (A-F/a-f) allowed for the hex digits.

  Note: NNMi displays IPv6 addresses as all lowercase.
- *Optional*. Omit leading zeros in each 2-byte hex value.
- `::` means a single contiguous sequence of all zero 2-byte hex values. However, `::` is allowed only one time per address. For example, the following three IPv6 address notations are equivalent:
  ```
  1080:0000:0000:0000:0008:0800:200C:417A
  1080:0:0:0:8:800:200c:417a
  1080::8:800:200C:417a
  ```
- For the right-most 32-bits, IPv4 dotted-decimal notation can replace the pair of 2-byte hex values. For example, the following two IPv6 address notations are equivalent:
  ```
  1080::5efe:10.7.150.201
  1080::5efe:a07:96c9
  ```

Types of IPv6 Addresses

IPv6 Address Range	Explanation
`2000:: to 3fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff`	**global unicast address**[1]
`fd00:: to fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff`	**unique local address**[2]
`fe80::XXXX:XXXX:XXXX:XXXX`	**link-local address**[3] (do not use as a seed)
`ff00:: to ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff`	**multicast address**[4] (do not use as a seed)

NNMi uses information gathered from Routers to establish subnet membership for Layer 3 connections. Make sure that important Routers in your network environment are SNMP enabled.

NNMi uses either of the following criteria to identify a Router:

- Each Router responds with appropriate values for sysServices (1.3.6.1.2.1.1.7) and ipForwarding (1.3.6.1.2.1.4.1). See RFC 1213-MIB for details.
- Identify Routers with the Device Profile configuration settings for a particular device type.

You must provide the appropriate SNMP Community Strings to NNMi. See "Configuring Communication Protocol" (on page 66).

To create a seed file:

In a text editor, type each entry on a separate line in the following format:

`<IP_address>` or `<hostname> #(optional comment to help identify the node)`

- <*IP_address*> = the IP address of the node
- <*hostname*> = the DNS fully-qualified or short hostname (*not case-sensitive*) of the node

To add discovery seeds by loading a seed file:

Use the `nnmloadseeds.ovpl` command:

<path>/<file_name> = the name of the file that contains your discovery seeds

Windows:
`%NnmInstallDir%\bin\nnmloadseeds.ovpl -f <path>\<file_name>`

---

[1](2000:: to 3fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff) A publically routable IPv6 unicast address, used for communication between nodes anywhere on the internet. The first part of the address is a global routing prefix in the 2000::/3 address space for your organization (assigned by the Internet Service Providers). The complete host address may either be manually configured or automatically assigned using IPv6 auto-configuration and neighbor discovery.
[2](fd00:: to fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff) A privately routable IPv6 unicast address used only for communication between nodes within your organization. The unique local addresses cannot be routed to the public internet. Comprised of a routing prefix in the fd00::/8 address spaces, assigned locally by your organization. And the full host address may be manually configured or automatically assigned using IPv6 auto-configuration and neighbor discovery.
[3]A non-routable IPv6 unicast address only used for communication with other nodes on the same link (LAN or VLAN). Link local addresses cannot be used for communication that must be forwarded through a router. IPv6 auto-configuration automatically assigns a unique link local address in the fe80::/10 address space to each IPv6-enabled interface on a system.
[4]Used to identify a group of hosts joined into a group, IPv4 multicast addresses are in the range 224.0.0.0 to 239.255.255.255 and IPv6 multicast addresses have the prefix ff00::/8

UNIX:
`/opt/OV/bin/nnmloadseeds.ovpl -f <path>/<file_name>`

A message displays, showing the number of added, invalid, and ignored discovery seeds. For example:

```
26 seeds added
0 seeds invalid
0 seeds duplicated
```

See the nnmloadseeds.ovpl Reference Page for more information.

Related Topics

"Discovery Seed Results" (on page 167)

"Delete Discovery Seeds" (on page 173)

From the Command Line, Add Discovery Seeds

There are many ways to provide discovery seeds. Discovery seeds are optional. See "Specify Discovery Seeds: Initial Routers or Specific Nodes to be Discovered" (on page 158)for more information.

You can add optional discovery seeds using the nnmloadseeds.ovpl command:

<seed_list> = the discovery seed entries (fully-qualified DNS hostname, short DNS hostname, or IP address)

Windows:
`%NnmInstallDir%\bin\nnmloadseeds.ovpl -n <seed_list>`

UNIX:
`/opt/OV/bin/nnmloadseeds.ovpl -n <seed_list>`

In the following example, the devices with a hostname of cisco4 and cisco5, and a device with the IP address of 12.6.91.5 are added as discovery seeds.

```
nnmloadseeds.ovpl -n cisco4 cisco5 12.6.91.5
```

Note: Identify the discovery seed by either a DNS-resolvable hostname or an IP address.

When adding individual discovery seeds using the **nnmloadseeds.ovpl** command:

- **Fully-qualified hostname** of the discovery seed (*not case-sensitive*)
- **IP address** of the discovery seed, specify a physical address

When providing IPv4 addresses as discovery seeds, click here for more information.

The following IPv4 addresses are considered invalid:

- `255.255.255.255`
- IP addresses that begin or end with 0 (zero)

When providing IPv6 addresses as discovery seeds, use IPv6 notation as defined in RFC 2373. click here for more information.

- 16-byte (128-bit) address, composed of eight groups of 2-byte (16-bit) hex values separated by colons (XXXX:XXXX: XXXX:XXXX: XXXX:XXXX: XXXX:XXXX)
- Uppercase and lowercase (A-F/a-f) allowed for the hex digits.

Note: NNMi displays IPv6 addresses as all lowercase.

- *Optional*. Omit leading zeros in each 2-byte hex value.
- `::` means a single contiguous sequence of all zero 2-byte hex values. However, `::` is allowed only one time per address. For example, the following three IPv6 address notations are equivalent:
  `1080:0000:0000:0000:0008:0800:200C:417A`
  `1080:0:0:0:8:800:200c:417a`
  `1080::8:800:200C:417a`
- For the right-most 32-bits, IPv4 dotted-decimal notation can replace the pair of 2-byte hex values. For example, the following two IPv6 address notations are equivalent:
  `1080::5efe:10.7.150.201`
  `1080::5efe:a07:96c9`

Types of IPv6 Addresses

IPv6 Address Range	Explanation
`2000:: to 3fff:ffff:ffff:ffff:ffff:ffff:ffff`	**global unicast address**[1]
`fd00:: to fdff:ffff:ffff:ffff:ffff:ffff:ffff`	**unique local address**[2]
`fe80::XXXX:XXXX:XXXX:XXXX`	**link-local address**[3] (do not use as a seed)
`ff00:: to ffff:ffff:ffff:ffff:ffff:ffff:ffff`	**multicast address**[4] (do not use as a seed)

Communicate any additional IP address requirements to your team to avoid unexpected discovery results.

NNMi uses information gathered from Routers to establish subnet membership for Layer 3 connections. Make sure that important Routers in your network environment are SNMP enabled.

NNMi uses either of the following criteria to identify a Router:

- Each Router responds with appropriate values for sysServices (1.3.6.1.2.1.1.7) and ipForwarding (1.3.6.1.2.1.4.1). See RFC 1213-MIB for details.
- Identify Routers with the Device Profile configuration settings for a particular device type.

You must provide the appropriate SNMP Community Strings to NNMi. See "Configuring Communication Protocol" (on page 66).

Related Topics

---

[1](2000:: to 3fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff) A publically routable IPv6 unicast address, used for communication between nodes anywhere on the internet. The first part of the address is a global routing prefix in the 2000::/3 address space for your organization (assigned by the Internet Service Providers). The complete host address may either be manually configured or automatically assigned using IPv6 auto-configuration and neighbor discovery.
[2](fd00:: to fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff) A privately routable IPv6 unicast address used only for communication between nodes within your organization. The unique local addresses cannot be routed to the public internet. Comprised of a routing prefix in the fd00::/8 address spaces, assigned locally by your organization. And the full host address may be manually configured or automatically assigned using IPv6 auto-configuration and neighbor discovery.
[3]A non-routable IPv6 unicast address only used for communication with other nodes on the same link (LAN or VLAN). Link local addresses cannot be used for communication that must be forwarded through a router. IPv6 auto-configuration automatically assigns a unique link local address in the fe80::/10 address space to each IPv6-enabled interface on a system.
[4]Used to identify a group of hosts joined into a group, IPv4 multicast addresses are in the range 224.0.0.0 to 239.255.255.255 and IPv6 multicast addresses have the prefix ff00::/8

"Discovery Seed Results" (on page 167)

"Delete Discovery Seeds" (on page 173)

Examine Discovery Results

When verifying discovery, you can do any of the following tasks:

- "Check Initial Progress of Discovery" (on page 166)
- "Verify Success of Discovery Seeds" (on page 167)
- "Examine Discovery Inventory " (on page 169)
- "Examine Layer 2 Discovery Results" (on page 170)
- "Examine Layer 3 Discovery Results" (on page 170)

Related Topics

"Node Discovery State Check" (on page 166)

"Discovery Seed Results" (on page 167)

Check Initial Progress of Discovery

During initial NNMi discovery of your network, you can check Spiral Discovery's progress in the following ways:

- Click **Help** → **System Information** (for more information see Displaying NNMi System Information):
  - Navigate to the **Database** tab to find the real-time list of discovery's progress.
  - Navigate to the **State Poller** tab to see a report of the health of the State Poller Service.
- To see state of discovery for a node, see "Node Discovery State Check" (on page 166).
- NNMi administrators can use the command line on any NNMi management server to generate a report about NNMi health. See the nnmhealth.ovpl Reference Page for more information.

Check this several times during a one hour period. The numbers in the Nodes, SNMP agents, Interfaces, IP addresses, and Layer 2 Connections fields stabilize when initial discovery is complete

Note: If you configure one or more Auto-Discovery Rules and you get unexpected results, check your ordering numbers. See "Configure Auto-Discovery Rules" (on page 141) for more information.

Node Discovery State Check

You can verify the current discovery state for a node.

To see the current Discovery State for a node:

1. Navigate to a **Node** form.
   a. From the workspaces navigation panel, select the workspace of interest. For example, **Inventory**.
   b. Select the node view of interest. For example **Nodes**.
   c. Click the Open icon in the row representing the configuration you want to see.
2. Locate the **Discovery State** attribute (in the Discovery section on the left side of the form).

Possible values include:

- **Newly Created** – Indicates the node and its IP addresses are in the NNMi database, but further information needs to be collected before state and status are determined.
- **Discovery Completed** – Indicates that discovery gathered all required information for the node.
- **Rediscovery in Process** – Indicates discovery is updating the information collected for the node.

Verify Success of Discovery Seeds

The discovery seeds provide the starting point for discovery.

To verify that each discovery seed was successfully discovered:

1. Navigate to the **Discovery Configuration** form.
   - From the workspace navigation panel, select the **Configuration** workspace.
   - Select the **Discovery Configuration**.
2. Locate the **Discovery Seeds** tab.
3. Check the value in the Discovery Seed Results column on each row of the table. A value of **Node Created** indicates the successful discovery of each discovery seed. See "Discovery Seed Results" (on page 167) for the meaning of other values and how to correct discovery problems.

Discovery Seed Results

When you add a discovery seed, the Discovery Service immediately tries to discover it (without waiting until the next regularly scheduled discovery interval). If discovery is not successful, NNMi tries again 10 minutes later, and continues trying. The time between each try is doubled until it reaches 1 week or equals your current discovery interval.

To see the current discovery results for each specified discovery seed:

1. Navigate to the **Discovery Configuration** form
   - From the workspace navigation panel, select the **Configuration** workspace.
   - Select **Discovery Configuration**.
2. Locate the **Discovery Seeds** tab. The table lists each discovery seed and the result that NNMi gathered when contacting the discovery seed.
3. Check the value in the **Discovery Seed Results** column on each row of the table.

Discovery Seed Results Values

Discovery Results	Description
New seed	You just entered a new discovery seed. When discovery begins, Discovery Results changes to "In progress". If the "New seed" value does not change, check to see if the Discovery Service needs to be restarted, see "Verify that NNMi Services are Running" (on page 47).
In progress	Discovery is in progress.
Node created	The discovery seed is successfully discovered and a new Node is created in the database.

Discovery Results	Description

Node created (non-SNMP device)	The hostname or IP address you provided is a non-SNMP device. The Node was discovered and added to the database, but no SNMP information is available because no SNMP agent responded.
	
	If this result is unexpected, the device might currently be down. Initiate an on-demand discovery poll using **Actions → Configuration Poll**, click here for more information. Or try the following:
	
	Check whether the IP address is accessible
	
	1. Type the following command to verify that the address is accessible:
	`ping <nodename>`
	
	Check the Access Control List
	
	1. Access the Node, and open the Access Control List (ACL).
	2. Verify that the NNMi management server address is in the list.
	
	Ensure that SNMP is working
	
	1. Use the nnmsnmpwalk.ovpl command. Type the following to verify that the address has an SNMP agent. Supply one specific MIB variable to limit network traffic to one object rather than requesting all possible SNMP values. For example, use the VendorID prefix with SNMPv1 or SNMPv2c:
	`nnmsnmpwalk -c <communityString> <nodename or IP address> <VendorID>`
	2. If the nnmsnmpwalk.ovpl fails:
	a. Use telnet to check the device's SNMP configuration to verify that SNMP is enabled.
	b. Verify that the address of the NNMi management server is listed in the SNMP Agent's Access list.
	
	Check your communication configuration
	
	1. Verify that SNMP communication is enabled for this device: "Configuring Communication Protocol" (on page 66).
	2. Verify that the device has a properly configured SNMPv1 or SNMPv2c *read community string*, or that the device has a properly configured SNMPv3 USM security setting.
	
	Note: *NNMi makes one attempt to contact each discovery seed.* After you correct the problem that caused NNMi to specify the seed as a non-SNMP device, NNMi updates the Node record during the next discovery cycle. However, this Discovery Results entry does not change, but everything is working properly.

Node not created (DNS name resolution failed)	The Domain Name System (DNS) protocol could not match the hostname you provided for this discovery seed with a valid IP address.
Node not created	The address or hostname you provided is a Node that already exists in the database.

Discovery Results	Description
(duplicate seed)	
Node not created (IPv6 disabled)	The address you provided is an IPv6 address. NNMi Advanced is required, and the IPv6 feature must be enabled. The hostname you provided has only IPv6 addresses. NNMi Advanced is required, and the IPv6 feature must be enabled.
Node not created (IPv6 link local address is invalid seed)	The address you provided is an IPv6 link local address, or the hostname you provided has only one address (an IPv6 link local address). Link local addresses cannot be used as seeds.
Node not created (license exceeded)	Discovery rejected this discovery seed because the number of devices previously discovered reached your licensed capacity limit. See "Extend a Licensed Capacity" (on page 1020).
Failed	Contact with this discovery seed failed due to an internal NNMi error. The problem might be related to discovery or to a system wide issue, such as running out of memory or having trouble with database access. Check the discovery log file (see "Verify that NNMi Services are Running" (on page 47)): • **Windows:** `%NnmDataDir%\log\nnm\nnm.0.0.log` • **UNIX:** `/var/opt/OV/log/nnm/nnm.0.0.log`

Related Topics:

"Specify Discovery Seeds: Initial Routers or Specific Nodes to be Discovered" (on page 158)

Examine Discovery Inventory

The best method for examining your discovered inventory depends on how you configure discovery.

To examine your Discovery Inventory:

1. In the **Workspace** navigation panel, open the **Inventory** workspace.
2. Select the **IP Addresses** view.
3. *Optional*. Verify that each IP address that you identified as a discovery seed is listed.
4. Verify that the IP addresses you expect to see are visible (based on any address ranges where Discover Included Nodes is enabled or disabled).
5. To check on the current discovery state for a particular node, see "Node Discovery State Check" (on page 166).

Note: If you configure one or more Auto-Discovery Rules and you get unexpected results, check your ordering numbers. See "Configure Auto-Discovery Rules" (on page 141) for more information.

Related Topics

Using the IP Addresses View

Using the Nodes View

Examine Layer 2 Discovery Results

Layer 2 represents your network's physical connections and LAN switch traffic routes.

To examine Layer 2 inventory and connectivity results:

1. In the **Workspace** navigation panel, open the **Inventory** workspace.
2. Select the **Nodes** view.
3. Select the node of interest.
4. Select **Actions** → **Layer 2 Neighbor View**.
5. Use the **Number of Hops** field to expand the area shown on the map.
6. Examine your network connectivity to ensure it is as expected. See "Add or Delete a Layer 2 Connection" (on page 176) if changes are required.

Note: If you configure one or more Auto-Discovery Rules and you get unexpected results, check your ordering numbers. See "Configure Auto-Discovery Rules" (on page 141) for more information.

To examine VLAN results:

1. In the **Workspace** navigation panel, open the **Inventory** workspace.
2. Select the **VLANs** view.
3. Select the VLAN of interest.
4. Click Open to open the VLAN form.
5. Verify that the list includes all nodes and ports assigned to this VLAN.

Note: If you configure one or more Auto-Discovery Rules and you get unexpected results, check your ordering numbers. See "Configure Auto-Discovery Rules" (on page 141) for more information.

Related Topics

Using the Layer 2 Neighbor View

Using the Layer 3 Neighbor View

Examine Layer 3 Discovery Results

Layer 3 represents your network's router traffic.

To examine Layer 3 inventory results:

1. In the **Workspace** navigation panel, open the **Inventory** workspace.
2. Select the **Nodes** view.
3. Select the router of interest.
4. Select **Actions** → **Layer 3 Neighbor View**.
5. Use the **Number of Hops** field to expand the area shown on the map.
6. Examine your network connectivity to ensure it is as expected. If changes are required, try the

following:

- Use **Actions** → **Configuration Poll**. See Using Actions to Perform Tasks for more information.
- Manually add or delete the connection. See "Add or Delete a Layer 2 Connection" (on page 176) .
- Verify that the addresses on each end of the connection are not listed in the Excluded IP Address filter. See "Configure an Excluded IP Addresses Filter" (on page 155).

Note: If you configure one or more Auto-Discovery Rules and you get unexpected results, check your ordering number for each rule. See "Configure Auto-Discovery Rules" (on page 141) for more information.

Related Topics

Using the Layer 2 Neighbor View

Using the Layer 3 Neighbor View

Keep Your Topology Accurate

With NNMi, discovery is ongoing. After initial discovery, NNMi checks periodically to ensure that the maps accurately reflect the state of your network. NNMi also updates the database to reflect any changes. See About Map Symbols for an explanation of symbols on the maps.

By default, NNMi uses the following methods to keep the maps accurate:

Spiral Discovery. NNMi uses neighbor information gathered from various devices on your network to discover all devices connected to your network.

Scheduled Rediscovery. Discovery occurs automatically at the interval you define. See"Adjust the Rediscovery Interval" (on page 138) for more information about setting the discovery schedule.

Delete Nodes. As an administrator, you can delete any nodes that you no longer use, or delete nodes to force NNMi to rediscover them. See "Delete Nodes" (on page 172)for more information.

Tip: NNMi also monitors the health of the discovered devices. The health is indicated by the color of the background shape of each device icon on the map. See "Status Colors" for more information. For information about how health monitoring works, see "About the State Poller" (on page 215) and "Monitoring Network Health" (on page 215).

Add or Delete Discovery Seeds. As an administrator, you can delete any Discovery Seeds that you no longer need. See "Delete Discovery Seeds" (on page 173) for more information.

Accurately Detect Interface Changes. If NNMi does not show an accurate list of interfaces in a particular device, you may need to update the settings in the related Device Profile. See "Detect Interface Changes (renumbering issues)" (on page 174).

Add Connections that are not automatically discovered. There are two methods for adding connections:

- ***Subnet Connection Rules*** are ideal for the following situations:
  - Point-to-point or point-to-multipoint connections between interfaces within subnets that have a prefix length ranging from 28-31.
  - IPv4 tunnel or other virtual connections between interfaces within subnets that have a prefix length ranging from 28-31.

NNMi uses Subnet Connection Rules to detect connections between interfaces associated with IPv4 addresses that *do not respond* to Layer 2 Discovery protocols (see the list of Topology Source protocols in Layer 2 Connection Form). Subnet Connection Rules take priority over the Layer 2 Discovery protocol results. For special cases, you can override a Subnet Connection Rule by using the Connection Editor command line tool. See "Configure Subnet Connection Rules" (on page 152) for more information.

- ***Connection Editor*** (nnmconnedit.ovpl command line tool) If your network management domain includes ATM, Frame Relay, or **MPLS**[1] links between wide area networks (WANs), you may need to use the connection editor to show the links in the Layer 2 Neighbor View maps within NNMi. For MPLS, you can provide multiple connections between two nodes. See "Add or Delete a Layer 2 Connection" (on page 176) and the nnmconnedit.ovpl Reference Page for more information.

Delete Connections. Use the Connection Editor (nnmconnedit.ovpl command line tool) to instruct NNMi to ignore certain connections. See "Add or Delete a Layer 2 Connection" (on page 176) and the nnmconnedit.ovpl Reference Page for more information

Delete Nodes

NNMi administrators can delete nodes from a table view, map view, or Node form. For example:

- Remove any nodes that are no longer being used in the network.
- When non-SNMP addresses that had the same DNS hostname are changed to have separate DNS hostnames, NNMi must completely rediscover the non-SNMP nodes to correctly update the database objects (node, interface, address, connection, and incidents).

To understand the results of deleting a Node, click here for more information.

- NNMi cleans up the database by deleting the following objects:
  - Any objects representing things contained in the deleted Node (for example, all of that node's interfaces and IP addresses).
  - Any related objects that are empty after deleting the Node (for example, subnets).
  - Any connections with only zero or one end points after deleting the Node.
- The time required for NNMi to finish deleting depends on the number of objects or related objects being deleted.
- During future discovery cycles, if the deleted Node meets the criteria for an Auto-Discovery Rule and appears in a monitored router's ARP cache, NNMi adds the Node back into the NNMi database during the next discovery cycle. To prevent this, create an Excluded IP Addresses filter for the addresses (see "Configure an Excluded IP Addresses Filter" (on page 155)).
- During future monitoring cycles, NNMi polls only objects currently in the database.
- Each Incident associated with the deleted Node is modified in the following ways, but not deleted from the NNMi database:
  - The **Status** attribute changes to **Closed**.
  - The **Correlation Notes** indicate the deletion of the associated node, interface, or address.
  - The **RCA State** attribute changes to **FALSE**.

[1]Multiprotocol Label Switching

Note: Incidents generated from SNMP traps or NNM 6.x/7.x Events (received from the deleted Node) appear in the Incident views, but remain unresolved.

- If you are viewing a Node that has recently been deleted by another user, the deleted Node appears as a transparent icon on the map until the map is refreshed using the **Refresh** icon. After **Refresh**, the deleted node is removed from the map. NNMi does not automatically refresh the connectivity or set of nodes in a map view, except on the Network Overview map.

Note: If you delete a Node with many interfaces and VLANs, you may see an error message indicating that the Node could not be deleted. This means the database was busy with discovery. Try again between discovery cycles.

If a deleted Node is one of your seeds, delete that seed from the Discovery Seeds table as well. See "Delete Discovery Seeds" (on page 173).

To delete one or more nodes (maximum 20 at one time):

1. Unmanage the nodes you want to delete.
    a. In a table view, select the nodes of interest by selecting the check box in the row or rows that represents each node you want to unmanage.
    b. Select **Actions** → **Management Mode** → **Unmanage**.
    c. Wait until the Status=*No Status* for each of the following objects:
        - Each Node to be deleted
        - Each Node's Interfaces, IP Addresses, Cards, Ports, and VLAN Ports
2. Do one of the following:
    - *Table views*: select the object of interest by selecting the check box in the row or rows that represents the objects of interest, and click the Delete icon. Each selected node is deleted from the NNMi database and removed from the current view.
    - *Map views*: click the map symbol representing the node you want to delete, and click **File** → **Delete Node**. The node is deleted from the NNMi database and removed from the current view.
    - *Node form*: select **File** → **Delete Node** and in the confirmation dialog, click **OK**. The form is automatically closed after NNMi deletes the Node.

    Note: If the delete fails, use the nnmnodedelete.ovpl command. Wait for the command to complete.

To delete any number of nodes:

Use the `nnmnodedelete.ovpl` command. See the nnmnodedelete.ovpl Reference Page.

Related Topics

Using Table Views

Using Map Views

Delete Discovery Seeds

There are two ways to delete discovery seeds from the NNMi Discovery configuration and the NNMi database.

Note: If you remove a Discovery Seed from Discovery Configuration, the corresponding node is not deleted from the topology database. See "Delete Nodes" (on page 172) for information about removing the entire node record from the topology database.

To delete seeds using the Discovery Configuration view:

1. Navigate to the **Discovery Seeds** tab in the Discovery Configuration form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Discovery Configuration**.
    c. Locate the **Discovery Seeds** tab.
2. To delete one or more discovery seeds, select any number of rows, and click the Delete icon.
3. Click **Save and Close**.

To delete any number of seeds at one time from the command line:

At the command line of the NNMi management server, type the nnmseeddelete.ovpl command.

Case-sensitive exactly as listed in the **Discovery Seeds** tab in the Discovery Configuration form, specify the hostname or IP address.

If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access" (on page 275) for more information.

```
nnmseeddelete.ovpl -seed <hostname/IP-address> -u <NNMiadminUserName> -p
<NNMiadminPassword>
```

Detect Interface Changes (renumbering issues)

Sometimes the hardware configuration of a device in your network changes (for example, adding a card with 20 interfaces to a router). When someone installs or removes interfaces from a device in your network:

- Some devices maintain a static list of MIB II `IfIndex` numbers. When interfaces are added, MIB II `IfIndex` numbers are added to the end of the current list of interfaces contained in that device. When interfaces are removed, the MIB II `IfIndex` numbers previously used by those interfaces are dropped from the list.
- Some devices reset all MIB II `IfIndex` numbers for the group of interfaces contained in that device each time a change occurs. Each manufacturer has a different strategy for identifying each interface and detecting when an existing interface is simply assigned to a different MIB-II `IfIndex` number or an interface is removed.

The NNMi administrator chooses which strategy is used for each manufacture/model. NNMi can use any of the following MIB-II objects to track the group of interfaces within a device. These MIB-II objects are known as **Interface Reindexing Types** in the Device Profile form. See Device Profile Form for more information.

Interface Reindexing Types

Interface Reindexing Type	Description
`ifAlias` value	Compares the `ifAlias` value on the interface with the discovered `ifAlias` value.
`ifDescr` value	Compares the `ifDescr` value on the interface with the discovered `ifDescr` value.

Interface Reindexing Type	Description
`ifIndex` value **Note**: Use `ifIndex` only for manufacturers/models that maintain a static `ifIndex` list.	Detects when an `ifIndex` value no longer exists. This setting does not detect interface renumbering.
`ifName` value	Compares the `ifName` value on the interface with the discovered `ifName` value.
Combination of `ifDescr` and `ifName` values	Compares the `ifDescr` and `ifName` values on the interface with the discovered `ifDescr` and `ifName` values.

The NNMi State Poller uses the Interface Reindexing Types described above for those interfaces for which **SNMP Interface Polling** is enabled. See"Configure Interface Monitoring" (on page 223) for more information. The Interface Reindexing Type determines the Device Profile value that the State Poller compares. If the value collected by the State Poller is different than the discovered value, State Poller stops polling the node and requests that the node be rediscovered. After the node is rediscovered, the State Poller starts to poll the node again.

If you know that hardware changes were made for interfaces in your network environment, but NNMi does not accurately reflect those changes as they happen, do the following:

1. Identify the manufacturer and model of the device with an interface setup that changed. Check the device's documentation to find out which MIB-II values are used to provide a positive identification for each interface.
2. Navigate to the appropriate **Device Profile** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Device Profiles** view.
   c. Click the **Device Model** column heading to sort the list by manufacturer's name.
   d. Locate the group of that manufacturer's devices.
   e. Click the Open icon in the row representing the configuration you want to see.
3. Locate the **Interface Reindexing Type** attribute, and click the drop-down list to display the available choices (see Interface Reindexing Type in the Device Profile Form help).
4. Select the appropriate MIB-II values for NNMi to use when determining whether an interface has moved and been renumbered, an interface was recently added, or an interface was removed.
5. Click **Save and Close** to return to the **Device Profile** view.
6. To verify that the problem is solved, do one of the following:
   - For immediate results, navigate to a Node view and select one of the problem devices.

     Click **Actions** → **Configuration Poll** to instruct NNMi to rediscover the information about interfaces in that device.

     Open the Node form for the device and verify that the list interfaces is correct.

- NNMi updates interface information for all matching manufacture/model devices in your network environment the next regularly scheduled Discovery or Monitoring cycles.

Add or Delete a Layer 2 Connection

Tip: If your network management domain includes ATM, Frame Relay, or Multi-Protocol Label Switching (MPLS) links between wide area networks (WANs), you may need to use the connection editor to show the links in the Layer 2 Neighbor View maps within NNMi. For MPLS, you can provide multiple connections between two nodes.

Use the NNMi nnmconnedit.ovpl command to add or delete connection data.

The `nnmconnedit.ovpl` command is used to generate a template XML file (shown in the following example). For each connection to be added or deleted, you provide information about the node and interface at both ends of the connection. Multiple `<connection>` elements are allowed within the template XML file.

```
<connectionedits>
 <connection>
    <operation>add or delete</operation>
    <node>node Name, Hostname or management IP address</node>
    <interface>ifName, ifAlias, ifDescr or ifIndex</interface>
    <node>node Name, Hostname, or management IP address</node>
    <interface>ifName, ifAlias, ifDescr or ifIndex</interface>
  </connection>
 </connectionedits>
```

Required Layer 2 Connection Attributes in the Connection Editor File

Attribute	Description
operation	Specify whether the connection is to be added or deleted.
node	Identify the node using any of the following *case-sensitive* values: • `node Name` • `Hostname` (*case-sensitive*) NNMi follows a set of rules to dynamically generate the value stored in the NNMi database for each Node's Hostname. Click here for details. **Note**: The actual Hostname *might be converted* to all uppercase or all lowercase before it is added to the NNMi database (depending on how the NNMi administrator configured settings in the `nms-topology.properties` file). See the information

Attribute	Description

about the `nms-topology.properties` file in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

- If the Node supports SNMP, NNMi requests the Hostname using the IP Address of the associated SNMP agent (the Management Address attribute value on the Node form).

  If the NNMi administrator chooses ☑ **Enable SNMP Address Rediscovery** in the Communication Configuration:

  - If the SNMP Agent does not respond, NNMi checks for another Management Address to request the Hostname, and the Hostname could change.
  - If the SNMP Agent associated with the node changes, the Management Address and Hostname could change.

  If the NNMi administrator disables ☐ **Enable SNMP Address Rediscovery** in the Communication Configuration:

  - If the SNMP Agent does not respond, NNMi uses the previously gathered Management Address attribute value to request the Hostname.
  - If the SNMP Agent associated with the node changes, NNMi uses the previously gathered Management Address attribute value to request the Hostname.

- If the Node does not support SNMP, no Management Address is available. NNMi requests a Hostname starting with the lowest IP Address associated with the node (a Discovery Seed value or an IP address value gathered from a neighboring device). NNMi uses the first Hostname provided. The Hostname might change during a future discovery cycle.

- `management IP address`

  NNMi follows a set of rules to determine which address is the best choice as the node's Management Address. Click here for details.

  Note: With NNMi Advanced, the NNMi administrator specifies whether NNMi prefers IPv4 or IPv6 addresses when selecting the Management Address. See the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`

  a. NNMi ignores the following addresses when determining which Management Address is most appropriate:

     - Any address of an administratively-down interface.

Attribute	Description
	○ Any address that is virtual (HSRP/VRRP). ○ Any IPv4 **Anycast Rendezvous Point IP Address**[1] or IPv6 Anycast address. ○ Any address in the reserved loopback network range. IPv4 uses 127/24 (`127.*.*.*`) and IPv6 uses `::1`. ○ Any IPv6 **link-local address**[2]. b. NNMi prefers loopback interfaces for management communication. ○ If a node has only one **loopback address**[3], and the associated SNMP agent responds to NNMi, that address becomes the Management Address. ○ If a node supports multiple loopback addresses, NNMi queries each loopback addresses, starting with the lowest number. NNMi uses the loopback address with the lowest number from which the SNMP agent responds (for example, 10.16.42.197 is a lower number than 10.16.197.42). c. If no loopback address responds, NNMi queries the last-known Management Address (if any). d. If no response, NNMi queries up to 10 of any remaining IP addresses in the node's IP address inventory, starting with the lowest number. NNMi uses the address with the lowest number from which the SNMP agent responds. e. If no response, NNMi might be configured to repeat the sequence using SNMPv1, SNMPv2c, or SNMPv3 in the order specified by the NNMi administrator (Communication Configurations *SNMP Minimum Security Level* settings). f. When all else fails, NNMi retains the last known Management Address (if any) and automatically changes the State of that SNMP Agent object to Critical. This process is repeated during each Auto-Discovery cycle, and the Management Address can change. For example, NNMi's inventory of addresses for the node expands, or the current Management Address does not respond to SNMP queries due to network problems or node reconfiguration. The NNMi administrator can prevent changes to the management address using the Communication Configurations *Enable SNMP Address Rediscovery* or *Preferred Management Address* setting.

[1]Rendezvous Point addresses are loopback addresses used for routers in multi-cast network configurations.
[2]A non-routable IPv6 unicast address only used for communication with other nodes on the same link (LAN or VLAN). Link local addresses cannot be used for communication that must be forwarded through a router. IPv6 auto-configuration automatically assigns a unique link local address in the fe80::/10 address space to each IPv6-enabled interface on a system.
[3]The address associated with the loopback interface. The loopback interface is a virtual interface on a device that provides a route for internal communication. Many vendors provide a specially configured loopback for management purposes. Exact details of how loopbacks are configured varies by vendor and model. See each device's documentation for details. NNMi identifies these loopback addresses by using IfType 24, softwareloopback from the IANA ifType-MIB.

Attribute	Description

interface	Identify the interface using one or more of the following (MIB-II) values:
	• `ifName`
	• `ifAlias`
	• `ifDescr`
	• `ifIndex` Note the following for `ifIndex`:
	  ▪ For interfaces in Non-SNMP nodes, always use the `ifIndex` value of 0 (zero).
	  ▪ For interfaces in SNMP nodes, choose other MIB-II values to identify the interface because often automatic interface renumbering causes confusion.

To add or delete a connection:

1. For the devices at both ends of the connection, gather the data required to identify the device and interface.
2. On the NNMi management server, at the command line, generate a connections template file using either `add` to create an add.xml template file or `delete` to create a delete.xml template file.

   In the following example, NNMi creates an add.xml file:
   ```
   nnmconnedit.ovpl -t add
   ```

   Note: If you specify add, NNMi creates the template file named `add.xml`. If you use delete, the template file is named `delete.xml`.
3. Open the template file in a text editor and fill in the correct information for each node and interface.
4. On the NNMi management server, at the command line, load the new connection information into the NNMi database:
   ```
   nnmconnedit.ovpl -f <add|delete>.xml
   ```

   For example, to load the add.xml template file, enter:

   ```
   nnmconnedit.ovpl -f add.xml
   ```
5. Open the Layer 2 Neighbor View map and verify the connection changes.

The connections you establish are listed in the Layer 2 Connections view in the Inventory workspace. To delete a connection, you must use the nnmconnedit.ovpl command (no Delete action is available in the Layer 2 Connections view).

Start Discovery On-Demand

NNMi provides the nnmnoderediscover.ovpl command line tool for initiating discovery. This tool allows NNMi administrators to do the following:

- Run discovery of a subset of your network domain to get the most recent data without waiting for the next-regularly schedules discovery cycle.

  For example, to immediately add newly deployed critical devices to the NNMi database without waiting for the next regularly-scheduled discovery cycle.
- Run discovery of your entire network on demand or using an automation script.
- Request updated discovery results from the Regional Managers in your network environment after

restoring the Global Manager to a previous state.

(*NNMi Advanced - Global Network Management feature*) Any change to the Node's Management Mode setting is immediately sent from a Regional Manager (NNMi management server) to the Global Manager. (Changes to Management Mode for other objects are sent during the next Auto-Discovery cycle on the Regional Manager.)

Note: This tool can help you synchronize the Global Manager if for some reason the original information from the Regional Managers is lost from the Global Manager's database.

See nnmnoderediscover.ovpl for more details.

Creating Groups of Nodes or Interfaces

Groups of nodes or interfaces are used for a variety of purposes within NNMi. Use of these groups is optional.

- Use node and interface groups to create custom view filters that help your team quickly sift through data in the NNMi views and identify the most important information. See Filter Views by Node or Interface Group.
- Special Actions are available for Node Groups and Interface Groups.
- Use Node Groups and Interface Groups to specify monitoring configuration settings. See"Monitoring Network Health" (on page 215). For example, configure a different health monitoring interval for each group.
- (*NNMi Advanced - Global Network Management feature*) On a Regional Manager, use Node Groups to limit the amount of data available to Global Managers in your network environment. See "Regional Manager: Create a Forwarding Filter (Limit the available Node information)" (on page 53) for more information.
- (*NNM iSPI Performance*) If you are using the HP Network Node Manager iSPI Performance for Metrics Software or HP Network Node Manager iSPI Performance for Traffic Software software, control performance monitoring and provide report filters by Node Group.

Once Node Groups or Interface Groups are defined, you can reuse them within any context (view filtering and NNMi configuration settings) or you can configure them to be hidden from the view filter lists.

View Filter Possibilities

Filter	Available in NNMi views based on: Object Type					
	Incident	Node	Interface	IP Address	Card	Node Component
Node Groups "Create Node Groups" (on page 181)	x	x	x	x		
Interface Groups "Create Interface Groups" (on page 197)			x	x	x	x

Create Node Groups

Node Groups are used for a variety of purposes in NNMi. See "Creating Groups of Nodes or Interfaces" (on page 181) for more information.

You can create any number of Node Groups in addition to the ones that NNMi provides (see "Node Groups Provided by NNMi" (on page 209)).

Node Group definitions match the way your team identifies important network devices. Each node group is defined using one or more of the following:

- Device Filters (by any combination of SNMP device category, vendor, family, profile)
- Additional Filters (Boolean expressions based on a list of object attributes)

- Additional Nodes (identified by *case-sensitive* Hostname)
- Child Node Groups (use any combination of Node Groups to create a filter)

NNMi combines the results of all Node Group configuration settings in the following manner:

- NNMi first evaluates Device Filters. If any exist, nodes must match *at least one* specification to belong to this Node Group.
- NNMi then evaluates any Additional Filters. Nodes *must also pass all* Additional Filters specifications to belong to this Node Group.
- Any Additional Nodes specified are *always* included in the Node Group, irregardless of any filters.
- Any Child Node Group results are treated the same as Additional Nodes.

To create Node Groups, do one or more of the following:

- "In the Console, Create Node Groups" (on page 182)
- "In a CSV File, Define Node Groups" (on page 196)

To verify the contents of a Node Group:

After the Node Group is saved, from the Node Group form, select **Actions** → **Show Members**. Special Actions are available for Node Groups and Interface Groups.

In the Console, Create Node Groups

Node Groups are used for a variety of purposes in NNMi. See "Creating Groups of Nodes or Interfaces" (on page 181) for more information.

You can create any number of Node Groups in addition to the ones that NNMi provides (see "Node Groups Provided by NNMi" (on page 209)).

To create a Node Group (if your role allows you to do this):

1. Navigate to the **Node Group** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Node Groups** view.
   c. Do one of the following:
      - To create a Node Group, click the New icon.
      - To edit a Node Group, click the Open icon in the row representing the Node Group you want to edit.
2. In the Node Group form, provide the required information in the Basics section.
3. (*NNM iSPI Performance*) Make the Node Group available within NNM iSPI Performance products (see NNM NNM iSPI Performance table).
4. Identify the nodes that belong to this Node Group.

   Do one or more of the following:

   - Specify a filter based on Device Profile settings using the Device Filters tab (any combination of category, vendor, family, or profile).

Tip: To base your filter on the SNMP system Object ID number, use the Additional Filters `sysOidNode` code.

- Specify a Node Group filter using the Additional Filters tab (use a variety of available codes to filter by object attribute values in the NNMi database).
- Specify individual nodes using the Additional Nodes tab (provide a list of Hostnames, as they appear in the NNMi database).
- Specify Child Node Groups using the Child Node Groups tab (use combinations of Node Groups to create a filter).

5. Click **Save and Close** to return to the Node Group form.
6. Click **Save and Close**.

   If you configured this Node Group for Monitoring, NNMi applies your changes during the next monitoring cycle. "Configure Monitoring Behavior" (on page 216).

To review a Node Group definition:

1. From the workspace navigation panel, select the **Inventory** workspace.
2. Select the **Node Groups** view.
3. Locate the row representing the Node Group settings you want to see and click the Open icon.
4. The Node Group form displays.

   Note: NNMi monitors the status of each Node Group over time. To check Node Group status information, access the Node Group form's Status tab.

5. When finished, click the Close icon.

Special Actions are available for Node Groups and Interface Groups.

Related Topics

"In a CSV File, Define Node Groups" (on page 196)

Specify Node Group Additional Filters

Use the Additional Filters Editor to create expressions that refine the requirements for membership in a Node Group. Make sure to design any complex Additional Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Additional Filters Editor.

If any Additional Filters are created, NNMi combines any Device Filters and Additional Filters using the AND Boolean operator as follows:

- NNMi first evaluates any Device Filters. Nodes must match *at least one* Device Filter specification to belong to this Node Group.
- NNMi then evaluates the Additional Filters expression. Nodes *must also match all* Additional Filters expression specifications to belong to this Node Group.

To create an Additional Filters expression:

1. Navigate to the **Node Group Form: Additional Filters** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.

b. Select **Node Group**.

c. Do one of the following:

- To create a Node Group definition, click the New icon.
- To edit a Node Group definition, click the Open icon in the row representing the Node Group definition you want to edit.

d. In the Node Group form, select the **Additional Filters** tab.

2. Establish the appropriate settings for the Additional Filters you need (see the Additional Filters Editor Components and Additional Filters Editor Buttons table). See "Guidelines for Creating Additional Filters for Node Groups" (on page 191) for more information.

a. Plan out the logic needed for your Filter String.

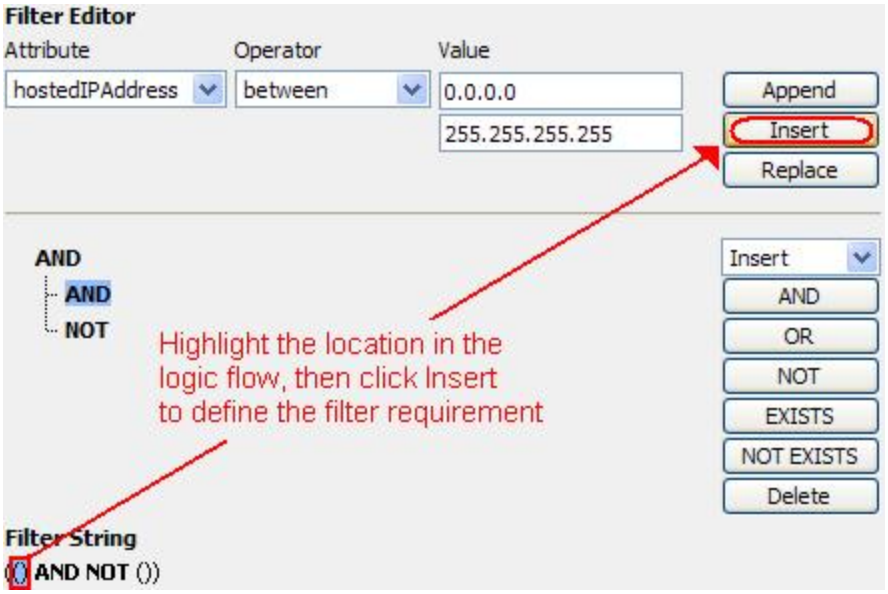
b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure. See "Add Boolean Operators in the Additional Filters Editor" (on page 193).

For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

```
(( ) AND NOT ( ))
```

c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the selected filter requirement.

For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



3. Click **Save and Close**.

Additional Filters Editor Components for Node Groups

Attribute	Description
Attribute	NNMi provides Additional Filters codes for a subset of object attributes. For more information about the available Additional Filter codes for each NNMi object type, click the link: • Node attribute codes [click here for a list of attribute codes]

Attribute	Description
	Values from the Basic Attributes listed on the Node Form: ■ hostname (Hostname, *case-sensitive*) ■ mgmtIPAddress (Management Address) ■ isSnmpNode (Agent Enabled) ■ isNnmSystemLocal (NNMi Management Server) **Values from the Node Form:General Tab**: ■ sysName (System Name) ■ sysLocation (System Location) ■ sysContact (System Contact) ■ sysOidNode (System Object ID) **Addresses from the Node Form: IP Addresses Tab**: ■ hostedIPAddress (Address) See "Node Groups of IPv4 or IPv6 Addresses " (on page 190) for ideas. **Unique Keys from the Node Form: Capabilities Tab**: ■ capability (Unique Key of the Capability) **Values from the Node Form: Custom Attributes Tab**: **Note**: When using `customAttrName`and `customAttrValue` pairs, use EXISTS if you want NNMi to consider Nodes that *do not have Custom Attributes* when evaluating the entire Filter String. Otherwise Nodes that do not have Custom Attributes are automatically excluded from the Node Group even if they have values that pass other aspects of your filter. ■ customAttrName (Custom Attribute Name) ■ customAttrValue (Custom Attribute Value) • Device Profile attribute codes [click here for a list of attribute codes] **Values from the Basics Attributes on the Device Profile Form:** NNMi matches the Label attribute values from the Device Profile Form for each of the following: ■ devCategoryNode (Device Category) ■ devVendorNode (Device Vendor) ■ devFamillyNode (Device Family) To filter on the SNMP system object ID number assigned to a particular make/model, use the sysOidNode attribute. See Values from the Node Form: General Tab. • Regional Manager attribute codes [click here for a list of attribute codes] **Values from the associated entry on the Regional Manager Form: Connection Tab**: ■ nnmSystemName (Hostname, *case-sensitive*) (*NNMi Advanced*) If the Global Network Management feature is enabled, this attribute

Attribute	Description
	value identifies a Regional Manager (NNMi management server).
Operator	The standard query language (SQL) operations to be used for the search. **Note**: Only the `is null` Operator returns null values in its search. Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `sysName = cisco2811` finds all devices with system name equal to **cisco2811**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `sysName != cisco2811` finds all system names other than **cisco2811**. • **<** Finds all values less than the value specified. Click here for an example. IPv4 example: `mgmtIPAddress < 15.239.255.255` finds all IP address values less than **15.239.255.255** IPv6 example: `mgmtIPAddress < ::ffff:0:0` finds all IP address values less than **::ffff:0:0** • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `mgmtIPAddress <= 15.239.255.255` finds all IP address values less than or equal to **15.239.255.255**. • **>** Finds all values greater than the value specified. Click here for an example. IPv4 example: `mgmtIPAddress > 15.238.0.0` finds all IP address values greater than **15.238.0.0** IPv6 example: `mgmtIPAddress > ::ffff:ffff:ffff` finds all IP address values greater than **::ffff:ffff:ffff** • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `mgmtIPAddress >= 15.238.0.0` finds all IP address values greater than or equal to **15.238.0.0**. • **between** Finds all values equal to and between the two values specified. Click here for an example. Example: `mgmtIPAddress between 15.238.0.10 15.238.0.120` finds all IPv4 address values equal to or greater than **15.238.0.10** and equal to or less than **15.238.0.120**. See "Node Groups of IPv4 or IPv6 Addresses " (on page 190) for more examples of using the **between** Operator. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `sysName in`

Attribute	Description

finds all systems with names that are **cisco2811** or **cisco5500**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**cisco2811, cisco550**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `sysName is not null` finds all systems that have a name value.

- **is null** Finds all blank values. Click here for an example.

  Example: `sysName is null` finds all systems that do not have an assigned name value.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The following attributes cannot be used with the `like` operator:

  - hostedIPaddress
  - mgmtIPaddress

  The asterisk (*) character means *any number of characters of any type at this location*.

  Note: For optimum performance, avoid beginning your search string with an asterisk (*).

  The question mark (?) character means *any single character of any type at this location*.

  Examples:

  - `sysName like cisco*` finds all system names that begin with **cisco**.
  - `sysName like cisco??*` finds all system names that *start with* cisco **followed by two characters**.
  - `sysName like rtr??bld5*` finds all system names that have *specific characters at an exact location*, positions 1-3 (rtr) and 6-9 (bld5).

- **not between** finds all values except those between the two values specified. Click here for an example.

  Example: `mgmtIPAddress not between 15.238.0.10 15.238.0.120` finds all IP address values less than **15.238.0.10** and greater than **15.238.0.120**.

  See "Node Groups of IPv4 or IPv6 Addresses " (on page 190) for more examples of using the **not between** Operator.

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `sysName not in`

  finds all system name values other than **cisco2811** and **cisco5500**.

  Note: As shown in the example, each value must be entered on a separate line.

Attribute	Description
	NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**cisco2811, cisco550**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The following attributes cannot be used with the `not like` operator: ▪ hostedIPaddress ▪ mgmtIPaddress The asterisk (*) character means *any number of characters of any type at this location*. The question mark (?) character means *any single character of any type at this location*. Examples: ▪ `sysName not like cisco*` finds all system names that do not begin with **cisco**. ▪ `sysName not like cisco??*` finds all system names that do not *begin with* cisco **followed by two characters**. ▪ `sysName not like rtr??bld5*` finds all system names that do not have *specific characters at an exact location*, positions 1-3 (rtr) and 6-9 (bld5).

Attribute	Description
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `in` and `not in` operators require that each value be entered on a separate line. • When entering a value for the Capability attribute, copy and paste the Unique Key value from the Node form: Capability tab.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude nodes with values that pass the expression that immediately follows the NOT. For example, when evaluating the following Filter String, NNMi includes nodes with a hostname that contains **router**, followed by any number of characters, followed by **hp.com** and excludes any nodes with a Device Profile that includes **Cisco** as the Vendor value: `(hostname like router*.hp.com OR NOT (devVendorNode = Cisco))`
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider nodes that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use EXISTS or NOT EXISTS, NNMi excludes from its search nodes that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes nodes with a hostname that includes **router**, followed by any number of characters, followed by **hp.com** as well as any nodes that have the Custom Attribute **edge** and that edge value is **true**: `(hostname like router*.hp.com OR EXISTS((customAttrName=edge AND`

Button	Description
	`customAttrValue=true)))`
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider nodes that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the nodes that match the expression that follows the NOT EXISTS. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use EXISTS or NOT EXISTS, NNMi excludes from its search nodes that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes nodes with a hostname that includes **router**, followed by any number of characters, followed by **hp.com** and excludes any nodes with Custom Attribute **edge** and that edge value is **true**. `(hostname like router*.hp.com OR NOT EXISTS((customAttrName=edge AND customAttrValue=true)))`
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Node Groups of IPv4 or IPv6 Addresses

Use the Node Group form's Additional Filters editor to create Node Groups based on the following criteria ("Specify Node Group Additional Filters" (on page 183)):

- All nodes that have *only* IPv4 addresses
  [click here for details of this filter.]

  Both of the following example Node Group's Additional Filters provide the same results. The first example uses IPv4 address notation. The second example uses IPv6 address notation:

  ```
  ((hostedIPAddress between 0.0.0.0 AND 255.255.255.255) AND NOT (hostedIPAddress not between 0.0.0.0 AND 255.255.255.255))
  ```

  or (*NNMi Advanced with IPv6 enabled*)

  ```
  ((hostedIPAddress between 0.0.0.0 AND 255.255.255.255) AND NOT (hostedIPAddress not between ::ffff:0:0 AND ::ffff:ffff:ffff))
  ```

- All nodes that have *any* IPv4 addresses (could also have IPv6)
  [click here for details of this filter.]

  The following example Node Group's Additional Filter finds any node that has at least one IPv4 address:

  ```
  (hostedIPAddress between 0.0.0.0 AND 255.255.255.255)
  ```

- (*NNMi Advanced with IPv6 enabled*) All nodes that have *only* IPv6 addresses
  [click here for details of this filter.]

  IPv6 addresses extend the number of possible IP addresses. The old IPv4 address range falls within the new IPv6 range. Valid IPv6 address values can be less than or greater than the old IPv4 range of

addresses. NNMi Advanced converts the IPv4 addresses to the new IPv6 notation, then stores and filters the IPv4 addresses as IPv6 addresses (`::ffff:a.b.c.d`).

Both of the following example Node Group's Additional Filters provide the same results. The first example uses IPv4 address notation. The second example uses IPv6 address notation:

```
((hostedIPAddress not between 0.0.0.0 AND 255.255.255.255) AND NOT
(hostedIPAddress between 0.0.0.0 AND 255.255.255.255))
```

or

```
((hostedIPAddress not between ::ffff:0:0 AND ::ffff:ffff:ffff) AND NOT
(hostedIPAddress between 0.0.0.0 AND 255.255.255.255))
```

- (*NNMi Advanced with IPv6 enabled*) All nodes that have *any* IPv6 addresses (could also have IPv4) [click here for details of this filter.]

  The following example Node Group's Additional Filter finds any node that has at least one IPv6 address:

  ```
  ((hostedIPAddress between ::0 AND ::fffe:ffff:ffff) OR (hostedIPAddress
  ::1:0:0:0 AND ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff))
  ```

- (*NNMi Advanced with IPv6 enabled*) All nodes that have *both* IPv4 and IPv6 addresses (also known as dual-stack nodes)
  [click here for details of this filter.]

  The following example Node Group's Additional Filter finds any node that has at least one IPv4 address and at least one IPv6 address:

  ```
  ((hostedIPAddress between 0.0.0.0 AND 255.255.255.255) AND (hostedIPAddress not
  between 0.0.0.0 AND 255.255.255.255))
  ```

Note: To maximize the performance of Additional Filters based on an IP Address range, avoid multiple filter expressions. For example, use the `between` operator instead of the greater than or equal to (>=) and less than or equal to (<=) operators that cause NNMi to use multiple queries for finding all addresses that match the filter.

Guidelines for Creating Additional Filters for Node Groups

The Additional Filters Editor enables you to create expressions to further define the nodes to be included in a Node Group. Make sure to design any complex Additional Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Additional Filters Editor.

When creating Additional Filters for a Node Group, note the following:

- NNMi treats each set of expressions associated with a Boolean Operator as if it were enclosed in parentheses and evaluated together rather than in order of grouping as the nesting implies. Therefore, when using the `AND` operator to combine expressions that include Custom Attributes, include only one customAttrName/customAttrValue pair in the expression. Otherwise, if you use multiple customAttrName and customAttrValue pairs with the `AND` operator, the results might not be as expected. Click here for an example.

  In the following example, because the `AND` Boolean operator indicates that NNMi should evaluate all of

the `customAttrname` and `customAttrvalue` pairs together, it is not possible for any nodes to match this Additional Filters expression:

Additional Filter Expression Example 1:

```
((customAttrName = capability) AND (customAttrValue =
com.hp.nnm.capability.card.fru)) AND ((customAttrName = location) AND
 (customAttrValue = datacenter1))
```

This is because `customAttrName` would need to match both `capability`*and* >`location` at the same time. However, if you use the `OR` operator to combine the `customAttrName` and `customAttrValue` pairs as shown in the following example, the filter should work as expected.

Additional Filter Expression Example 2:

```
((customAttrName = capability) AND (customAttrValue =
com.hp.nnm.capability.card.fru)) OR ((customAttrName = location) AND
 (customAttrValue = datacenter1))
```

Using the Node values listed in the following table, all three nodes (nodeA, nodeB, and nodeC) pass the filter in Example 2 because each of these nodes has either the value `com.hp.nnm.capability.card.fru` for `capability`*or* the value `datacenter1` for `location`.

Example Data

Node Name	capabilty	customAttrName	customAttrValuee
nodeA	com.hp.nnm.capability.card.fru	location	datacenter1
nodeB	com.hp.nnm.capability.card.fru	<undefined>	<undefined>
nodeC	<undefined>	location	datacenter1

- Use the EXISTS and NOT EXISTS operators when you want NNMi to consider nodes that do not have any Capabilities or Custom Attributes when evaluating the Filter String. See "Specify Node Group Additional Filters" (on page 183) for more information.
- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The AND and OR Boolean Operators must contain at least two expressions as shown in the example below.

```
AND
  sysName like cisco*
  sysName != cisco2811
  OR
     sysLocation = Boston
     sysContact In (Johnson,Hickman)
```

NNMi evaluates the expression above as follows:

```
sysName like cisco* AND sysName != cisco2811 AND (sysLocation = Boston OR
sysContact in (Johnson, Hickman))
```

  - NNMi finds all nodes with a (system name) sysName beginning with **cisco**, except not **cisco2811**.
  - Of these nodes, NNMi then finds all nodes with a (system location) sysLocation of **Boston** or (system contact name) sysContact of **Johnson** or **Hickman**.

- NNMi evaluates only those nodes that contain values for *all* of the attributes included in the Additional Filter expression. Click here for an example.

  If your Node Group filter expression includes the `capability` and `customAttrName` attributes, then NNMi evaluates only nodes that have a value defined for *both* `capability` and `customAttrName`. For example, if you create a Node Group using the following Additional Filters expression, then NNMi evaluates only those nodes that have a value defined for `capability` and a value defined for `customAttrName`:

  ```
  (capability = com.hp.nnm.capability.card.fru) OR (customAttrName = location)
  ```

  Using the Node values listed in the following table, NNMi only evaluates nodeA. This is because nodeA contains a value for `capability` and a value for `customAttrName`. NNMi does not evaluate nodeB because it does not have a value for `customAttrName`. NNMi does not evaluate nodeC because it does not have a value for `capability`. NodeA also passes Node Group Additional Filter because its `capability` value of `com.hp.nnm.capability.card.fru` matches the value specified in the Additional Filter expression. Therefore, only nodeA is included in this example Node Group.

  Example Data

Node Name	capabilty	customAttrName	customAttrValuee
nodeA	com.hp.nnm.capability.card.fru	location	datacenter1
nodeB	com.hp.nnm.capability.card.fru	<undefined>	<undefined>
nodeC	<undefined>	location	datacenter1

  Tip: You can populate a placeholder value, such as "none" or "undefined" for any attribute that you want to use in an Additional Filter.

- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Additional Filters Editor. For example, you append to or replace, the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators. See "Add Boolean Operators in the Additional Filters Editor" (on page 193) for more information.
- You can drag any of the following items to a new location in the Filter String:
  - Filter Editor Options: AND, OR, NOT, EXISTS, NOT EXISTS
  - Filter Expression (Attribute, Operator and Value)
- When moving items in the Filter String, note the following:
  - Click the item you want to move before dragging it to a new location.
  - As you drag a selected item, an underline indicates the target location.
  - If you are moving the selection up, NNMi places the item above the target location.
  - If you are moving the selection down, NNMi places the item below the target location.
  - If you attempt to move the selection to an invalid target location, NNMi displays an error message.

Add Boolean Operators in the Additional Filters Editor

When adding or deleting Boolean Operators using the Additional Filters Editor, note the following:

- Add your highest level Boolean operator first. For example, **AND** is the highest level Boolean operator in the following expression

  (sysName like cisco* OR sysName like hp*) **AND** (sysLocation = Boston OR sysContact in Johnson,Hickman)

- Add each additional Boolean Operator before the expressions to which it applies.
- Select the appropriate Boolean Operator in the expression before you add the expressions to which the Boolean Operator applies.
- When a Boolean Operator is selected and you click **Delete**, any expressions that are associated with the Boolean Operator are also deleted.

  In the example expression below, If you select **AND** and then click **Delete,** the Additional Filters Editor deletes the entire expression.

Click here for an example for creating Node Group Additional Filters.

Node Group Additional Filters Expression Example

```
((sysName like cisco* OR sysName like hp*) AND (sysLocation = Boston OR sysContact
in (Johnson, Hickman)))
```

To add the expression above, after you are in the Additional Filters Editor, follow these steps:

1. Click **AND**.
2. Click **OR**.
3. Select the **OR** you just added to the expression.
4. In the **Attribute** field select **sysName** from the drop-down list.
5. In the **Operator** field, select **like** from the drop-down list.
6. In the **Value** field, enter **cisco***.
7. Click **Append**.
8. In the **Attribute** field, select **sysName** from the drop-down list.
9. In the **Operator** field, select **like** from the drop-down list.
10. In the **Value** field, enter **hp***.
11. Click **Append**.
12. Select the **AND** that you previously added to the expression.
13. Click **OR**.
14. Select the **OR** you just added to the expression.
15. In the **Attribute** field, select **sysLocation** from the drop-down list.
16. In the **Operator** field, select **=** from the drop-down list.
17. In the **Value** field, enter **Boston**.

18. Click **Append**.
19. In the **Attribute** field, select **sysContact** from the drop-down list.
20. In the **Operator** field, select **in** from the drop-down list.
21. In the **Value** field:
    a. enter **Johnson** and press **<Enter>.**
    b. On the new line, enter **Hickman.**
22. Click **Append**.
23. Click **Save** to save your Additional Filters.
24. Select **Actions** → **Show Members** to view the members of the Node Group that is a result of this filter.

Click here for an example for creating an Interface Group Additional Filters.

Interface Group Additional Filters Expression Example

```
((ifName like ATM* AND ifName != ATMS/0/A) AND (ifSpeed = 10 OR ifSpeed = 100))
```

To add the expression above, follow these steps:

1. Click **AND**.
2. Click **AND**.
3. Select the **AND** you just added to the expression.
4. In the **Attribute** field select **ifName** from the drop-down list.
5. In the **Operator** field, select **like** from the drop-down list.
6. In the **Value** field, enter **ATM***.
7. Click **Append**.
8. In the **Attribute** field, select **ifName** from the drop-down list.
9. In the **Operator** field, select **!=***not equal to* from the drop-down list.
10. In the **Value** field, enter **ATMS/0/A**.
11. Click **Append**.
12. Select the first **AND** that you added to the expression.
13. Click **OR**.
14. Select the **OR** you just added to the expression.
15. In the **Attribute** field, select **ifSpeed** from the drop-down list.
16. In the **Operator** field, select **=** from the drop-down list.
17. In the **Value** field, enter **10**.
18. Click **Append**.
19. In the **Attribute** field, select **ifSpeed** from the drop-down list.
20. In the **Operator** field, select **=** from the drop-down list.

21. In the **Value** field, enter **100**.
22. Click **Append**.
23. Click **Save** to save your Additional Filters.
24. Select **Actions** → **Show Members** to view the members of the Node Group that is a result of this filter.

In a CSV File, Define Node Groups

Node Groups are used for a variety of purposes in NNMi. See "Creating Groups of Nodes or Interfaces" (on page 181) for more information.

You can create any number of Node Groups in addition to the ones that NNMi provides (see "Node Groups Provided by NNMi" (on page 209)).

You can create a Node Group by either using the NNMi console or a comma separated values (CSV) file. For example, if you have Node Group information in a Microsoft Excel spreadsheet, you can save this information as a .csv file and use the `nnmloadnodegroups.ovpl` command to add this node group information to NNMi.

Certain columns in the CSV file define the following aspects of the Node Group:

- Columns 1-3 define the Basic settings.
- *Optional*. Column 4 defines Child Node Groups settings.

  Any Child Node Group members are *always* included in the Node Group, irregardless of any filters. The specified Child Node Groups must already exist in NNMi or be defined in the CSV file in addition to the Parent Node Group.
- *Optional*. Column 5 defines Device Filters settings.

  NNMi first evaluates Device Filters. If any are defined, Nodes must match *at least one* specification to belong to this Node Group.
- *Optional*. Column 6 defines Additional Nodes settings.

  Any Additional Nodes specified are *always* included in the Node Group, irregardless of any filters.
- *Optional*. Columns 7-11 define Additional Filters (only a *subset* of codes are available in the CSV file, see nnmloadnodegroups.ovpl for the current list. You must use the NNMi console to define the Node Group if you need other choices).

  Note: If you do not like the Boolean logic assigned by default when you import your CSV file, use the Additional Filters Editor to change things after importing.

  If a Node passes the Device Filters, NNMi then evaluates any Additional Filters. Nodes *must also pass all* Additional Filters specifications to belong to this Node Group.

To create a Node Group using a comma separated values (CSV) text file, use the `nnmloadnodegroups.ovpl` command:

See the nnmloadnodegroups.ovpl Reference Page for more information about the `nnmloadnodegroups.ovpl` command, including requirements for the CSV file. You must provide a CSV file with a specific syntax and order. Each column in the CSV file has a pre-defined meaning.

Tip: If your goal is to *merge* new information into an existing Node Group, use nnmloadnodegroups.ovpl to create a *new Node Group* with the additional settings. Then use the Node Group form to assign that new Node Group as a *Child Node Group* of the original Node Group.

```
nnmloadnodegroups.ovpl -r [true|false] -u <NNMiadminUsername> -p
<NNMiadminPassword> -f <CSV file name>
```

CSV file name is the name of the CSV file that contains the Node Group information.

`-r true` means *all the settings* for any existing Node Group with the same `Name` are overwritten with the values in your CSV file. *This is not a merge, it is a complete replacement of that Node Group configuration!*

`-r false` (defautl) means if the Node Group `Name` already exists, the `nnmloadnodegroups.ovpl` command does not change the previous settings.

Create Interface Groups

Node Groups are used for a variety of purposes in NNMi. See "Creating Groups of Nodes or Interfaces" (on page 181) for more information.

Interface Group definitions match the way your team identifies important network devices. Each interface group can include one or more interface-type specifications (based on industry-standard IANA ifType-MIB variables).

You can create any number of Interface Groups in addition to the ones that NNMi provides (see "Interface Groups Provided by NNMi" (on page 212)).

When determining membership in this Interface Group, NNMi combines the results of all Interface Group configuration settings in the following manner:

- NNMi first evaluates ifType Filters. If any exist, interfaces must match *at least one* specification to belong to this Interface Group.
- NNMi then evaluates any Additional Filters. Interfaces *must also pass all* Additional Filters specifications to belong to this Interface Group.
- If a Node Group is specified for this Interface Group, any interface in this group must be contained in a node that is a member of the Node Group specified in the Basics section.

Interface Groups are used for a variety of purposes in NNMi:

- Interface Groups are filters for interface views, IP address views, HP Network Node Manager iSPI Performance for Metrics Software, and HP Network Node Manager iSPI Performance for Traffic Software.
- Interface Groups can control how NNMi monitors network devices. For example, instruct NNMi to never generate ICMP or SNMP queries to any interface used for Voice-Over-IP within your network.

To define an Interface Group (if your role allows you to do this):

1. Navigate to the **Interface Group** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select the **Interface Groups** view.
    c. Do one of the following:
        - To create an Interface Group, click the New icon.
        - To edit an Interface Group, click the Open icon in the row representing the Interface Group you want to edit.

2. Provide the Basics for this interface group, such as Name, Notes, and behavior designations (see Interface Group Form help).
3. *Optional.* Navigate to the **Interface Type Filters** tab.

   Identify one or more interface types that belong to this group:

   - To add an Interface Type filter, click the New icon, and continue.
   - To change an Interface Type filter, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an Interface Type filter, select a row and click the Delete icon.
4. In the Interface Type Filter form, click the Lookup icon and select one of the options from the drop-down menu:

   - Quick View to display summary information for the currently selected IfType.
   - Quick Find to view and select from the list of all existing IfTypes (for more information see "Use the Quick Find Window" (on page 28)).
   - Open to display the details of the currently selected IfType.
   - New to create a new IfType (see "Add New IfTypes (Interface Types) to the List" (on page 208)).
5. *Optional.* Navigate to the **Additional Type Filters** tab.

   Use the Additional Filters Editor to filter based on the current values of a subset of Interface object attributes. See "Specify Interface Group Additional Filters" (on page 198).
6. Click **Save and Close** to return to the Interface Group form.
7. Click **Save and Close**.

   If you configured this Interface Group for Monitoring, NNMi applies your changes during the next monitoring cycle. See "Configure Monitoring Behavior" (on page 216).

To review an Interface Group definition:

1. From the workspace navigation panel, select the **Inventory** workspace.
2. Select the **Interface Groups** view.
3. Locate the row representing the Interface Group and click the Open icon.
4. The Interface Group form displays.
5. When finished, click the Close icon.

Special Actions are available for Node Groups and Interface Groups.

Specify Interface Group Additional Filters

The Additional Filters Editor enables you to create expressions to further define the interfaces to be included in an Interface Group. Make sure to design any complex Additional Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Additional Filters editor.

If any Additional Filters are created:

- NNMi first evaluates any Interface Type filter. Nodes must match *at least one* specification to belong to this Interface Group.
- NNMi then evaluates the Additional Filters expression. Nodes *must also match all* Additional Filters expression specifications to belong to this Interface Group.

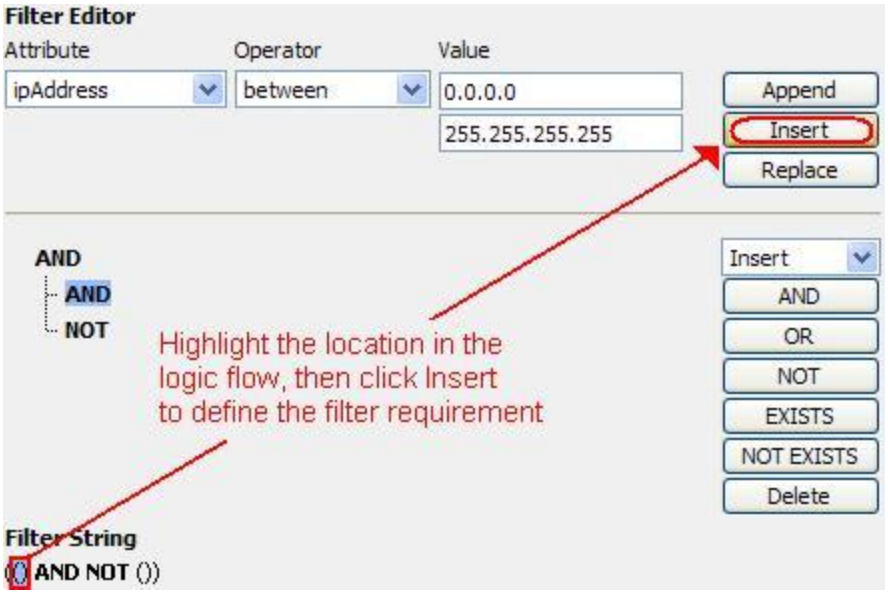
To create any Additional Filters expression:

1. Navigate to the **Interface Group Form: Additional Filters** tab.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Interface Groups**.
    c. Do one of the following:
        - To create an Interface Group definition, click the New icon.
        - To edit an Interface Group definition, click the Open icon in the row representing the configuration you want to edit.
    d. In the Interface Group form, select the **Additional Filters** tab.
2. Establish the appropriate settings for the Additional Filters you need. (See the Additional Filters Editor Components, Additional Filters Editor Buttons table. See also "Guidelines for Creating Additional Filters for Interface Groups" (on page 207).)
    a. Plan out the logic needed for your Filter String.
    b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

        For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

        ```
        (( ) AND NOT ( ))
        ```
    c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the selected filter requirement.

For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



3. Click **Save and Close**.

Additional Filters Editor Components for Interface Groups

Attribute	Description
Attribute	NNMi provides Additional Filters codes for a subset of object attributes. For more information about each one, click the link: • Interface attribute codes [click here for a list of attribute codes] **Values from the Basic Attributes listed on the Interface Form**: ▪ ifName (Name) ▪ hostedOn (Hosted On Node) ▪ ifPhysAddress (Physical Address) **Values from the Interface Form: General Tab**: ▪ ifAlias (InterfaceAlias) ▪ ifDesc (InterfaceDescription) ▪ ifIndex (InterfaceIndex) ▪ ifSpeed (Interface Speed) **Addresses from the Interface Form: IP Addresses Tab**: ▪ ipAddress (IP Address associated with the interface) See "Interface Groups of IPv4 or IPv6 Addresses " (on page 206) for ideas. **Unique Keys from the Interface Form: Capabilities Tab**: ▪ capability (Unique Key of the Capability) **Values from the Interface Form: Custom Attributes Tab**:

Attribute	Description
	Note: When using `customAttrName`and `customAttrValue` pairs, use EXISTS if you want NNMi to consider Nodes that *do not have Custom Attributes* when evaluating the entire Filter String. Otherwise Nodes that do not have Custom Attributes are automatically excluded from the Node Group even if they have values that pass other aspects of your filter. ▪ customAttrName (Custom Attribute Name) ▪ customAttrValue (Custom Attribute Value) • Node attribute codes [click here for a list of attribute codes] **Values from the Basics information on the Node Form:** ▪ isSnmpInterface (Agent Enabled) **Values from the Node Form: General Tab**. ▪ sysOidInterface (System Object ID) • Device Profile attribute codes [click here for a list of attribute codes] **Values from the Basics information on the Device Profile Form**: NNMi matches the Label attribute values from the Device Profile Form for each of the following: ▪ devCategoryInterface (Device Category) ▪ devVendorInterface (Device Vendor) ▪ devFamilyInterface (Device Family) To filter on the parent node's SNMP system object ID number (assigned to a particular make/model), use the sysOidInterface attribute. See Values from the Interface Form: General Tab. • VLAN attribute codes [click here for a list of attribute codes] **Values from the Basic Attributes on the VLAN form:**: **Note**: To maximize performance, when you want to filter interfaces based on a VLAN Id or VLAN Name, avoid using multiple filter expressions. For example, use the `between` operator instead of the greater than or equal to (>=) and less than or equal to (<=) operators. ▪ vlanid (VLAN Id) ▪ vlanName (Name) • Port attribute codes [click here for a list of attribute codes] **Values from the Basic Attributes on the Port form:**: **Note**: If the interface has multiple ports, the interface is selected if there is a match on any one port associated with the interface. ▪ configuredDuplexSetting (Configured Duplex Setting) See Port form for a list of possible values.
Operator	The standard query language (SQL) operations to be used for the search.

Attribute	Description

Note: Only the `is null` Operator returns null values in its search.

Valid operators are described below.

- **=** Finds all values equal to the value specified. Click here for an example.

  Example: `ifName=Fa0/14` finds all interface names that are equal to **Fa0/14**.

- **!=** Finds all values not equal to the value specified. Click here for an example.

  Example:`ifName != lan0` finds all interface names other than **lan0**.

- **<** Finds all values less than the value specified. Click here for an example.

  Example: `ifSpeed <= 100000000` finds all interfaces with an (interface speed) ifSpeed less than **100 Mbps**.

- **<=** Finds all values less than or equal to the value specified. Click here for an example.

  Example: `ifSpeed <= 100000000` finds all interfaces with an (interface speed) ifSpeed less than or equal to **100 Mbps**.

- **>** Finds all values greater than the value specified. Click here for an example.

  Example: `ifSpeed >= 10000000` finds all interfaces with an (interface speed) ifSpeed greater than **10 Mbps**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ifSpeed >= 10000000` finds all interfaces with an (interface speed) ifSpeed greater than or equal to **10 Mbps**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ifSpeed between 10000000 100000000` finds all interfaces with an (interface speed) ifSpeed equal to or greater than **10 Mbps** and equal to or less than **100 Mbps**.

  See "Interface Groups of IPv4 or IPv6 Addresses " (on page 206) for more examples of using the **between** Operator.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ifName in`

  finds all interfaces with names that are **Fa0/14** or **Fa0/15**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**Fa0/14, Fa0/15**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ifName is not null` finds all interfaces that have a name value.

Attribute	Description

- **is null** Finds all blank values. Click here for an example.

  Example:`ifName is null` finds all interfaces that do not have an assigned name value.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The following attributes cannot be used with the `like` operator:

  - ifIndex
  - ifSpeed
  - IPAddress

  The asterisk (*) character means *any number of characters of any type at this location*.

  The question mark (?) character means *any single character of any type at this location*.

  Examples:

  - `ifName like ATM*` finds all interface names that begin with **ATM**.
  - `ifName like Ethernet??*` finds all interface names that *begin with***Ethernet** followed by two characters.
  - `ifName like 10/???BASE-TX*` finds all interface names that have *specific characters at an exact location*, positions 1-3 (10/) and 7-13 (BASE-TX).

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ifSpeed not between 10000000 100000000` finds all interfaces with an (interface speed) ifSpeed less than **10 Mbps** and greater than **100 Mbps**.

  See "Interface Groups of IPv4 or IPv6 Addresses " (on page 206) for more examples of using the **not between** Operator.

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ifName not in`

  finds all interface name values other than **Fa0/14** or **Fa0/15**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**Fa0/14, Fa0/15**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example.

  The following attributes cannot be used with the `not like` operator:

  - ifIndex

Attribute	Description
	■ ifSpeed ■ IPAddress The asterisk (*) character means *any number of characters of any type at this location*. The question mark (?) character means *any single character of any type at this location*. Examples: ■ `ifName not like ATM*` finds all interface names that do not begin with **ATM**. ■ `ifName not like Ethernet??*` finds all interface names that do not *begin with***Ethernet** followed by two characters. ■ `ifName not like 10/???BASE-TX*` finds all interface names that do not have *specific characters at an exact location*, positions 1-3 (10/) and 7-13 (BASE-TX).

Attribute	Description
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `in` and `not in` operators require that each value be entered on a separate line. • When entering a value for the Capability attribute, copy and paste the Unique Key value from the Interface form: Capability tab.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the NOT. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) ifDesc containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) ifName value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use EXISTS or NOT EXISTS, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) ifDesc containing **VLAN**, as well as any Interfaces Custom Attribute

Button	Description
	Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the NOT EXISTS. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use EXISTS or NOT EXISTS, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) ifDesc containing **VLAN**, and excludes any Interfaces that have the Custom Attribute **Role** and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Interface Groups of IPv4 or IPv6 Addresses

Use the Interface Group form's Additional Filters Editor to create Interface Groups based on the following criteria ("Specify Interface Group Additional Filters" (on page 198)):

- All interfaces that have *only* IPv4 addresses
  [click here for details of this filter.]

  Both of the following example interface Group's Additional Filters provide the same results. The first example uses IPv4 address notation. The second example uses IPv6 address notation:

  ```
  ((ipAddress between 0.0.0.0 AND 255.255.255.255) AND NOT (ipAddress not between 0.0.0.0 AND 255.255.255.255))
  ```

  or (*NNMi Advanced with IPv6 enabled*)

  ```
  ((ipAddress between 0.0.0.0 AND 255.255.255.255) AND NOT (ipAddress not between ::ffff:0:0 AND ::ffff:ffff:ffff))
  ```

- All interfaces that have *any* IPv4 addresses (could also have IPv6)
  [click here for details of this filter.]

  The following example interface Group's Additional Filter finds any interface that has at least one IPv4 address:

  ```
  (ipAddress between 0.0.0.0 AND 255.255.255.255)
  ```

- (*NNMi Advanced with IPv6 enabled*) All interfaces that have *only* IPv6 addresses
  [click here for details of this filter.]

  IPv6 addresses extend the number of possible IP addresses. The old IPv4 address range falls within the new IPv6 range. Valid IPv6 address values can be less than or greater than the old IPv4 range of addresses. NNMi Advanced converts the IPv4 addresses to the new IPv6 notation, then stores and filters the IPv4 addresses as IPv6 addresses (`::ffff:a.b.c.d`).

  Both of the following example interface Group's Additional Filters provide the same results. The first example uses IPv4 address notation. The second example uses IPv6 address notation:

  ```
  ((ipAddress not between 0.0.0.0 AND 255.255.255.255) AND NOT (ipAddress between 0.0.0.0 AND 255.255.255.255))
  ```

  or

  ```
  ((ipAddress not between ::ffff:0:0 AND ::ffff:ffff:ffff) AND NOT (ipAddress between 0.0.0.0 AND 255.255.255.255))
  ```

- (*NNMi Advanced with IPv6 enabled*) All interfaces that have *any* IPv6 addresses (could also have IPv4)
  [click here for details of this filter.]

  The following example interface Group's Additional Filter finds any interface that has at least one IPv6 address:

  ```
  ((ipAddress between ::0 AND ::fffe:ffff:ffff) OR (ipAddress ::1:0:0:0 AND ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff))
  ```

- (*NNMi Advanced with IPv6 enabled*) All interfaces that have *both* IPv4 and IPv6 addresses (also known as dual-stack interfaces)
  [click here for details of this filter.]

  The following example interface Group's Additional Filter finds any interface that has at least one IPv4 address and at least one IPv6 address:

  ```
  ((ipAddress between 0.0.0.0 AND 255.255.255.255) AND (ipAddress not between 0.0.0.0 AND 255.255.255.255))
  ```

Note: To maximize the performance of Additional Filters based on an IP Address range, avoid multiple filter expressions. For example, use the `between` operator instead of the greater than or equal to (>=) and less than or equal to (<=) operators that cause NNMi to use multiple queries for finding all addresses that match the filter.

Guidelines for Creating Additional Filters for Interface Groups

The Additional Filters Editor enables you to create expressions to further define the interfaces to be included in an Interface Group. Make sure to design any complex Additional Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Additional Filters Editor.

When creating any Additional Filters for an Interface Group, note the following:

- Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together. View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- When using the AND operator to combine expressions that include Custom Attributes, include only one customAttrName/customAttrValue pair in a sub-expression.
- The AND and OR Boolean Operators must contain at least two expressions as shown in the example below.

```
AND
   ifName like ATMS*
   ifName != ATMS/0/A
    OR
       ifSpeed = 10000000
       ifSpeed = 100000000
```

Note: As shown in the example above, you must use the actual ifSpeed number.

NNMi evaluates the expression above as follows:

```
(ifName like ATMS* AND ifName != ATMS/0/A) AND (ifSpeed = 10000000 OR ifSpeed =
100000000)
```

  - NNMi finds all interfaces with an (interface name) ifName that begins with **ATMS**, but does not include **ATMS/0/A**.
  - Of these interfaces, NNM then finds all interfaces with an (interface speed) ifSpeed of **10 Mbps** or **100 Mbps**.
- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Additional Filters Editor. For example, you append to or replace, the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators. See "Add Boolean Operators in the Additional Filters Editor" (on page 193) for more information.
- You can drag any of the following items to a new location in the Filter String:
  - Filter Editor Options: AND, OR, NOT, EXISTS, NOT EXISTS
  - Filter Expression (Attribute, Operator and Value)
- When moving items in the Filter String, note the following:
  - Click the item you want to move before dragging it to a new location.
  - As you drag a selected item, an underline indicates the target location.
  - If you are moving the selection up, NNMi places the item above the target location.
  - If you are moving the selection down, NNMi places the item below the target location.
  - If you attempt to move the selection to an invalid target location, NNMi displays an error message.

Add New IfTypes (Interface Types) to the List

Interface Type definitions cover all known industry-standard IANA ifType-MIB variables at the time of the release of NNMi. Interface Groups can be built using Interface Type filters. See "Create Interface Groups" (on page 197)

The Interface Types view is provided because:

- Occasionally new Interface Types are added between releases of NNMi. If your team acquires new devices that contain new interface types, you can add the new interface type to the NNMi list of Interface Type definitions.
- When NNMi discovers a new Interface Type, NNMi automatically adds a new entry in the Interface Types view. NNMi detects the assigned IANA ifType-MIB number. NNMi uses that number in both the IfType attribute and the number attribute values. Use this view to provide a more meaningful IfType text string and optional description.

To configure an IANA ifType-MIB definition:

1. Navigate to the **IfTypes** view:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select the **IfTypes** view.
2. Do one of the following:
    - To create an Interface Type definition, click the New icon, and continue.
    - To edit an Interface Type definition, click the Open icon in the row representing the configuration you want to edit, and continue.
    - To delete an Interface Type definition, select a row and click the Delete icon.
3. In the Interface Type form, provide the ifType text string, number, and description.
4. Click **Save and Close**.

Node Groups Provided by NNMi

NNMi Provides the following kinds of Node Groups:

- Node Groups as Predefined View Filters. These Node Groups can also be used for Monitoring Configuration if you find them useful.
- "Island Node Groups" (on page 211). NNMi automatically creates Island Node Groups whenever it detects changes in Layer 2 connections. An Island Node Group is a group of fully-connected nodes that NNMi displays in a group that is not connected to the rest of the topology.

Node Groups As Predefined View Filters

NNMi provides the following Node Groups. You can configure these Node Groups with specific information about your management domain and change them to meet your needs.

Node Groups can be used to filter table views, map views, HP Network Node Manager iSPI Performance for Metrics Software, and HP Network Node Manager iSPI Performance for Traffic Software.

Node Groups Provided by NNMi

Name	Purpose
Important Nodes	**Caution**: Do not delete this Node Group. This Node Group is used by the Causal Engine. Any devices in this group receive special treatment. When a current member of this group stops responding, the Causal Engine generates a "Node Down" incident and sets the device status to Critical. For example, when a WAN Edge Device is in the shadow of another problem (and, therefore, NNMi would normally not generate an incident about that WAN edge router), NNMi generates a

Name	Purpose
	"Node Down" incident because the router is listed in this Important Nodes group. This Node Group is empty by default. Consider populating this group with critical servers that run important applications and critical WAN routers. (*NNM iSPI Performance*) This group automatically becomes a filter for Performance Reports (unless the group has no members). The NNMi administrator can change this default behavior. See "In the Console, Create Node Groups" (on page 182).
Microsoft Windows Systems	This Node Group includes any device manufactured by Microsoft. The Node Group definition is populated with one vendor entry. Any Microsoft devices within your management domain are automatically included in this Node Group.
Networking Infrastructure Devices	This Node Group is populated with a list of categories for network devices. Any devices within your management domain that match these categories are automatically included in this Node Group. Devices in this group are automatically monitored for Node Component fault metrics. (*NNM iSPI Performance*) This group automatically becomes a filter for Performance Reports (unless the group has no members). The NNMi administrator can change this default behavior. See "In the Console, Create Node Groups" (on page 182). (*HP Network Node Manager iSPI Network Engineering Toolset Software*) By default, NNMi automatically uses NNM iSPI NET diagnostic flows to monitor devices in this group.
Non-SNMP Devices	This Node Group includes any device that does not respond to SNMP. The Node Group definition is populated with one entry for a null MIB II sysObjectID value. Any device within your management domain that fails to respond to SNMP queries is automatically included in this Node Group.
Routers	This Node Group is populated with a list of categories for network devices that represent routers. Any router, switch-router, or gateway within your management domain is included in this Node Group. See Node Capabilities Provided by NNMi for more information. This filter is used to create the Routers Node Group map that NNMi provides by default in the Topology Maps workspace. Devices in this group are automatically monitored for Node Component fault metrics (*NNM iSPI Performance*) Devices in this group are automatically monitored for performance, including Node Component performance metrics . This group automatically becomes a filter for Performance Reports. The NNMi administrator can change this default behavior. See "Set Default Monitoring" (on page 219), "Configure Node Monitoring" (on page 234), and "In the Console, Create Node Groups" (on page 182) for more information.

Name	Purpose
Switches	This Node Group is populated with a list of categories for network devices that represent switches. Any switch, ATM switch, or switch-router within your management domain is included in this Node Group. See Node Capabilities Provided by NNMi for more information. This filter is used to create the Switches Node Group map that NNMi provides by default in the Topology Maps workspace.

Node Groups Provided by *NMi Advanced*

Name	Purpose
Virtual Machines	*NNMi Advanced*. Virtual machines being hosted on a VMware ESX server. These servers are identified by a `com.hp.nnm.capability.node.VM` capability.
VMware ESX Hosts	*NNMi Advanced*. A VMware ESX server that is hosting virtual machines. These servers are identified by a `com.hp.nnm.capability.node.hypervisor.vmware.ESX` capability.

Related Topics

"Island Node Groups" (on page 211) (dynamically generated Node Groups)

Island Node Groups

An Island Node Group is a group of fully-connected nodes discovered by NNMi, and NNMi determines this group is not connected to the rest of the topology.

An example of an environment with multiple Island Node Groups is a financial institution or retail store with many branches or stores. Each branch or store might be connected to other branches or stores with a WAN (Wide Area Network) connection. Each branch or store appears as an isolated island of nodes in the NNMi topology.

NNMi automatically updates Island Node Group discovery information whenever it detects changes in Layer 2 connections. NNMi uses the Discovery Interval to determine when the updates actually occur.

Note the following about Island Node Groups:

- NNMi selects a representative node in each Island Node Group as the Source Node associated with an Island Node Group incident. The representative node is selected using the following criteria:
  - Sort all routers in the Node Group alphabetically by name and choose the first one in the list
  - If no routers are in the Node Group, sort all nodes in the Node Group alphabetically by name and choose the first one in the list.
- Island Node Groups are identified using "Island" in the Node Group Name. NNMi also assigns each Island Node Group name a number to ensure the name is unique.
- Island Node Groups are manage internally. Therefore, NNMi administrators should not modify Island Node Group configurations. NNMi overrides any user changes the next time NNMi updates the Island Node Group discovery information.
- Island Node Groups must have at least two nodes.
- How the Status of Island Node Groups is calculated cannot be changed.

The only possible Status values for Island Node Groups are Unknown and Normal. Unknown indicates that NNMi cannot reach any nodes in the group. Normal indicates that NNMi can reach at least one node in the group.

Related Topics

"Node Groups As Predefined View Filters" (on page 209)

Interface Groups Provided by NNMi

NNMi Provides the following Interface Groups as predefined view filters. These Interface Groups can also be used for Monitoring Configuration if you find them useful.

Feel free to populate these Interface Groups with specific information about your management domain and change them to meet your needs.

Interface Groups Provided by NNMi

Name	Purpose
ISDN Interfaces	This Interface Group includes multiple interface types known to be commonly used for ISDN purposes. Any interface within your management domain that meets the defined criteria is automatically included in this Interface Group.
Link Aggregation	*NNMi Advanced*. This Interface Group includes all of the Link Aggregation Aggregator Interfaces discovered in the network. See Layer 2 Neighbor View Map Objects for more information about Aggregator Interfaces. Use **Actions** → **Show Members** to identify the Link Aggregation Aggregator Interfaces in this group.
Point to Point Interfaces	This Interface Group includes multiple interface types known to be commonly used for point-to-point purposes. Any interface within your management domain that meets the defined criteria is automatically included in this Interface Group.
Software Loopback Interfaces	This Interface Group includes any interface that is IfType 24, software loopback from the IANA ifType-MIB. Any interface within your management domain that meets this **loopback address**[1] criteria is automatically included in this Interface Group.
VLAN Interfaces	This Interface Group includes interfaces of ifType l2vlan. The NNMi default Monitoring Configuration settings enable fault monitoring for these interfaces, but disable performance monitoring (because collection of performance data for VLAN interfaces tends to be problematic).
Voice Interfaces	This Interface Group includes multiple interface types known to be commonly used for voice purposes. Any interface within your management domain that meets the defined criteria is automatically included in this Interface Group.

[1]The address associated with the loopback interface. The loopback interface is a virtual interface on a device that provides a route for internal communication. Many vendors provide a specially configured loopback for management purposes. Exact details of how loopbacks are configured varies by vendor and model. See each device's documentation for details. NNMi identifies these loopback addresses by using IfType 24, softwareloopback from the IANA ifType-MIB.

Add Custom Attributes to a Node or Interface Object

If you determine that you want to keep track of additional information about a node or interface, you can add Custom Attributes to these objects. For example, you might determine that you want to track the owner of your nodes on the network. You might also want to track the serial number for each node.

Tip: When defining Node Group definitions or Interface Group definitions, you can use the Additional Filters' `customAttrName/customAttrValue` pairs and the EXISTS expression as a filter for membership in the group.

Custom attributes can be created in two ways:

- Open a particular Node form or Interface form (see below).
- Use the nnmloadattributes.ovpl command line tool to "Add Custom Attributes to Multiple Nodes or Interfaces" (on page 214).

To add Custom Attributes to a node object:

1. Navigate to the **Custom Attributes** tab:
   a. From the workspace navigation panel, select a workspace that contains a Node view. For example, the **Inventory** workspace.
   b. Click the Open icon in the row representing the node with settings you want to edit.
   c. Select the **Custom Attributes** tab.
2. Click the New icon to create a Custom Attribute.
3. Enter a Name and Value. See Node Custom Attributes Form for more information.
4. Click **Save and Close** to return to the main Node Form.
5. Click **Save and Close** to save your changes.

To add Custom Attributes to an interface object:

1. Navigate to the **Custom Attributes** tab:
   a. From the workspace navigation panel, select a workspace that contains an Interfaces view. For example, the **Inventory** workspace.
   b. Click the Open icon in the row representing the interface with settings you want to edit.
   c. Select the **Custom Attributes** tab.
2. Click the New icon to create a Custom Attribute.
3. Enter a Name and Value. See Interface Custom Attributes Form for more information.
4. Click **Save and Close** to return to the main Interface Form.
5. Click **Save and Close** to save your changes.

Add Custom Attributes to Multiple Nodes or Interfaces

Tip: When defining Node Group definitions or Interface Group definitions, you can use the Additional Filters' `customAttrName/customAttrValue` pairs and the EXISTS expression as a filter for membership in the group.

Custom attributes can be created in two ways:

- Open a particular Node form or Interface form, and use the New button to add one (see "Add Custom Attributes to a Node or Interface Object" (on page 213)).
- Use the nnmloadattributes.ovpl command line tool to add Custom Attributes to multiple Nodes (see below),

The nnmloadattributes.ovpl command line tool enables you to load Custom Attributes from a comma-separated values (CVS) file. This feature is useful if you have information about a large number of nodes or interfaces defined in an external data storage, and you would like to load that information into the NNMi database as Custom Attributes. For example:

- Node location information in a Microsoft Excel spreadsheet where you track the location of each node: You can save this information as a .csv file. Use the `nnmloadattributes.ovpl` command to define **BldgLocation** as a Custom Attribute and load the location values for each node into the NNMi database. You can then create a Node Group with an Additional Filters specification using **BldgLocation** as the `customAttrName` and the location of interest, such as **Building Five Upper** as the `customAttrValue`.
- Interface information in a comma-separated value file where you track the name of customers assigned to each interface: Use the `nnmloadattributes.ovpl` command to define **Customer** as a Custom Attribute and load the name values for each customer into the NNMi database. You can then create an Interface Group with an Additional Filters specification using **Customer** as the `customAttrName` and a customer name, such as **Hewlett Packard** as the `customAttrValue`.

To load Custom Attributes for Nodes or Interfaces using a comma-separated file:

See the nnmloadattributes.ovpl Reference Page for more information about the `nnmloadattributes.ovpl` command, including requirements for the CSV file. You must provide a CSV file with a specific syntax and order. Each column in the CSV file has a pre-defined meaning.

```
nnmloadattributes.ovpl -u <NNMiadminUsername> -p <NNMiadminPassword> -r
[true|false] -t node -f <CSV file name>
```

`-r true` = the value of any existing Custom Attribute with the same `customAttrName` is overwritten with the value in your CSV file. The default setting is `-r false` = if the `customAttrName` already exists, the `nnmloadnodegroups.ovpl` command does not change the previous `customAttrValue`.

`-t` is used to specify the object type on which the attributes should be loaded. Use `node` to load Custom Attribute information for nodes.

CSV file name is the name of the CSV file that contains the Node Custom Attribute information.

Monitoring Network Health

Note: If you are using NNMi Advanced, also see "Monitor Router Redundancy Groups (NNMi Advanced)" (on page 246).

Before NNMi can monitor the health of your network, the following tasks must be completed:

- "Configuring Communication Protocol" (on page 66)
- "Discovering Your Network" (on page 113)

For the most flexibility, also complete these tasks:

- Review the "Interface Groups Provided by NNMi" (on page 212) and "Node Groups Provided by NNMi" (on page 209).
- Create your own groups by "Creating Groups of Nodes or Interfaces" (on page 181).

The State Poller and the Causal Engine work together to automatically monitor the health of your network. Many of the tasks you normally do to troubleshoot network problems are now automated. To learn more about how this works, see the following topics:

- "About the State Poller" (on page 215)
- "The NNMi Causal Engine and Monitoring" (on page 216)

You control which network devices NNMi monitors. By monitoring only the devices that are important within your network environment, you keep the amount of traffic generated by NNM to a minimum. You can configure NNMi to check devices with status *other than critical* less frequently (if at all) to prevent unimportant incidents from showing up in the Incident views.

To configure the polling policies that control how NNMi monitors devices in your network, see "Configure Monitoring Behavior" (on page 216). You can configure NNMi monitoring behavior to meet your needs.

About the State Poller

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 Node Component monitoring and Router Redundancy Group monitoring.

State Poller gathers information in the following area and updates the ***State*** field on each object's form:

- Verifies that each monitored IP Address is responding to ICMP ping.
- Verifies that each monitored SNMP Agent is responding to SNMP queries.
- Issues an SNMP query for the following:
  - Each monitored interface, requesting the current value for MIB II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.)
  - Router Redundancy Groups.
  - Node Component data.
- By default, State Poller monitors interfaces connected to another known interface through a Layer 2 connection.

- You can extend monitoring to include the following:
  - Unconnected interfaces
  - Interfaces that have an IP address (for example a router interface that services mobile laptop machines)
  - (*HP Network Node Manager iSPI Performance for Metrics Software*). The State Poller also collects performance data and monitors thresholds. See "Purchase an HP Network Node Manager i Smart Plug-in" (on page 953).

The State Poller stores the results of the queries in the NNMi database and notifies the Causal Engine of any changes. The Causal Engine gathers additional information about the overall health of each interface and SNMP agent. Using this information the Causal Engine calculates the ***Status*** of each node, interface, and SNMP agent (see "The NNMi Causal Engine and Monitoring" (on page 216) for more information).

To configure the behavior of the State Poller, see "Configure Monitoring Behavior" (on page 216).

The NNMi Causal Engine and Monitoring

The Causal Engine actively gathers information about your network devices from incoming incidents and traps. The Causal Engine also uses the data gathered by State Poller and by Discovery to calculate the current health status of each node, interface, IP address, SNMP agent, and connection.

The health status is dynamic (based on what the environment looks like *now*).

The NNMi Causal Engine communicates device health information in the following ways:

- In the database, the Causal Engine stores a multitude of information about each device. You can access this information in the Node, Interface, IP Address, SNMP Agent, and connection forms.
- On the maps, the color of the background shape for each map icon changes to the color that represents the most currently calculated health status, based on the Causal Engine calculations for that node, interface, address, or connection (click here for information about status colors).
- On forms for Nodes, Interfaces, IP addresses, SNMP Agents, and connections, the Causal Engine updates the Status attribute to show the current status: **Normal**, **Warning**, **Minor**, **Major**, **Critical**, **Unknown**, or **No Status**.
- The Status column in table views is updated.

The Causal Engine also uses health status information to determine root cause. See "The NNMi Causal Engine and Incidents" (on page 308) for more information about the Causal Engine, incidents, and root cause analysis.

Configure Monitoring Behavior

Certain devices in your network are the most important ones. You and your team must keep those devices up and running at all times. Adjust NNMi monitoring behavior to focus on the important devices and to check devices with status *other than critical* less frequently (if at all).

Note: NNMi does not poll any private interface, IPv4 **Anycast Rendezvous Point IP Address**[1] or IPv6 Anycast address.

[1]Rendezvous Point addresses are loopback addresses used for routers in multi-cast network configurations.

Based on your individual situation, adjust the NNMi behavior to meet your needs. NNMi applies your Monitoring Configuration settings in the following sequence:

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 Setting**: 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. This means that 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 1 or 2, NNMi applies the default Monitoring Configuration settings.

Tasks for Configuring the Monitoring Behavior

Task	How
"Set Global Monitoring" (on page 217).	*Optional*. Use the Global Control group.
"Set Default Monitoring" (on page 219).	Use the Default Settings tab to establish monitoring behavior for any devices that are discovered, but not included in any Node Settings or Interface Settings definitions.
"Configure Node Monitoring" (on page 234)	*Optional*. Use the Node Settings tab. Configure settings based on Node Groups to customize the way NNMi monitors certain groups of devices in your environment. Prerequisite: "Create Node Groups" (on page 181).
Fine tune behavior for specific types of Interfaces,see "Configure Interface Monitoring" (on page 223) .	*Optional*. Use the Interface Settings tab. Configure settings based on Interface Groups to customize the way NNMi monitors certain interface types in your environment. Prerequisite: "Create Interface Groups" (on page 197).
Detect Interface Changes	*Optional*. Use Device Profiles to configure how to detect interface changes.

Set Global Monitoring

Note: To suspend all SNMP traffic generated by NNMi, rather than only the State Poller Service SNMP traffic, see "Communication Region Form" (on page 80) and "Specific Node Settings Form (Communication Settings)" (on page 96).

To temporarily turn off all NNMi monitoring activity without tampering with your customized monitoring configuration settings:

1. Navigate to the **Monitoring Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Monitoring Configuration**.
2. Locate the **Global Control** group box.
3. Clear the ☐ check box preceding each setting that you want to disable or set the ☑ check box preceding each setting that you want to enable(see table).
4. Click **Save and Close**. NNMi applies your changes. The next regularly scheduled monitoring cycle uses the new settings.

   To verify that State Poller is working as expected, see the report on the State Poller tab in **Help** → **System Information**.

Global Control

Name	Description
Enable State Polling	If ☑ enabled, State Poller monitors all managed objects (for example, interfaces, IP addresses, and SNMP agents) by issuing ICMP pings and SNMP read-only queries for MIB II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the overall health of the device and is supplied by the SNMP Agent.) You can also configure NNMi so that State Poller gathers additional information about Node Components and Router Redundancy Groups. If ☐ disabled: • Previously discovered devices remain with the last calculated state/status. • Newly discovered devices are set to "No Status" with map-symbol background shape color set to beige.
Enable Card Polling	If ☑ enabled, State Poller monitors all managed cards. See Card Form for more information about card metrics. **Note**: Card monitoring is enabled by default. If ☐ disabled: • Previously discovered cards are assigned a State of **Not Polled** and a Status of **No Status** for Card metrics. • Newly discovered cards are assigned a State of **Not Polled** and a Status of **No Status**.
Enable Node Component Polling	If ☑ enabled, State Poller monitors Node Component metrics for all managed nodes. See Node Form: Node Component Tab for more information about Node Component metrics. **Note**: Node Component monitoring is enabled by default. Only the health of Fan and Power Supply Node Components are propagated to the Node level. If ☐ disabled: • Previously discovered devices are assigned a State of **Not Polled** and a Status of **No Status** for Node Component metrics. • Node Component metrics for newly discovered devices are assigned a State of **Not Polled** and a Status of **No Status**.

Name	Description
Enable Router Redundancy Group Polling (*NNMi Advanced*)	If ☑ enabled, NNMi monitors all managed Router Redundancy Groups. See Router Redundancy Group View (*NNMi Advanced*) for more information about Router Redundancy Groups. **Note**: Router Redundancy Group monitoring is enabled by default. If ☐ disabled: • Previously discovered Router Redundancy Groups are assigned a State of **Not Polled** and a Status of **No Status**. • Newly discovered Router Redundancy Groups are assigned a State of **Not Polled** and a Status of **No Status**.

Set Default Monitoring

The choices you make for "defaults" apply only to devices with interfaces, IP addresses, cards, SNMP agents (Management Addresses), Tracked Objects, Router Redundancy Groups, or Node Component monitoring settings that are not covered by any Interface Settings or Node Settings definitions.

To establish default NNMi monitoring behavior:

1. Navigate to the **Defaults Settings** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Monitoring Configuration**.
   c. Locate the **Defaults Settings** tab.
2. Locate the **Default Fault Monitoring** group box.
3. Configure the Default Fault Monitoring behavior (see Default Fault Monitoring table).
4. (*HP Network Node Manager iSPI Performance for Metrics Software*) If the HP Network Node Manager iSPI Performance for Metrics Software is installed, locate the **Default Performance Monitoring** group box.

   Configure the Default Performance Monitoring behavior (see the Default Fault Monitoring table and Default Performance Monitoring table).
5. By default, NNMi monitors only interfaces that are connected to other interfaces. When SNMP polling is enabled, NNMi automatically detects most connections. See "Add or Delete a Layer 2 Connection" (on page 176) for information about manual overrides.

   Optional. If you want to expand default monitoring behavior to include unconnected Interfaces, indicate your choices in the Default Extend the Scope of Polling Beyond Connected Interfaces group box
6. *Optional*. To establish custom monitoring behavior for one or more groups of interfaces, configure Interface Settings, see "Configure Interface Monitoring" (on page 223).
7. *Optional*. To establish custom monitoring behavior for one or more groups of nodes, configure Node Settings, see "Configure Node Monitoring" (on page 234).
8. Click **Save and Close**. NNMi applies your changes. The next regularly scheduled monitoring cycle uses the new settings.

To verify that State Poller is working as expected, see the report on the State Poller tab in **Help** → **System Information**.

Default Fault Monitoring (choose one or none)

Attribute	Description
Enable ICMP Management Address Polling	If ☑ enabled, State Poller only issues ICMP (ping) requests to IP addresses classified as *management addresses* in the NNMi database. **Note**: In the Global Control section of the Monitoring Configuration form, the Enable State Polling attribute must be enabled, too. If ☐ disabled, State Poller does one of the following: • If neither this attribute nor *Enable ICMP Fault Polling* is selected, State Poller does not use ICMP to monitor nodes covered by this configuration setting. • If *Enable ICMP Fault Polling* is selected, State Poller uses ICMP to monitor ALL IP addresses covered by this configuration setting. Changing the default monitoring settings for the management addresses takes effect immediately. To verify the change, see "Verify Monitoring Configuration Settings" (on page 246).
Enable ICMP Fault Polling **Note**: This monitoring option is useful for devices that do not support SNMP. By default, this feature is enabled for the "Non-SNMP Devices" Node Group.	If ☑ enabled, State Poller issues ICMP (ping) requests to verify the availability of discovered IP address. **Note**: In the Global Control section of this form, the Enable State Polling attribute must be enabled, too. If ☐ disabled, State Poller does the following: • If neither this attribute nor *Management IP Address Polling* is selected, State Poller does not use ICMP to monitor nodes covered by this configuration setting. • IP addresses (both previously discovered and newly discovered) have a State attribute value of "Not Polled" and a Status attribute value of "No Status" with the color of the IP address map-symbol set to beige. See Layer 3 Neighbor View. • If both ICMP and SNMP are disabled for a Node, the Node has a Status attribute value of "No Status" and the color of the Node map-symbol background shape is set to beige. **Tip**: To turn off ICMP polling within a subset of your network environment, use the Communication Configuration workspace Region definitions. You can define your own Regions that identify any unreachable addresses in your management domain (for example, the **private IP addresses**[1]).
Enable SNMP Interface Fault Polling	If ☑ enabled, State Poller monitors all interfaces by issuing SNMP read-only queries to devices assigned to this level of the monitoring hierarchy. By default, any connected interface is monitored for MIB II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) If you have *unconnected*

[1] These are IPv4 addresses that can be reused in home and office local area networks (LANs). Following the standards set by RFC 1918 and RFC 4193 (10.*.*.*, 169.254.*.*, 172.16-31.*.*, and 192.168.*.*)

Attribute	Description
	interfaces that you want to monitor, expand NNMi monitoring behavior with the Poll Unconnected Interfaces and the Poll Interfaces Hosting IP Addresses attributes. **Note**: The following attributes must also be enabled: • In the Global Control section of this form, the Enable State Polling attribute must be enabled, too. See Layer 2 Neighbor View. (See "Set Global Monitoring" (on page 217) for more information.) • In the Communication Configuration view, enable State Poller queries with the applicable Enable SNMP Communication attributes (see "Configuring Communication Protocol" (on page 66) for more information). If ☐ disabled, for devices assigned to this level of the monitoring hierarchy: • Causal Engine calculates Status based only on IP address State. • The Interface objects previously discovered change to a State attribute value of "Not Polled" and a Status attribute value of "No Status" (plus any related map-symbol changes to a beige color).
Enable Card Fault Polling	Use this attribute to poll fault metrics for cards. Card fault metrics include Administrative State, Operational State, and Standby State. **Note**: Card Fault Polling is enabled by default. If ☑ enabled, NNMi gathers fault data related to the card fault metrics in devices assigned to this level of the monitoring hierarchy. If ☐ disabled, NNMi does not extend data collection behavior to include card fault data about devices assigned to this level of the monitoring hierarchy. **Note**: NNMi uses the same polling interval set for the Fault Polling Interval.
Enable Node Component Fault Polling **Note**: By default, this feature is enabled for the "Routers" and "Networking Infrastructure Devices" Node Groups.	Use this attribute to poll Node Component fault metrics. Node Component fault metrics include the following: Fan, Power Supply, Temperature, and Voltage. **Note**: Node Component Fault Polling is disabled by default. Only the health of the Power Supply and Fan Node Components are propagated to the Node level. If ☑ enabled, NNMi gathers fault data related to the Node Component fault metrics in devices assigned to this level of the monitoring hierarchy. If ☐ disabled, NNMi does not extend data collection behavior to include Node Component fault data about devices assigned to this level of the monitoring hierarchy. **Note**: NNMi uses the same polling interval set for the Fault Polling Interval.

Attribute	Description
Fault Polling Interval	The time that State Poller waits between issuing queries to gather information for any of the following that are enabled: ICMP Polling, SNMP Polling, Poll Unconnected Interfaces, and Poll Interfaces Hosting IP addresses. **Note**: NNMi monitors SNMP agents (Management Addresses) according to this Fault Polling Interval, *even if ICMP Polling, SNMP Polling, Poll Unconnected Interfaces, and Poll Interfaces Hosting IP addresses are all disabled*. To prevent an SNMP Agent's address from being monitored, one of the following must be true: State Polling is disabled, current Communication Configuration settings turn off SNMP for the SNMP agent's address, or the parent Node is set to Not Managed or Out of Service.

Default Performance Monitoring (*HP Network Node Manager iSPI Performance for Metrics Software*)

Attribute	Description
Enable SNMP Interface Performance Polling	(*HP Network Node Manager iSPI Performance for Metrics Software*) Use this attribute to extend the range of polling data that NNMi collects. HP Network Node Manager iSPI Performance for Metrics Software uses the additional data in a series of performance reports. See "Purchase an HP Network Node Manager i Smart Plug-in" (on page 953) for more information. When enabled, network traffic increases on your network because NNMi gathers performance data about each member of this group on a regular schedule. If ☑ enabled, NNMi gathers performance data from Interfaces, CPU, memory, and buffers in devices assigned to this level of the monitoring hierarchy. If ☐ disabled, NNMi does not extend data collection behavior to include performance data about devices assigned to this level of the monitoring hierarchy. **Note**: The Enable State Polling field must be enabled, too. By default the performance of connected interfaces and addresses is monitored. If you have *unconnected* interfaces that you want to monitor, expand NNMi monitoring behavior by enabling Poll Unconnected Interfaces.
Enable Node Component Performance Polling **Note**: By default, this feature is enabled for the "Routers" Node Group if HP Network Node Manager iSPI Performance for Metrics Software is installed.	(*HP Network Node Manager iSPI Performance for Metrics Software*) Use this attribute to poll Node Component performance. An NNMi administrator can set the threshold for node components related to the following performance metrics: CPU utilization, memory utilization, buffer utilization, buffer miss rate, and buffer failure rate. **Note**: Node Component Performance Polling is disabled by default. If ☑ enabled, NNMi gathers performance data related to the Node Component performance metrics in devices assigned to this level of the monitoring hierarchy. If ☐ disabled, NNMi does not extend data collection behavior to include Node Component performance data about devices assigned to this level of the monitoring hierarchy. **Note**: NNMi uses the same polling interval set for the Performance Polling Interval.

Attribute	Description
Performance Polling Interval	(*HP Network Node Manager iSPI Performance for Metrics Software*) Use this field to set the time period that NNMi waits between issuing network traffic to gather performance data for the HP Network Node Manager iSPI Performance for Metrics Software.

Default Extend the Scope of Polling Beyond Connected Interfaces

Attribute	Description
Poll Unconnected Interfaces	If ☑ enabled, NNMi monitors all interfaces within discovered devices (both connected and unconnected). All interfaces are monitored for MIB II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) **Note**: The Enable State Polling field must be enabled, and SNMP polling of some type must be enabled (for example, Enable SNMP Fault Monitoring and Enable SNMP Performance Polling). If ☐ disabled, State Poller polls according to other configuration settings. **Tip**: Your discovery configuration choices may need to be adjusted to get the results you want. For example, to meet the “connected” criteria for interfaces in switches that do not have an IP address you must add the device to which the interface is connected as a discovery seed. See"Discovery Seeds (as a starting point)" (on page 119).
Poll Interfaces Hosting IP Addresses **Note**: This monitoring option is useful for Router interfaces. By default, this feature is enabled for the "Routers" Node Group.	If ☑ enabled, any unconnected interface that has one or more addresses associated with it is monitored for MIB II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) **Note**: The Enable State Polling field must be enabled, and SNMP polling of some type must be enabled (for example, Enable SNMP Fault Monitoring and Enable SNMP Performance Polling). By monitoring the Interface (in addition to the IP address), NNMi can make more informed decisions about the health of each IP address associated with an unconnected interface. If ☐ disabled, State Poller polls according to other configuration settings. **Tip**: The Communication Configuration workspace provides a method of overriding this setting for specific Regions. You can define your own Region to easily turn off polling to any unreachable addresses in your management domain (for example, the **private IP addresses**[1]).

Configure Interface Monitoring

Before you start, you must establish one or more Interface Group definitions that identify the interface types to which these monitoring settings will apply. NNMi provides nearly 250 interface types to choose from.

[1]These are IPv4 addresses that can be reused in home and office local area networks (LANs). Following the standards set by RFC 1918 and RFC 4193 (10.*.*.*, 169.254.*.*, 172.16-31.*.*, and 192.168.*.*)

Interface monitoring applies to matching interfaces and the IP addresses that are hosted on those interfaces. See also, "Interface Groups Provided by NNMi" (on page 212).

To establish monitoring behavior for one or more predefined Interface Groups:

1. Navigate to the **Interface Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Monitoring Configuration**.
   c. Locate the **Interface Settings** tab.
   d. Do one of the following:
      - To create an Interface Settings definition, click the New icon.
      - To edit an Interface Settings definition, click the Open icon in the row representing the Interface Settings definition you want to edit.
      - To delete an Interface Settings definition, select a row and click the Delete button
2. Establish the appropriate settings to identify this Interface Setting definition (see Basics table).
3. *Optional*. Configure the Fault Monitoring behavior for this Interface Setting definition (see Fault Monitoring table).
4. (*HP Network Node Manager iSPI Performance for Metrics Software*) If the HP Network Node Manager iSPI Performance for Metrics Software software is installed:
   - Configure the Performance Monitoring behavior for this Interface Setting definition (see Performance Monitoring table).
   - *Optional*. Navigate to the Threshold Settings tab to configure the HP Network Node Manager iSPI Performance for Metrics Software software. See "Configure Threshold Monitoring for Interfaces (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 228) for more information.
5. By default, NNMi monitors only interfaces that are connected to other interfaces. When SNMP polling is enabled, NNMi automatically detects most connections. See "Add or Delete a Layer 2 Connection" (on page 176) for information about manual overrides.

   Optional. If you want to expand monitoring behavior for this group to include unconnected Interfaces, indicate your choices in the Extend the Scope of Polling Beyond Connected Interfaces group box.
6. Click **Save and Close** to return to the Monitoring Configuration form.
7. Click **Save and Close**. NNMi applies your changes. The next regularly scheduled monitoring cycle uses the new settings.

   Caution: When you establish monitoring configuration settings, NNMi must recalculate membership in all Node Groups and Interface Groups. This can take some time and slow down your system. Consider making this change during a slow time in your network environment.

   To verify that State Poller is working as expected, see the report on the State Poller tab in **Help** → **System Information**.

Optional. Customize the node monitoring behavior. See "Configure Node Monitoring" (on page 234) . Also see "Detect Interface Changes (renumbering issues)" (on page 174)

Basics

Attribute	Description
Ordering	Enter a unique string (any length), characters 0 through 9. Consider using increments of 100 for the flexibility to insert additional items between existing items over time. NNMi decides which monitoring configurations apply to a node or interface based on the ordering number assigned to the configuration definitions. NNMi monitors the device according to the first match (checked from lowest number to highest number within each category). Categories are read in sequence. Click here for a description of the sequence. 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 Setting**: 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. This means that 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 1 or 2, NNMi applies the default Monitoring Configuration settings. No duplicate Ordering numbers are allowed. Each Interface Setting ordering number must be unique.
Interface Group	Choose one predefined Interface Group from the list. See "Create Interface Groups" (on page 197) for more information. (*NNMi Advanced with IPv6 enabled*) See also "Interface Groups of IPv4 or IPv6 Addresses " (on page 206).

Fault Monitoring

Attribute	Description
Enable ICMP Management Address Poling	If ☑ enabled, State Poller only issues ICMP (ping) requests to IP addresses classified as *management addresses* in the NNMi database. **Note**: In the Global Control section of the Monitoring Configuration form, the Enable State Polling attribute must be enabled, too. If ☐ disabled, State Poller does one of the following: • If neither this attribute nor *Enable ICMP Fault Polling* is selected, State Poller does not use ICMP to monitor nodes covered by this configuration setting. • If *Enable ICMP Fault Polling* is selected, State Poller uses ICMP to monitor ALL IP addresses covered by this configuration setting.

Attribute	Description
Enable ICMP Fault Polling **Note**: This monitoring option is useful for devices that do not support SNMP.	If ☑ enabled, State Poller issues ICMP (ping) requests to verify the availability of discovered IP address. **Note**: In the Global Control section of this form, the Enable State Polling attribute must be enabled, too. If ☐ disabled, State Poller does the following: • If neither this attribute nor *Management IP Address Polling* is selected, State Poller does not use ICMP to monitor nodes covered by this configuration setting. • IP addresses (both previously discovered and newly discovered) have a State attribute value of "Not Polled" and a Status attribute value of "No Status" with the color of the IP address map-symbol set to beige. See Layer 3 Neighbor View. • If both ICMP and SNMP are disabled for a Node, the Node has a Status attribute value of "No Status" and the color of the Node map-symbol background shape is set to beige. **Tip**: To turn off ICMP polling within a subset of your network environment, use the Communication Configuration workspace Region definitions. You can define your own Regions that identify any unreachable addresses in your management domain (for example, the **private IP addresses**[1]).
Enable SNMP Interface Fault Polling	If ☑ enabled, State Poller monitors all interfaces by issuing SNMP read-only queries to devices assigned to this level of the monitoring hierarchy. By default, any connected interface is monitored for MIB II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) If you have *unconnected* interfaces that you want to monitor, expand NNMi monitoring behavior with the Poll Unconnected Interfaces and the Poll Interfaces Hosting IP Addresses attributes. **Note**: The following attributes must also be enabled: • In the Global Control section of this form, the Enable State Polling attribute must be enabled, too. See Layer 2 Neighbor View. (See "Set Global Monitoring" (on page 217) for more information.) • In the Communication Configuration view, enable State Poller queries with the applicable Enable SNMP Communication attributes (see "Configuring Communication Protocol" (on page 66) for more information). If ☐ disabled, for devices assigned to this level of the monitoring hierarchy: • Causal Engine calculates Status based only on IP address State. • The Interface objects previously discovered change to a State attribute value of "Not Polled" and a Status attribute value of "No Status" (plus any related map-symbol changes to a beige color).
Fault Polling Interval	The time that State Poller waits between issuing queries to gather information for any of the following that are enabled: ICMP Polling, SNMP Polling, Poll Unconnected Interfaces, and Poll Interfaces Hosting IP addresses. **Note**: NNMi monitors SNMP agents (Management Addresses) according to this Fault

[1]These are IPv4 addresses that can be reused in home and office local area networks (LANs). Following the standards set by RFC 1918 and RFC 4193 (10.*.*.*, 169.254.*.*, 172.16-31.*.*, and 192.168.*.*)

Attribute	Description
	Polling Interval, *even if ICMP Polling, SNMP Polling, Poll Unconnected Interfaces, and Poll Interfaces Hosting IP addresses are all disabled*. To prevent an SNMP Agent's address from being monitored, one of the following must be true: State Polling is disabled, current Communication Configuration settings turn off SNMP for the SNMP agent's address, or the parent Node is set to Not Managed or Out of Service.

Performance Monitoring (*HP Network Node Manager iSPI Performance for Metrics Software*)

Attribute	Description
Enable SNMP Interface Performance Polling	(*HP Network Node Manager iSPI Performance for Metrics Software*) Use this attribute to extend the range of polling data that NNMi collects. HP Network Node Manager iSPI Performance for Metrics Software uses the additional data in a series of performance reports. See "Purchase an HP Network Node Manager i Smart Plug-in" (on page 953) for more information. When enabled, network traffic increases on your network because NNMi gathers performance data about each member of this group on a regular schedule. If ☑ enabled, NNMi gathers performance data from Interfaces, CPU, memory, and buffers in devices assigned to this level of the monitoring hierarchy. If ☐ disabled, NNMi does not extend data collection behavior to include performance data about devices assigned to this level of the monitoring hierarchy. **Note**: The Enable State Polling field must be enabled, too. By default the performance of connected interfaces and addresses is monitored. If you have *unconnected* interfaces that you want to monitor, expand NNMi monitoring behavior by enabling Poll Unconnected Interfaces.
Performance Polling Interval	(*HP Network Node Manager iSPI Performance for Metrics Software*) Use this field to set the time period that NNMi waits between issuing network traffic to gather performance data for the HP Network Node Manager iSPI Performance for Metrics Software.

Extend the Scope of Polling Beyond Connected Interfaces

Attribute	Description
Poll Unconnected Interfaces	If ☑ enabled, NNMi monitors all interfaces within discovered devices (both connected and unconnected). All interfaces are monitored for MIB II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) **Note**: The Enable State Polling field must be enabled, and SNMP polling of some type must be enabled (for example, Enable SNMP Fault Monitoring and Enable SNMP Performance Polling). If ☐ disabled, State Poller polls according to other configuration settings. **Tip**: Your discovery configuration choices may need to be adjusted to get the results you want. For example, to meet the “connected” criteria for interfaces in switches that do not have an IP address you must add the device to which the interface is connected as a discovery seed. See"Discovery Seeds (as a starting point)" (on page 119).

Attribute	Description
Poll Interfaces Hosting IP Addresses **Note**: This monitoring option is useful for Router interfaces.	If ☑ enabled, any unconnected interface that has one or more addresses associated with it is monitored for MIB II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) **Note**: The Enable State Polling field must be enabled, and SNMP polling of some type must be enabled (for example, Enable SNMP Fault Monitoring and Enable SNMP Performance Polling). By monitoring the Interface (in addition to the IP address), NNMi can make more informed decisions about the health of each IP address associated with an unconnected interface. If ☐ disabled, State Poller polls according to other configuration settings. **Tip**: The Communication Configuration workspace provides a method of overriding this setting for specific Regions. You can define your own Region to easily turn off polling to any unreachable addresses in your management domain (for example, the **private IP addresses**[1]).

Configure Threshold Monitoring for Interfaces (*HP Network Node Manager iSPI Performance for Metrics Software*)

Use the Threshold Settings form to configure NNMi and the HP Network Node Manager iSPI Performance for Metrics Software to monitor thresholds in your network environment. (See "Purchase an HP Network Node Manager i Smart Plug-in" (on page 953) for more information about the HP Network Node Manager iSPI Performance for Metrics Software.) If you set thresholds, NNMi can generate an Incident when any threshold is violated. Examples of the types of threshold you can set for an interface include the following: (See Monitored Attributes in the table below for a complete list.)

- Input and output utilization
- Input and output error rates
- Input and output discard rates

HP Network Node Manager iSPI Performance for Metrics Software provides exceptions reports to track the frequency of threshold breaches. You can open these reports with **Actions → Reporting - Report Menu** in the incident, node, or interface views and forms. (See HP Network Node Manager iSPI Performance for Metrics Software Actions.)

To establish threshold monitoring behavior for the HP Network Node Manager iSPI Performance for Metrics Software:

1. *Prerequisite*. After enabling Performance Monitoring for an Interface Group and before setting thresholds, analyze performance data over time to determine wise threshold settings for each group. See "Determine Reasonable Threshold Settings (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 231).

[1]These are IPv4 addresses that can be reused in home and office local area networks (LANs). Following the standards set by RFC 1918 and RFC 4193 (10.*.*.*, 169.254.*.*, 172.16-31.*.*, and 192.168.*.*)

Note: When performance polling is enabled, network traffic increases on your network while NNMi gathers performance data.

2. Navigate to the **Thresholds Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Monitoring Configuration**.
   c. Navigate to the **Interface Settings** tab.
   d. Do one of the following:
      - To create an Interface Settings definition, click the New icon.
      - To edit an Interface Settings definition, click the Open icon in the row representing the Interface Settings definition you want to edit.
3. Verify that Performance Monitoring is enabled for this Interface Settings definition.
4. In the **Interface Settings** form, navigate to the **Threshold Settings** tab.
5. Do one of the following:
   - To create a threshold definition, click the New icon.
   - To edit a threshold definition, click the Open icon in the row representing the threshold definition you want to edit.
   - To delete a threshold definition, select a row and click the Delete icon.
6. Select the attribute you want to monitor and establish the threshold values for that attribute (see Basic Threshold Settings table). For examples of setting meaningful thresholds, see "Examples of Threshold Monitoring (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 232).
7. Click **Save and Close** to return to the **Interface Settings** form.
8. Click **Save and Close** to return to the **Monitoring Configuration** form.
9. Click **Save and Close**. NNMi applies your changes during the next regularly scheduled monitoring cycle.

   Note: Threshold Incidents are disabled by default within NNMi to prevent Incident storms. If you are ready to generate Threshold Incidents, see "Generate Performance Threshold Incidents (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 426). See also "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312) for a description of the special custom incident attributes available in Threshold Incidents.

   To verify that State Poller is working as expected, see the report on the State Poller tab in **Help** → **System Information**.

Basic Threshold Settings

Attribute	Description
Monitored Attribute	NNMi gathers data to calculate thresholds for the following values. Click any item in this list for more details about that value. **Note**: NNMi also displays the Monitored Attributes that apply to nodes. See "Configure Threshold Monitoring for Nodes (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 240) for more information about these attributes.

Attribute	Description
	• Input Utilization The total number of incoming octets traversing the interface as a percentage of the total possible number of octets (based on the ifSpeed value). From Interface to Interface, the exact MIB variables queried vary based on interface speed and whether the system supports the high speed counters for interfaces. Each interface in an Interface Groups has its utilization calculated by taking the total traffic on all administratively up interfaces in the group and dividing that by the total possible bandwidth. **Tip**: To override the ifSpeed value returned by the device's SNMP agent, see the Interface form. • Output Utilization The total number of outbound octets traversing the interface as a percentage of the total possible number of octets (based on the ifSpeed value). From Interface to Interface, the exact MIB variables queried vary based on interface speed and whether the system supports the high speed counters for interfaces. Each interface in an Interface Group has its utilization calculated by taking the total traffic on all administratively up interfaces in the group and dividing that by the total possible bandwidth. **Tip**: To override the ifSpeed value returned by the device's SNMP agent, see the Interface form. • Input Error Rate Rate based on the reported change in the number of input packets on the interface and the packet error count. What constitutes an error is system specific, but likely includes such issues as bad packet checksums, incorrect header information, and runt packets. • Output Error Rate Rate based on the reported change in the number of output packets on the interface and the packet error count. What constitutes an error is system specific, but likely includes such issues as collisions and buffer errors. • Input Discard Rate Rate based on the reported change in the number of input packets on the interface and the discarded packet count. Packets may be discarded because of a variety of issues, including receive buffer overflows, congestion, or system specific issues. • Output Discard Rate Rate based on the reported change in the number of output packets on the interface and the discarded packet count. Packets may be discarded because of a variety of issues, including transmission buffer overflows, congestion, or system specific issues.
High Value	Designate the percentage above which would indicate a value in the High range. Valid entries are 0.00 through 100.00 **Note**: If you use 100.00 the threshold is disabled because it cannot be crossed.

Attribute	Description
High Value Rearm	Designate the percentage that when reached would indicate the end of a high threshold situation. Valid entries are 0.00 through 100.00 **Note**: The High Value Rearm must be less than or equal to the High Value and greater than the Low Value Rearm.
High Trigger Count	Designate the number of consecutive polling cycles in which the value must remain in the High range before the threshold state changes to High. **Note**: The interface performance values are the average value over the entire polling interval, so a trigger count of 1 is usually appropriate.
Low Value	The Low Value must be less than or equal to the High Value. Designate the percentage below which would indicate a value in the low range. Valid entries are 0.00 through 100.00 **Note**: If you use 0.00 the threshold is disabled because it cannot be crossed.
Low Value Rearm	Designate the percentage that when reached would indicate the end of a low threshold situation. Valid entries are 0.00 through 100.00. **Note**: The Low Value Rearm must be greater than or equal to the Low Value and less than the High Rearm Value.
Low Trigger Count	Designate the number of consecutive polling cycles in which the value must remain in the Low range before the threshold state changes to Low. **Note**: The interface performance values are the average value over the entire polling interval, so a trigger count of 1 is usually appropriate.

Determine Reasonable Threshold Settings (*HP Network Node Manager iSPI Performance for Metrics Software*)

You must decide how to define normal behavior for devices in the associated Node Group or Interface Group. You can then set reasonable thresholds for the group, and avoid Threshold Incident storms. See "Examples of Threshold Monitoring (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 232).

Create a Node Group or Interface Group filter that includes the devices you want to monitor. Export the Node Group or Interface Group filter to HP Network Node Manager iSPI Performance for Metrics Software. See "Creating Groups of Nodes or Interfaces" (on page 181).

Enable Performance Monitoring for the Node Group or Interface Group. See "Configure Node Monitoring" (on page 234) or "Configure Interface Monitoring" (on page 223). Then wait a minimum of 24 hours before following the steps below.

Access the HP Network Node Manager iSPI Performance for Metrics Software Headline report:

1. In the NNMi console, click **Actions** → **Reporting - Report Menu**.
2. Click the link for **Headline**. The Headline report displays data from the past 24 hours from the time you request the report. So if you run the report at 5.03 p.m., the report includes data since 5.03 p.m. yesterday. Click the **Help** link in the report if you need information about how to use this report.
3. Open the **Topology Filters** panel and restrict your view to the network elements for which you are determining thresholds.

4. Click **Confirm Selection** to return to the report.
5. Open the **Time Controls** panel and select a start time and interval.
6. Click **Confirm Selection**.
7. The report appears using the filters you specified.
8. Study the Range & Exceptions graphs to guide your decision about what constitutes reasonable threshold settings. See online help for this report for information about how to read this report.

Examples of Threshold Monitoring (*HP Network Node Manager iSPI Performance for Metrics Software*)

You can configure interface threshold monitoring if the HP Network Node Manager iSPI Performance for Metrics Software is installed. See "Purchase an HP Network Node Manager i Smart Plug-in" (on page 953) for more information.

Several examples are presented These examples are not intended to be recommendations. Consider all aspects of your network environment and set performance thresholds that are meaningful in your environment:

- Thresholds to Monitor Utilization on WAN Connections
- Thresholds to Monitor Utilization on Important Interfaces
- Thresholds to Monitor Important Interfaces for Discards
- Thresholds to Monitor Important Interfaces for Errors

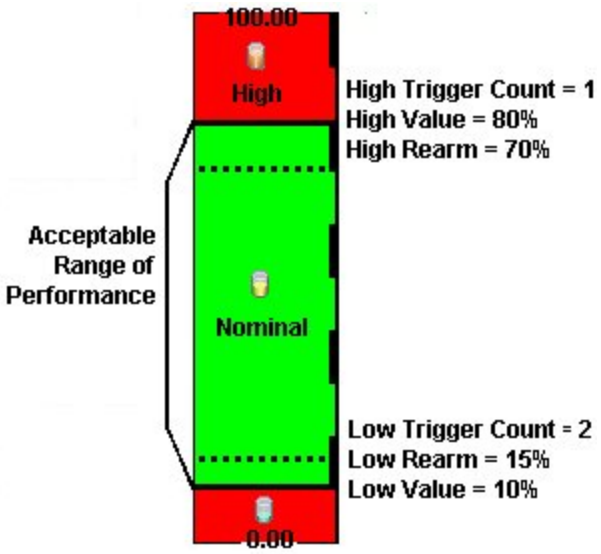
Example 1: Monitor Utilization on WAN Connections

You want to monitor the connections between two sites to verify that your service provider is meeting their guaranteed throughput volume. You pay a fixed cost for a specific bandwidth over this WAN interface.

- Monitor for under-utilization which wastes money (less than 10%).

  Tip: If you don't care about under-utilization, set Low Value and Rearm to 0% as shown in Example 2.

- Monitor for over-utilization, which may result in performance bottlenecks or service provider surcharges (greater than 80%).



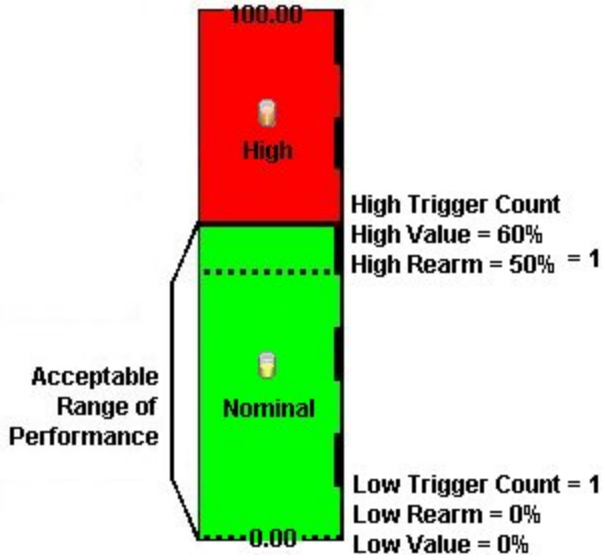
Note: Sometimes an Interface's MIB II ifspeed value is not reported accurately. This may result in threshold calculations outside the 0.00 - 100.00 range. If this happens, the Interface threshold State set to "Unavailable." To correct the problem:

1. Access the **Inventory** workspace
2. Open **Interface** view.
3. Open the form for the Interface that is reporting a threshold state of "Unavailable."
4. Navigate to the **General** tab.
5. Enter a valid entry in **Input Speed** or **Output Speed** (this overrides the value returned by the device's SNMP agent so that NNMi can accurately calculate utilization thresholds).

Example 2: Monitor Utilization on Important Interfaces

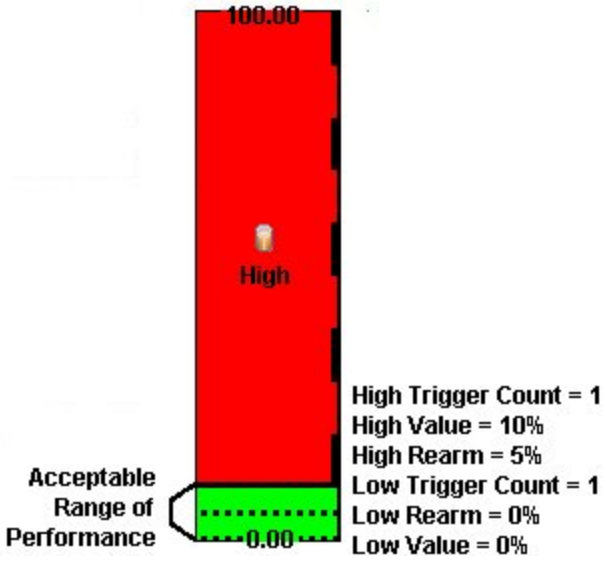
You want to monitor an important Ethernet interface and be notified if it is getting overloaded.

An Ethernet interface configured for full-duplex operation has an acceptable operating range of 0-60%. When average utilization is greater than 60%, NNM generates a High Threshold incident.



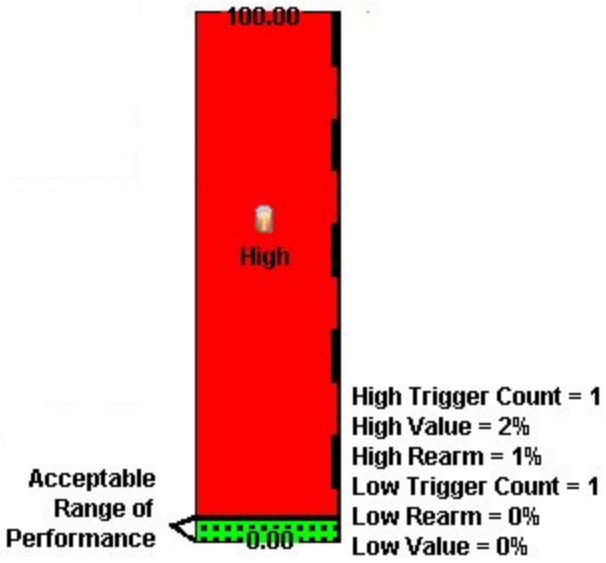
Example 3: Monitor Important Interfaces for Discards

You want to know anytime an interface is dropping data. The acceptable limit for interface discards is 10%. The threshold state is High when the discard rate exceeds 10% and returns to Nominal when the discard rate drops below 5%.



Example 4: Monitor Important Interfaces for Errors

You want to know if packet errors occur. The acceptable limit for packet errors is 2%.The threshold state is High Level (HL) when the error rate exceeds 2% and returns to normal when the error rate drops below 1%.



Configure Node Monitoring

Before you start, you must establish one or more Node Group definitions that identify the nodes to which these monitoring settings will apply. See also, "Node Groups Provided by NNMi" (on page 209).

To establish monitoring behavior for a predefined Node Group:

1. Navigate to the **Node Settings** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.

b. Select **Monitoring Configuration**.

c. Locate the **Node Settings** tab.

d. Do one of the following:

- To create a Node Settings definition, click the New icon.
- To edit a Node Settings definition, click the Open icon in the row representing the Node Settings definition you want to edit.

2. Establish the appropriate settings to identify this Node Setting definition (see Basics table).
3. *Optional*. Configure the Fault Monitoring behavior for this Node Setting definition (see Fault Monitoring table).
4. (*HP Network Node Manager iSPI Performance for Metrics Software*) If the HP Network Node Manager iSPI Performance for Metrics Software is installed:
   - Configure the Performance Monitoring behavior for this Node Setting definition (see Performance Monitoring table).
   - *Optional*. Navigate to the Threshold Settings tab to configure the HP Network Node Manager iSPI Performance for Metrics Software. See "Configure Threshold Monitoring for Nodes (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 240) for more information.
5. By default, NNMi monitors only interfaces that are connected to other interfaces. When SNMP polling is enabled, NNMi automatically detects most connections. See "Add or Delete a Layer 2 Connection" (on page 176) for information about manual overrides.

   Optional. If you want to expand monitoring behavior for this group to include unconnected Interfaces, indicate your choices in the Extend the Scope of Polling Beyond Connected Interfaces group box.
6. Click **Save and Close**. NNMi applies your changes. The next regularly scheduled monitoring cycle uses the new settings.

   Caution: When you establish monitoring configuration settings, NNMi must recalculate membership in all Node Groups and Interface Groups. This can take some time and slow down your system. Consider making this change during a slow time in your network environment.

   To verify that State Poller is working as expected, see the report on the State Poller tab in **Help** → **System Information**.

Optional. Customize the interface monitoring behavior. See "Configure Interface Monitoring" (on page 223)
.

Basics

Attribute	Description
Ordering	Enter a unique string (any length), characters 0 through 9. Consider using increments of 100 for the flexibility to insert additional items between existing items over time. NNMi decides which monitoring configurations apply to a node or interface based on the ordering number assigned to the configuration definitions. NNMi monitors the device according to the first match (checked from lowest number to highest number within each category). Categories are read in sequence. Click here for a description of the sequence. 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.

Attribute	Description
	2. **Node Setting**: 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. This means that 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 1 or 2, NNMi applies the default Monitoring Configuration settings. No duplicate Ordering numbers are allowed. Each Node Setting ordering number must be unique.
Node Group	Choose one predefined Node Group from the list. See "Create Node Groups" (on page 181) for more information. (*NNMi Advanced with IPv6 enabled*) See also "Node Groups of IPv4 or IPv6 Addresses " (on page 190).

Fault Monitoring (choose one or none)

Attribute	Description
Enable ICMP Management Address Polling	If ☑ enabled, State Poller only issues ICMP (ping) requests to IP addresses classified as *management addresses* in the NNMi database. **Note**: In the Global Control section of the Monitoring Configuration form, the Enable State Polling attribute must be enabled, too. If ☐ disabled, State Poller does one of the following: • If neither this attribute nor *Enable ICMP Fault Polling* is selected, State Poller does not use ICMP to monitor nodes covered by this configuration setting. • If *Enable ICMP Fault Polling* is selected, State Poller uses ICMP to monitor ALL IP addresses covered by this configuration setting.
Enable ICMP Fault Polling **Note**: This monitoring option is useful for devices that do not support SNMP. By default, this feature is enabled for the "Non-SNMP Devices" Node Group.	If ☑ enabled, State Poller issues ICMP (ping) requests to verify the availability of discovered IP address. **Note**: In the Global Control section of this form, the Enable State Polling attribute must be enabled, too. If ☐ disabled, State Poller does the following: • If neither this attribute nor *Management IP Address Polling* is selected, State Poller does not use ICMP to monitor nodes covered by this configuration setting. • IP addresses (both previously discovered and newly discovered) have a State attribute value of "Not Polled" and a Status attribute value of "No

Attribute	Description
	Status" with the color of the IP address map-symbol set to beige. See Layer 3 Neighbor View. • If both ICMP and SNMP are disabled for a Node, the Node has a Status attribute value of "No Status" and the color of the Node map-symbol background shape is set to beige. **Tip**: To turn off ICMP polling within a subset of your network environment, use the Communication Configuration workspace Region definitions. You can define your own Regions that identify any unreachable addresses in your management domain (for example, the **private IP addresses**[1]).
Enable SNMP Interface Fault Polling	If ☑ enabled, State Poller monitors all interfaces by issuing SNMP read-only queries to devices assigned to this level of the monitoring hierarchy. By default, any connected interface is monitored for MIB II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) If you have *unconnected* interfaces that you want to monitor, expand NNMi monitoring behavior with the Poll Unconnected Interfaces and the Poll Interfaces Hosting IP Addresses attributes. **Note**: The following attributes must also be enabled: • In the Global Control section of this form, the Enable State Polling attribute must be enabled, too. See Layer 2 Neighbor View. (See "Set Global Monitoring" (on page 217) for more information.) • In the Communication Configuration view, enable State Poller queries with the applicable Enable SNMP Communication attributes (see "Configuring Communication Protocol" (on page 66) for more information). If ☐ disabled, for devices assigned to this level of the monitoring hierarchy: • Causal Engine calculates Status based only on IP address State. • The Interface objects previously discovered change to a State attribute value of "Not Polled" and a Status attribute value of "No Status" (plus any related map-symbol changes to a beige color).

[1]These are IPv4 addresses that can be reused in home and office local area networks (LANs). Following the standards set by RFC 1918 and RFC 4193 (10.*.*.*, 169.254.*.*, 172.16-31.*.*, and 192.168.*.*)

Attribute	Description
Enable Card Fault Polling	Use this attribute to poll fault metrics for cards. Card fault metrics include Administrative State, Operational State, and Standby State. **Note**: Card Fault Polling is enabled by default. If ☑ enabled, NNMi gathers fault data related to the card fault metrics in devices assigned to this level of the monitoring hierarchy. If ☐ disabled, NNMi does not extend data collection behavior to include card fault data about devices assigned to this level of the monitoring hierarchy. **Note**: NNMi uses the same polling interval set for the Fault Polling Interval.
Enable Node Component Fault Polling **Note**: By default, this feature is enabled for the "Routers" and "Networking Infrastructure Devices" Node Groups.	Use this attribute to poll Node Component fault metrics. Node Component fault metrics include the following: Fan, Power Supply, Temperature, and Voltage. **Note**: Node Component Fault Polling is disabled by default. Only the health of the Power Supply and Fan Node Components are propagated to the Node level. If ☑ enabled, NNMi gathers fault data related to the Node Component fault metrics in devices assigned to this level of the monitoring hierarchy. If ☐ disabled, NNMi does not extend data collection behavior to include Node Component fault data about devices assigned to this level of the monitoring hierarchy. **Note**: NNMi uses the same polling interval set for the Fault Polling Interval.
Fault Polling Interval	The time that State Poller waits between issuing queries to gather information for any of the following that are enabled: ICMP Polling, SNMP Polling, Poll Unconnected Interfaces, and Poll Interfaces Hosting IP addresses. **Note**: NNMi monitors SNMP agents (Management Addresses) according to this Fault Polling Interval, *even if ICMP Polling, SNMP Polling, Poll Unconnected Interfaces, and Poll Interfaces Hosting IP addresses are all disabled*. To prevent an SNMP Agent's address from being monitored, one of the following must be true: State Polling is disabled, current Communication Configuration settings turn off SNMP for the SNMP agent's address, or the parent Node is set to Not Managed or Out of Service.

Performance Monitoring (*HP Network Node Manager iSPI Performance for Metrics Software*)

Attribute	Description
Enable SNMP Interface Performance Polling	(*HP Network Node Manager iSPI Performance for Metrics Software*) Use this attribute to extend the range of polling data that NNMi collects. HP Network Node Manager iSPI Performance for Metrics Software uses the additional data in a series of performance reports. See "Purchase an HP Network Node Manager i Smart Plug-in" (on page 953) for more information. When enabled, network traffic increases on your network because NNMi gathers performance data about each member of this group on a regular schedule. If ☑ enabled, NNMi gathers performance data from Interfaces, CPU, memory, and buffers in devices assigned to this level of the monitoring hierarchy. If ☐ disabled, NNMi does not extend data collection behavior to include

Attribute	Description
	performance data about devices assigned to this level of the monitoring hierarchy. **Note**: The Enable State Polling field must be enabled, too. By default the performance of connected interfaces and addresses is monitored. If you have *unconnected* interfaces that you want to monitor, expand NNMi monitoring behavior by enabling Poll Unconnected Interfaces.
Enable Node Component Performance Polling **Note**: By default, this feature is enabled for the "Routers" Node Group if HP Network Node Manager iSPI Performance for Metrics Software is installed.	(*HP Network Node Manager iSPI Performance for Metrics Software*) Use this attribute to poll Node Component performance. An NNMi administrator can set the threshold for node components related to the following performance metrics: CPU utilization, memory utilization, buffer utilization, buffer miss rate, and buffer failure rate. **Note**: Node Component Performance Polling is disabled by default. If ☑ enabled, NNMi gathers performance data related to the Node Component performance metrics in devices assigned to this level of the monitoring hierarchy. If ☐ disabled, NNMi does not extend data collection behavior to include Node Component performance data about devices assigned to this level of the monitoring hierarchy. **Note**: NNMi uses the same polling interval set for the Performance Polling Interval.
Performance Polling Interval	(*HP Network Node Manager iSPI Performance for Metrics Software*) Use this field to set the time period that NNMi waits between issuing network traffic to gather performance data for the HP Network Node Manager iSPI Performance for Metrics Software.

Extend the Scope of Polling Beyond Connected Interfaces

Attribute	Description
Poll Unconnected Interfaces	If ☑ enabled, NNMi monitors all interfaces within discovered devices (both connected and unconnected). All interfaces are monitored for MIB II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) **Note**: The Enable State Polling field must be enabled, and SNMP polling of some type must be enabled (for example, Enable SNMP Fault Monitoring and Enable SNMP Performance Polling). If ☐ disabled, State Poller polls according to other configuration settings. **Tip**: Your discovery configuration choices may need to be adjusted to get the results you want. For example, to meet the "connected" criteria for interfaces in switches that do not have an IP address you must add the device to which the interface is connected as a discovery seed. See"Discovery Seeds (as a starting point)" (on page 119).
Poll Interfaces Hosting IP Addresses **Note**: This monitoring option is useful for Router interfaces. By default, this feature is enabled for the "Routers" Node Group.	If ☑ enabled, any unconnected interface that has one or more addresses associated with it is monitored for MIB II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) **Note**: The Enable State Polling field must be enabled, and SNMP polling of some type must be enabled (for example, Enable SNMP Fault Monitoring and Enable SNMP Performance Polling). By monitoring the Interface (in addition to the IP address), NNMi can make more informed decisions about the health of each IP address associated with an unconnected interface. If ☐ disabled, State Poller polls according to other configuration settings. **Tip**: The Communication Configuration workspace provides a method of overriding this setting for specific Regions. You can define your own Region to easily turn off polling to any unreachable addresses in your management domain (for example, the **private IP addresses**[1]).

Configure Threshold Monitoring for Nodes (*HP Network Node Manager iSPI Performance for Metrics Software*)

The Threshold Settings form is used only to configure threshold monitoring when HP Network Node Manager iSPI Performance for Metrics Software is installed. See "Purchase an HP Network Node Manager i Smart Plug-in" (on page 953) for more information. If you set thresholds, NNMi generates an Incident when any threshold is violated. Examples of the types of threshold you can set for a node include the following: (See Monitored Attributes in the table below for a complete list.)

- CPU 5 second utilization
- CPU 1 minute utilization
- CPU 5 minute utilization
- Memory utilization

[1]These are IPv4 addresses that can be reused in home and office local area networks (LANs). Following the standards set by RFC 1918 and RFC 4193 (10.*.*.*, 169.254.*.*, 172.16-31.*.*, and 192.168.*.*)

- Buffer utilization
- Buffer miss rate
- Buffer failure rate

The HP Network Node Manager iSPI Performance for Metrics Software provides exceptions reports to track frequency. HP Network Node Manager iSPI Performance for Metrics Softwaremonitors your network traffic. Network traffic increases while NNMi gathers performance data.

To establish threshold monitoring behavior for the HP Network Node Manager iSPI Performance for Metrics Software:

1. After enabling Performance Monitoring for a Node Group and before setting thresholds, analyze performance data over time to determine wise threshold settings for each group. See "Determine Reasonable Threshold Settings (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 231).
2. Navigate to the **Thresholds Settings** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Monitoring Configuration**.
    c. Navigate to the **Node Settings** tab.
    d. Do one of the following:
        - To create a Node Settings definition, click the New icon.
        - To edit a Node Settings definition, click the Open icon in the row representing the Node Settings definition you want to edit.
3. *Prerequisite*. Verify that Performance Monitoring is enabled for this Node Settings definition.
4. In the **Node Settings** form, navigate to the **Threshold Settings** tab.
5. Do one of the following:
    - To create a threshold definition, click the New icon.
    - To edit a threshold definition, click the Open icon in the row representing the threshold definition you want to edit.
    - To delete a threshold definition, select a row and click the Delete icon.
6. Select the attribute you want to monitor and establish the threshold values for that attribute (see Basic Threshold Settings table). For examples of setting meaningful thresholds, see "Examples of Threshold Monitoring (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 232).
7. Click **Save and Close** to return to the Node Settings form.
8. Click **Save and Close** to return to the Monitoring Configuration form.
9. Click **Save and Close**. NNMi applies your changes during the next regularly scheduled monitoring cycle.

    Note: Threshold Incidents are disabled by default within NNMi to prevent Incident storms. If you are ready to generate Threshold Incidents, see "Generate Performance Threshold Incidents (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 426). And to learn

about your incident configuration choices, see "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312) for a description of the special custom incident attributes available in Threshold Incidents.

To verify that State Poller is working as expected, see the report on the State Poller tab in **Help** → **System Information**.

Basic Threshold Settings

Attribute	Description
Monitored Attribute	NNMi gathers data to calculate thresholds for the following values. Click any item in this list for more details about that value. **Note**: NNMi also displays the Monitored Attributes that apply to interfaces. See "Configure Threshold Monitoring for Interfaces (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 228) for more information about these attributes • CPU 5Sec Utilization Percentage of CPU usage in relation to the total amount of CPU available. This percentage is measure at 5-second intervals. • CPU 1Min Utilization Percentage of CPU usage in relation to the total amount of CPU available. This percentage is measured at 1-minute intervals. • CPU 5Min Utilization Percentage of CPU usage in relation to the total amount of CPU available. This percentage is measured at 5-minute intervals. • Memory Utilization Percentage of memory usage in relation to the total amount of memory available. • Buffer Utilization Percentage of buffer usage in relation to the total amount of buffer space available. • Buffer Miss Rate Counter indicating that the number of available buffers in the pool has dropped below the minimum level. • Buffer Failure Rate Percentage value based on the number of buffer failures caused by insufficient memory when trying to create additional buffers.
High Value	Designate the percentage above which would indicate a value in the High range. Valid entries are 0.00 through 100.00 **Note**: If you use 100.00 the threshold is disabled because it cannot be crossed.
High Value Rearm	Designate the percentage that when reached would indicate the end of a high threshold situation. Valid entries are 0.00 through 100.00 **Note**: The High Value Rearm must be less than or equal to the High Value and greater than the Low Value Rearm.
High Trigger Count	Designate the number of consecutive polling cycles in which the value must remain in the High range before the threshold state changes to High.
Low Value	The Low Value must be less than or equal to the High Value. Designate the percentage below which would indicate a value in the low range. Valid

Attribute	Description
	entries are 0.00 through 100.00 **Note**: If you use 0.00 the threshold is disabled because it cannot be crossed.
Low Value Rearm	Designate the percentage that when reached would indicate the end of a low threshold situation. Valid entries are 0.00 through 100.00. **Note**: The Low Value Rearm must be greater than or equal to the Low Value and less than the High Rearm Value.
Low Trigger Count	Designate the number of consecutive polling cycles in which the value must remain in the Low range before the threshold state changes to Low.

Configure Node Group Status

NNMi enables an NNMi administrator to configure the Node Group status calculations using either of the following methods:

- Assign the Node Group the most severe status of any Node Group member. This is the default method for obtaining Node Group Status.
- Configure the percentage thresholds for one or more Node Group target statuses. For example, when defining percentage values for a target status of **Critical**, you might change the default so that 30 percent of the nodes in the group must have a status other than Normal, for the Node Group Status to be **Critical**.

Tip: Use the **Actions** → **Status Details** to see how NNMi calculates the status for a selected Node Group.

To configure Node Group status calculations, do the following:

1. Navigate to the **Status Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Status Configuration.**
2. Make one of the following configuration choices:
   - To assign the Node Group the most severe Status of any Node Group member, in the **Status Configuration** form, under **Global Control,** make sure **Propagate Most Severe Status** is checked:

     Propagate Most Severe Status ☑
   - To configure percentage values for a Node Group Target Status, do the following:
     - In the **Status Configuration** form, under **Global Control,** make sure the **Propagate Most Severe Status** is cleared:

       Propagate Most Severe Status ☐
     - Configure the percentage values for a Node Group Target Status
3. Click **Save and Close**.

   NNMi applies your changes after the configuration is saved. Node Group status is updated anytime a Node Group membership changes.

Configure Percentage Values for the Target Status

NNMi enables you to configure how the status of a Node Group is calculated.

Note: The percentage is calculated using only those nodes in the Node Group that have a Management Mode value of **Managed**. For example, if a Node Group includes 10 nodes and 3 of the nodes are **Not Managed**, 5 of the nodes have a Status of **Normal**, and 2 have a status of **Critical,** the percentage of **Critical** nodes is 2/7 * 100.

To configure the percentage values for a Node Group Target Status:

1. Navigate to the **Status Configuration** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Status Configuration.**
2. Locate the **Node Group Status Settings** tab.
3. Do one of the following:
    - To create a Node Group Status Settings definition, click the New icon.
    - To edit a Node Group Status Settings definition, click the Open icon in the row representing the Node Group Status Settings definition you want to edit.
    - To delete a Node Group Settings definition, select a row and click the Delete button
4. Establish the appropriate settings to identify this Node Group Status Settings definition. (See the **"Node Group Status Settings Form" (on page 244)** form)

Note: You can only define one configuration for each Target Status.

Node Group Status Settings Form

The Node Group Status Settings form is used to configure the percentage thresholds for a Node Group Target Status. The percentage thresholds you specify define what percentage of nodes within the group must have a particular Status. When the percentage thresholds are reached, the Node Group is assigned the associated Target Status. For example, when defining percentage thresholds for a target status of **Critical**, you might change the default so that 10 percent of the nodes in the group must have a status of **Critical** for the Node Group Status to be **Critical**.

Note: Use a percentage threshold between 0 (zero) and 1 (for example, .01) to indicate the Target Status to be reached when one node in the Node Group reaches a specified Status. For example, if you want the Node Group Status to be set to **Critical** when the Status of one node in the Group becomes **Critical**, enter a percentage less than one for the **Critical %** value.

To define percentage thresholds for a Target Status:

1. Navigate to the **Node Group Status Settings** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Status Configuration**.
    c. Navigate to the **Node Group Status Settings** tab.

d. Do one of the following:

- To create a Node Group Status Settings definition, click the New icon.
- To edit a Node Group Settings definition, click the Open icon in the row representing the Node Group Settings definition you want to edit.
- To delete a Node Group Settings definition, select a row and click the Delete icon.

2. Set the Target Status and percentages you want (see Basic Attributes table).
3. Click **Save and Close** to return to the **Status Configuration** form.
4. Click **Save and Close**. NNMi applies your changes after the configuration is saved.

Basics Attributes

Attribute	Description
Target Status	The Status you are configuring. This Status is assigned to the Node Group whenever the specified percentage thresholds are reached. Note the following: • Whether all or one of the percentage thresholds must be reached for a Target Status configuration depends on the Boolean operator you select. The default Boolean operator is OR. (Also see Combine with AND below.) • If you do not specify any percentages for a Target Status, it does not appear as a Status for a Node Group.
Critical %	Specifies the required percentage of nodes in the group that must have a Status value set to **Critical** before NNMi assigns the Target Status.
Major %	Specifies the required percentage of nodes in the group that must have a Status value set to **Major** before NNMi assigns the Target Status.
Minor %	Specifies the required percentage of nodes in the group that must have a Status value set to **Minor** before NNMi assigns the Target Status.
Warning %	Specifies the required percentage of nodes in the group that must have a Status value set to **Warning** before NNMi assigns the Target Status.
Non-Normal %	Specifies the required percentage of nodes in the group that must have a Status value set to any of the following before NNMi assigns the Target Status: • Critical • Major • Minor • Warning
Unknown %	Specifies the required percentage of nodes in the group that must have a Status value set to **Unknown** before NNMi assigns the Target Status.
Combine with AND	Specifies that you want NNMi to combine the percentage thresholds you enter using the AND Boolean operator. When using this option, note the following: • All percentage thresholds you enter must be reached for the Node Group to be assigned

Attribute	Description
	the Target Status. • The percentage thresholds you enter must not exceed 100 percent.

Monitor Router Redundancy Groups (*NNMi Advanced*)

NNMi monitors state and priority information for any discovered HSRP and VRRP objects in the network. These objects include Router Redundancy Members and Tracked Objects. See Router Redundancy Group View for more information about Router Redundancy Groups and the HSRP or VRRP objects associated with them.

The polling interval used is the Fault Polling Interval that is set for the node associated with the Router Redundancy Member or Tracked Object.

If you do not want these objects polled:

- Set the Management Mode for each node to **Not Managed** or **Out of Service**. See "Stop or Start Managing a Node, Interface, Card, Address, or Node Component" (on page 250) for more information about Management Mode.
- Disable all Router Redundancy Group monitoring. See "Set Global Monitoring" (on page 217).

NNMi Advanced also uses these HSRP and VRRP objects when calculating a Path View between two nodes that have IPv4 addresses. See Path View with NNMi Advanced for more information.

Current Health of the State Poller Service

At any time, you can check the current health statistics about the State Poller Service.

To see a report of the health of the State Poller Service, click **Help** → **System Information** and navigate to the **State Poller** tab. For more information see Displaying NNMi System Information.

The State Poller Service contributes towards discovery and ongoing monitoring. See "About Each NNMi Service" (on page 46).

Verify Monitoring Configuration Settings

After you configure the monitoring settings, you can check the configuration for a particular object to verify that everything is working correctly. Examples of objects on which you can verify monitoring configuration settings include Nodes, Interfaces, IP addresses, Router Redundancy Groups, Tracked Objects, and Node Components. Use the **Actions** → **Monitoring Settings** menu item to display a report.

(*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager:

- Node managed by the Global Manager = Actions → Monitoring Settings displays a report, provided by the Global Manager (NNMi management server).
- Node managed by a Regional Manager = Actions → Monitoring Settings accesses that Regional Manager (NNMi management server) and requests the report.

Note: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.

To verify the monitoring configuration for a Node, Interface, IP address, or Card:

1. Navigate to the view for that object (for example, **Inventory** workspace, **Nodes)** view.
2. Select the object of interest by selecting the ☑ check box that precedes the object information.
3. Select **Actions** → **Monitoring Settings**.

   NNMi displays the monitoring configuration settings for the selected object.

Note: This menu item also is available on any object's form.

To verify the monitoring configuration for a Router Redundancy Member:

1. Navigate to a Router Redundancy Group view (for example, **Inventory** workspace, **Router Redundancy Groups** view).
2. Click the Open icon in the row representing the Router Redundancy Group configuration you want to see.
3. From the Router Redundancy Members tab, click the Open icon in the row representing the Router Redundancy Member configuration you want to edit.

   NNMi displays the monitoring configuration settings for the selected object.
4. Select **Actions** → **Monitoring Settings**.

To verify the monitoring configuration for a Tracked Object:

1. Navigate to a Router Redundancy view (for example, **Inventory** workspace, **Router Redundancy Groups** view).
2. Click the Open icon in the row representing the Router Redundancy Group configuration you want to see.
3. From the Router Redundancy Members tab, click the Open icon in the row representing the Router Redundancy Group Member configuration you want to see.
4. From the Tracked Objects tab, click the Open icon in the row representing the Tracked Object configuration you want to see.
5. Select **Actions** → **Monitoring Settings**.

   NNMi displays the monitoring configuration settings for the selected object.

To verify the monitoring configuration for a Node Component:

1. Navigate to the view for that object (for example, **Inventory** workspace, **Nodes** view).
2. Click the Open icon in the row representing the Node Component Configuration you want to see.
3. Select the **Node Component** tab.
4. Select the Node Component of interest by selecting the ☑ check box that precedes the object information.
5. Select **Actions** → **Monitoring Settings**.

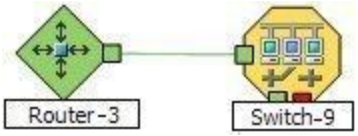
   NNMi displays the monitoring configuration settings for the selected object.

Check status and connectivity of important interfaces.

1. Open a Layer 2 Neighbor View map of each important interface's parent device. See Viewing Maps (Network Connectivity).
2. Each connected interface has a little square symbol around the edge of the parent device's map symbol. For example:

3. Hover your mouse over the square to verify the identify of your important interface on the map.
4. Verify that the status color of each important interface is not Unknown or **Unmanaged**[1] (see About Status Colors). For example:

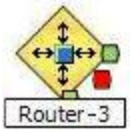
5. By default, NNMi only monitors the health of connected interfaces. A line appears on the map between interfaces when they are connected. For example:



6. If you need to add a connection, see "Add or Delete a Layer 2 Connection" (on page 176).

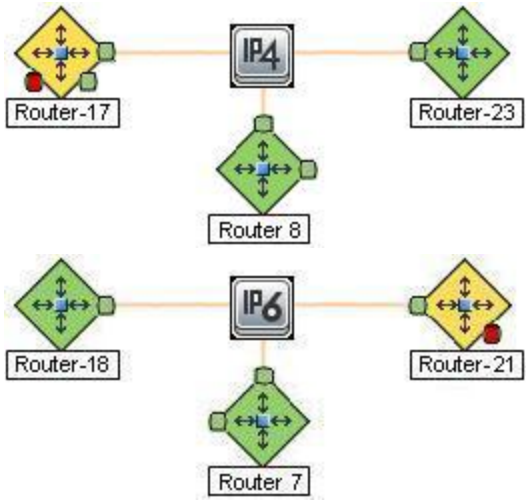
Check status and connectivity of important addresses.

1. Open a Layer 3 Neighbor View map of each important parent device. See Viewing Maps (Network Connectivity).
2. Each address that is connected to another address in the same subnet has a little hexagon symbol around the edge of the parent device's map symbol. For example:

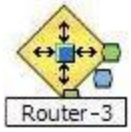


3. Hover your mouse over the hexagon to verify the identify of your important address on the map.
4. NNMi monitors the health of addresses only if you enable ICMP Address Monitoring. A line appears on the map between addresses when they are connected. The line represents the subnet. For example:

[1]Indicates the Management Mode is "Not Managed" or "Out of Service".



5. If ICMP Address Monitoring is enabled, verify that the status color of each important address is not Unknown or **Unmanaged**[1] (see About Status Colors). For example:



6. If you need to add a connection, see "Add or Delete a Layer 2 Connection" (on page 176).

See "Configure Monitoring Behavior" (on page 216) for information about establishing monitoring behavior.

Monitor Status Distribution for Network Objects

NNMi enables you to view the overall health of your network by providing Stacked Area Graphs that display the distribution of Node, Interface, and IP Address Status information over time.

Tip: If you do not want to display unpolled objects (No Status), use the **File** → **Select Area** menu option and clear the **No Status** check box.

To view Status Distribution Graphs:

1. Select **Tools** → **Status Distribution Graphs**.
2. Select the object type for which you want to display Status distribution. For example, **Node Status**.

   NNMi displays a Stacked Area Graph of the distribution of the object's Status over time.

   See **Help** → **Using Stacked Area Graphs** from the Graph menu bar for more information about using Stacked Area Graphs.

See "Configure Monitoring Behavior" (on page 216) for information about establishing monitoring behavior.

[1] Indicates the Management Mode is "Not Managed" or "Out of Service".

Stop or Start Managing a Node, Interface, Card, Address, or Node Component

NNMi administrators can specify that a node, interface, card, address, or node component should no longer be discovered or monitored (Management Mode = Unmanaged) or is out of service (Management Mode = Out of Service). For additional information see the form for each object:

Reasons you might want to change the management mode include:

- The node is temporarily out of service.
- You determine that NNMi should never monitor a particular node, interface, card, IP address, or node component.

NNMi provides two management modes for each object (as described in the table). For more information, see the following topics:

- "View the Management Mode for Objects in Your Network" (on page 251)
- "How NNMi Assigns the Management Mode to an Interface, Card, Node Component, or Address" (on page 254)
- "How the NNMi Administrators Change a Management Mode" (on page 256)
- "Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " (on page 257)

Management Modes

Name	Description
Node Management Mode or Management Mode	For Node objects, this value is set by the NNMi administrator. The node-level Management Mode affects the Management Mode of objects associated with the node. For interface, card, node component, or address, the Management Mode cannot be set by the NNMI administrator. NNMi calculates value. The Management Mode for an interface, card, or node component is computed based on the Management Mode for the node. The Management Mode value for an address is calculated based on the Management Mode for the associated interface (if any) or based on the Management Mode for the node. Possible values include: **Managed** - Used to indicate a node, interface, or address should be discovered and monitored by NNMi. **Not Managed** - Used to indicate that NNMi should not discover or monitor the object. For example, the object might not be accessible because it is in a private network. **Out of Service** - Used to indicate a node, interface, or address is unavailable because it is out of service. NNMi does not discover or monitor these objects. **Tip**: Some objects have child objects (for example, Nodes contain interfaces, and interfaces can contain IP addresses). To change the Management Mode back to **Managed** or **Inherited** for the selected object and all associated child objects, use the **Actions → Management Mode → Managed (Reset All)**.

Name	Description
Direct Management Mode	For interfaces, cards, node components, and addresses, this value is set by the NNMi administrator. NNMi uses this value to compute the Management Mode values in the previous row in this table. Possible values include: **Inherited** - For interfaces, cards, and node components, this value is used to indicate that the object should inherit the Management Mode from the node in which it resides. For addresses, this value is used to indicate that the Management Mode should be inherited from the associated interface, if one exists. Otherwise the Management Mode is inherited from the node in which it resides. **Not Managed** - Used to indicate that NNMi should not discover or monitor the object. For example, the object might not be accessible because it is in a private network. **Out of Service** - Used to indicate the object is unavailable because it is out of service. Reasons might include the interface, card, or node component is being repaired or there is a known problem with the address. NNMi does not discover or monitor these objects.

View the Management Mode for Objects in Your Network

NNMi provides the Management Mode workspace so that you can quickly view lists of all nodes, interfaces, cards, addresses, or node components that NNMi is not currently discovering or monitoring. For information about these views:

The following tables describes each possible Management Mode and Direct Management Mode value. The available Management Mode values depend on the object type (node, interface, card, address, or node component). Management Modes are either set automatically by NNMi or set by the NNMi administrators.

Management Mode Values

Object	Value	Description
Node	Managed	Used to indicate that the node should be managed by NNMi. This means it will be discovered and monitored.
Node	Not Managed	Used to indicate you do not plan to manage the node. For example, the node might not be accessible because it is in a private network. NNMi does not discover or monitor these objects.
Node	Out of Service	Used to indicate the node is unavailable because it is out of service. Reasons might include that the device is being repaired or there is a known problem with the device. NNMi does not discover or monitor these objects.

Direct Management Mode Values

Object	Value	Description
Interface, Card, Address, or Node Component	Not Managed	Used to indicate you do not plan to manage the interface, card, address, or node component. After the Direct Management Mode is set to **Not Managed**, NNMi no longer discovers or monitors the object.

Object	Value	Description
Interface, Card, Address, or Node Component	Out of Service	Used to indicate that the object is out of service. NNMi does not discover or monitor these objects. An interface, card, or node component will not be managed again until the Direct Management Mode is set to **Inherited** and its associated node is set to **Managed**. An address will not be managed again until the Management Mode of any associated interface is set to Inherited and the node's Management Mode is set to Managed.
Interface, Card, Address, or Node Component	Inherited	Used to indicate that the object should assume the Management Mode of the node in which it is hosted. **Note**: To manage the interface, card or node component, the Management Mode of the node in which the interface is hosted must be **Managed**.
Address	Inherited	The address assumes the Management Mode of the interface, if any, with which the address is associated. If the address is not associated with an interface, it assumes the Management Mode of the node in which it is hosted. **Note**: To manage the address, the Management Mode of the address' interface, if any, must be calculated to be **Managed**. The Management Mode of the node in which the interface and address are hosted must be set to **Managed**.

Unmanaged Nodes View

The Unmanaged Nodes view identifies all of the nodes with a management mode of either **Not Managed** or **Out of Service**. These are the nodes that are no longer being discovered or monitored.

Use this view to select nodes and change the Management Mode to **Managed**. For information:

For each node, you can identify its overall status (for example, **Normal**, **Warning**, **Minor**, **Major**, **Critical** and **Unknown**), device category (for example, switch), name, system name, management address, system location (the current value of the sysLocation MIB variable), device profile, whether the SNMP Agent is enabled, the date and time the status was last modified, and any notes included for the node.

See "Using the Nodes View" for more information about uses for nodes views.

Related Topics

"How the NNMi Administrators Change a Management Mode" (on page 256)

"How NNMi Assigns the Management Mode to an Interface, Card, Node Component, or Address" (on page 254)

"Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " (on page 257)

"View the Management Mode for Objects in Your Network" (on page 251)

Unmanaged Interfaces View

Tip: See Interface Form for more information about the attributes that appear in each column in this view.

The Unmanaged Interfaces view identifies all of the interfaces with a Management Mode of **Not Managed** or **Out of Service**. These are the interfaces that are no longer being discovered or monitored.

Use this view to select interfaces and change the Management Mode to **Managed**. For information:

For each interface, you can identify the interface's overall status (for example, **Normal**, **Warning**, **Minor, Major, Critical**, and **Unknown**), administrative state, operational state, the management mode of the interface, the management mode of the associated node, the node on which the interface resides (Hosted on Node), the interface name, type, speed, and alias, the date the interface status and state was last changed, and any notes included for the interface.

See Interfaces View (Inventory) for more information about uses for the interfaces views.

Related Topics

"How the NNMi Administrators Change a Management Mode" (on page 256)

"How NNMi Assigns the Management Mode to an Interface, Card, Node Component, or Address" (on page 254)

"Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " (on page 257)

"View the Management Mode for Objects in Your Network" (on page 251)

Unmananaged Addresses View

Tip: See IP Address Form for more information about the attributes that appear in each column in this view.

The Unmanaged Addresses view identifies all of the addresses with a Management Mode of **Not Managed** or **Out of Service**. These are the addresses that are no longer being discovered or monitored.

Use this view to select addresses and change the Management Mode to **Managed**. For information:

For each IP address, you can identify its status, state, management mode, the management mode of its associated node, the IP address value, the name of the interface on which the address resides (**In Interface**), the name of the node on which the address resides (**Hosted on Node**), subnet (**In Subnet**) and prefix length (**PL**), the date and time in which the status was last modified, and any notes included for the IP address.

See IP Addresses View (Inventory) for more information about uses for address views.

Related Topics

"How the NNMi Administrators Change a Management Mode" (on page 256)

"How NNMi Assigns the Management Mode to an Interface, Card, Node Component, or Address" (on page 254)

"Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " (on page 257)

"View the Management Mode for Objects in Your Network" (on page 251)

Unmanaged Cards View

Tip: See Card Form for more information about the attributes that appear in each column in this view.

The Unmanaged Cards view identifies all of the cards with a Management Mode of **Not Managed**or **Out of Service**. These are the cards that are no longer being discovered or monitored.

Use this view to select cards and change the Management Mode to **Managed**. For information:

For each card, you can identify its status, management mode, the management mode of the node on which it resides, the administrative state, the operational state, the name of the node on which the card resides (**Hosted On Node**), the date and time the status was last modified, its name, model, type, serial number, firmware version, hardware version, software version, index, the name of the card on which the card resides, if any, any Redundant Group to which the card belongs, the date and time the state was last modified, the card Description, and any notes included for the card.

Related Topics

"How the NNMi Administrators Change a Management Mode" (on page 256)

"How NNMi Assigns the Management Mode to an Interface, Card, Node Component, or Address" (on page 254)

"Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " (on page 257)

"View the Management Mode for Objects in Your Network" (on page 251)

Unmanaged Node Components View

Tip: See Node Component Form for more information about the attributes that appear in each column in this view.

The Unmanaged Node Components view identifies all of the Node Components with a Management Mode of **Not Managed** or **Out of Service**. These are the Node Components that are no longer being discovered or monitored.

Use this view to select Node Components and change the Management Mode to **Managed**. For information:

For each Node Component, you can identify its Status, Management Mode, the Management Mode of the node on which it resides, its Name, type, the name of the node on which it resides (**Hosted On Node**), and the date and time the Status was last modified.

Related Topics

"How the NNMi Administrators Change a Management Mode" (on page 256)

"How NNMi Assigns the Management Mode to an Interface, Card, Node Component, or Address" (on page 254)

"Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " (on page 257)

"View the Management Mode for Objects in Your Network" (on page 251)

How NNMi Assigns the Management Mode to an Interface, Card, Node Component, or Address

NNMi administrators can instruct NNMi to no longer manage an interface, card, node component, or address by setting the *Direct Management Mode* value (see "How the NNMi Administrators Change a

Management Mode" (on page 256)). NNMi then calculates the overall Management Mode based on the current Management Mode of all the associated objects.

For example, if you are specifying the Direct Management Mode for an address, NNMi uses the following values to determine the Management Mode value for the address:

- Direct Management Mode you enter for the address
- Management Mode of the associated interface, if any
- Management Mode of the node that contains the address

The following table lists possible value combinations for each object's management mode.

To check the current Management Mode setting for objects see "View the Management Mode for Objects in Your Network" (on page 251) and .

Interface, Card, and Node Component

Management Mode (Node) Calaculated by NNMi	Direct Management Mode (Interface, Card, or Node Component)	Management Mode (Interface, Card, or Node Component)
Managed	Inherited	Managed
Not Managed	Inherited	Not Managed
Out of Service	Inherited	Out of Service
Managed	Not Managed	Not Managed
Not Managed	Not Managed	Not Managed
Out of Service	Not Managed	Not Managed
Managed	Out of Service	Out of Service
Not Managed	Out of Service	Out of Service
Out of Service	Out of Service	Out of Service

Address

Management Mode (Node)	Direct Management Mode (Interface)	Direct Management Mode (Address)	Management Mode (Address)
Managed	Inherited	Inherited	Managed
Not Managed	Inherited	Inherited	Not Managed
Out of Service	Inherited	Inherited	Out of Service
Managed	Not applicable*	Inherited	Managed
Not Managed	Not applicable*	Inherited	Not Managed
Out of Service	Not applicable*	Inherited	Out of Service
Managed	Not Managed	Inherited	Not Managed

Management Mode (Node)	Direct Management Mode (Interface)	Direct Management Mode (Address)	Management Mode (Address)
Not Managed	Not Managed	Inherited	Not Managed
Out of Service	Not Managed	Inherited	Not Managed
Managed	Not Managed	Not Managed	Not Managed
Not Managed	Not Managed	Not Managed	Not Managed
Out of Service	Not Managed	Not Managed	Not Managed
Managed	Not applicable*	Not Managed	Not Managed
Not Managed	Not applicable*	Not Managed	Not Managed
Out of Service	Not applicable*	Not Managed	Not Managed
Managed	Out of Service	Inherited	Out of Service
Not Managed	Out of Service	Inherited	Out of Service
Out of Service	Out of Service	Inherited	Out of Service
Managed	Out of Service	Out of Service	Out of Service
Not Managed	Out of Service	Out of Service	Out of Service
Out of Service	Out of Service	Out of Service	Out of Service
Managed	Not applicable*	Out of Service	Out of Service
Not Managed	Not applicable*	Out of Service	Out of Service
Managed	Not applicable*	Out of Service	Out of Service

* Used to indicate there is no associated interface

How the NNMi Administrators Change a Management Mode

Caution: (*NNMi Advanced - Global Network Management feature*) If your NNMi console is a Global Manager and the selected object is being managed by a Regional Manager (another NNMi management server in your network environment), you cannot change the Management Mode setting unless you log onto the Regional Manager (NNMi management server).

The NNMi administrator can change the Management Mode in one of the following ways:

- Open the object's form, do one of the following, and then select **File** → **Save and Close**:
  - Use the Management Mode attribute's drop-down menu to choose an available Management Mode for that object.
  - Use **Actions** → **Management Mode** and choose an available Management Mode for that object.

  Note: If you are updating the Direct Management Mode for an interface, card, node component, or address, NNMi also updates its Management Mode value after you reopen or refresh the form.
- Open a view that contains the object and do the following:

a. Select the object of interest:
   - In a table view, check the ☑ selection box that precedes the object information.
   - In a map view, single-click the object.

b. Select **Actions** → **Management Mode** and choose an available Management Mode for that object.

   Tip: Some objects have child objects (for example, Nodes contain interfaces, and interfaces can contain IP addresses). To change the Management Mode back to **Managed** or **Inherited** for the selected object and all associated child objects, use the **Actions** → **Management Mode** → **Managed (Reset All)**.

Note: The NNMi administrator can also change the management mode of a node or interface using the nnmmanagementmode.ovpl command.

Make sure you review this information: "Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " (on page 257).

Related Topics:

"How NNMi Assigns the Management Mode to an Interface, Card, Node Component, or Address" (on page 254)

"View the Management Mode for Objects in Your Network" (on page 251)

Understand the Effects of Setting the Management Mode to Not Managed or Out of Service

NNMi administrators can instruct NNMi to no longer manage an interface, card, node component, or address by selecting a *Management Mode* value on the object's form or by using **Actions** → **Management Mdoe**. See "How the NNMi Administrators Change a Management Mode" (on page 256).

The results of setting the management mode to **Not Managed** or **Out of Service** for an object, depends on whether you are setting the value for a node, interface, address, card, or node component:

- **Nodes: Management Mode**

  For nodes, setting the Management Mode to **Not Managed** or **Out of Service** has the following effects:

  - No incidents are generated for the node
  - The node's SNMP Agent is excluded from fault polling.
  - The node's interfaces or addresses are excluded from fault and performance polling.
  - NNMi quits gathering Node Component data.
  - NNMi deletes all Polled Instances associated with the **Not Managed** or **Out of Service** node.
  - The Active State for any Custom Poller Nodes associated with the **Not Managed** or **Out of Service** node becomes **Inactive**.
  - The node is removed from any associated Router Redundancy Groups.
  - Traps related to the node, interface, card, node component, or address, (for example, coldStart or warmStart) are not stored.
  - The node is excluded from discovery.
  - **Actions** → **Configuration Poll** is no longer available for this node.

- The status of a node is set to **No Status**.
- **Actions → Status Poll** is no longer available for the node or incident related to that node.

- **Interfaces: Direct Management Mode**

  For interfaces, setting the Direct Management Mode to **Not Managed** or **Out of Service** has the following effects:

  - No incidents are generated for the interface.
  - The interface and any related addresses are excluded from fault and performance polling.
  - The Administrative State and Operational State of the interface are set to **Not Polled.**
  - The Status of the interface is set to **No Status**.
  - Traps related to the interface (for example, LinkUp or LinkDown), will not be stored.

- **IPv4 / IPv6 Addresses: Direct Management Mode**

  For addresses, setting the Direct Management Mode to **Not Managed** or **Out of Service** has the following effects:

  - No incidents are generated for the address.
  - The State of the address is set to **Not Polled**.
  - The address is excluded from fault and performance polling.
  - Traps related to the address are not stored.

- **Cards and Node Components: Direct Management Mode**

  Cards and Node Components, setting the Direct Management Mode to **Not Managed** or **Out of Service** has the following effects:

  - No incidents are generated for the card or node component.
  - The State of the object is set to **Not Polled**.
  - The card or node component is excluded from fault and performance polling.
  - The Status remains set to the last known Status value.
  - Traps related to the card or node component are not stored.

NMi provides the Management Mode workspace so that you can quickly view lists of all nodes, interfaces, cards, addresses, or node components that NNMi is not currently discovering or monitoring. For information about these views:

To change the Management Mode back to **Managed**, use the **Actions → Management Mode → Managed**.

Tip: Some objects have child objects (for example, Nodes contain interfaces, and interfaces can contain IP addresses). To change the Management Mode back to **Managed** or **Inherited** for the selected object and all associated child objects, use the **Actions → Management Mode → Managed (Reset All)**.

Configuring the NNMi User Interface

NNMi enables an NNMi administrator to configure the following global user interface features:

- The console time out interval
- The initial map view to display in the Topology Maps workspace
- Whether NNMi displays unlicensed features that require a special license, such as NNMi Advanced

For information about the additional user interface configurations available, including creating user accounts, configuring Node Group map settings, setting the default values for maps and Line Graphs, and configuring menus and menu items:

To configure user interface features, do the following:

1. Navigate to the **User Interface Configuration** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **User Interface Configuration**.
2. Make your Global Control configuration choices (see the Global Control Attributes table).
3. Make your additional configuration choices. Click here for a list of choices .
4. Click **Save and Close** to apply your changes.
5. To apply your Console Timeout or Initial View configuration changes, sign out of the NNMi console. After restarting the console, your changes should take effect.

Global Control Attributes for User Interface Configuration

Attribute	Description
Console Timeout	NNMi's default session inactivity timeout value is 18 hours. Use this attribute to change the timeout interval in days, hours, and minutes. **Note**: The minimum timeout value is 1 minute. After this period, if no mouse movement occurs, the consoles locks and the user is prompted to sign in again. **Tip**: If your network operation center (NOC) has a large screen where a map of the most important nodes is continuously displayed, use a launched view. See "Launch a Troubleshooting Workspace View" (on page 975). A map session launched with a URL never times out. The map launched automatically updates every 30 seconds. (If you are using Mozilla Firefox, also see Configure Mozilla Firefox Timeout Interval.)
Initial View	Use this attribute to specify the initial view to be automatically displayed in the NNMi console by default. When selecting a view from the drop-down menu list, note the following: • Use the value **None (blank)** to specify that you do not want a default view automatically displayed by default. • If the Node Group you select has been removed, NNMi uses None (blank view). • To select a Node Group map you have created: ▪ *Prerequisite*. Use the **Node Group Map Settings** configuration workspace to create a Node Group map and enter a Topology Ordering number that lists the Node

Attribute	Description
	Group map as the first or last map in the Topology Maps workspace. See "Configure Basic Settings for a Node Group Map" (on page 282) for more information. ■ For the **Initial View** attribute: ○ If you placed the Node Group map as the first entry in the Topology Maps workspace, select **First Node Group in Topology Maps workspace.** ○ If you placed the Node Group map as the last entry in the Topology Maps workspace, select **Last Node Group in Topology Maps workspace**.
Dafault Author	The Default Author attribute specifies the Author attribute NNMi should use by default when you create a new instance of an object in NNMi. For example you might create a new incident configuration. The Author attribute identifies who provided that instance of an object. The Author attribute value is also useful for filtering objects in certain views and when using the NNMi Export/Import feature. Either keep the Default Author value of **Customer** or enter an Author attribute value representing you or your organization. The Default Author value you specify then appears in the Author selection list in any appropriate form and appears by default as the Author value when you create a new instance of an object. See Author form for important information.

Attribute	Description
Enable URL Redirect	Before enabling URL Redirect, verify that the NNMi management server's official Fully Qualified Domain Name (FQDN) is set correctly and the DNS name is resolvable from any remote systems that need to access the NNMi management server. If the official FQDN does not meet these requirements, users will view errors when trying to access the NNMi console. To view the NNMi management server's official FQDN, do one of the following: • Select **Help** → **System Information** and click the **Server** tab. • Use the nnmhealth.ovpl command line tool. • Use the nnmofficialfqdn.ovpl command line tool. **Tip**: To change the official FQDN, use the nnmsetofficialfqdn.ovpl command line tool. When ☑ URL Redirect is enabled, a user can sign into the NNMi console using any hostname (*not case-sensitive*) or IP address that is valid for the NNMi management server. (*NNMi Advanced's Global Network Management feature or HP Network Node Manager i Software Smart Plug-ins (iSPIs)*) For environments configured with Single Sign-On (SSO) among multiple servers (which normally requires users to provide the official Fully Qualified Domain Name (FQDN) that was configured during NNMi installation), this attribute enables NNMi to redirect URLs that contain the IP address or any hostname associated with the NNMi management server to the official FQDN. **Note**: All NNMi management servers participating in Global Network Management or Single Sign-On (SSO) must have synchronized time stamps.
Show Unlicensed Features	By default, NNMi displays menus, views, and workspaces that require an additional license. If you do not have the required license, NNMi labels these features as `Unlicensed` or `Evaluation`. `Evaluation` indicates the License Type is `Instant-On` or `Temporary`. To determine which Unlicensed or Evaluation features could be displayed in your NNMi console, click here for more information. • Access **Help** → **Documentation Library** → **Release Notes**and click the **Licensing** link. • Access **Help** → **System Information** and click the **Extension** tab. • Access **Help** → **System Information** and click the **Product** tab and click the **View Licensing Information** button. See "Purchase an HP Network Node Manager i Smart Plug-in" (on page 953) for more information about possible HP Smart Plug-ins. To hide Unlicensed or Evaluation features from the NNMi console, clear the **Show Unlicensed Features**☐ check box. (Recommended if you do not plan to install a permanent license for these features.) To display Unlicensed or Evaluation features in the NNMi console, select the **Show Unlicensed Features**☑ check box.

Registration Attributes for User Interface Configuration

Attribute	Description
Last Modified	Indicates the last date and time that any of the user interface attributes were modified.

NNMi also enables you to configure features specific to Node Group Maps. See "Define Node Group Map Settings" (on page 281) for more information.

Controlling Access to NNMi

As administrator, you control who accesses the NNMi console. The role you assign to each user determines which workspaces and actions are available within the NNMi console. To accomplish this, configure the following:

- "Configure Sign-In Access" (on page 262)
- "Disable a User's Access to NNMi (User Principals)" (on page 272)
- "Troubleshoot NNMi Access" (on page 273)
- "Set Up Command Line Access" (on page 275)

Tip: You can configure the NNMi management server to use https/SSL (secure sockets layer encryption) so that data passed between client and server is encrypted. (See the "Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`).

When configuration is complete, share information about NNMi with your team:

- "Communicate to Your Team" (on page 277)

Configure Sign-In Access

First decide which pre-defined NNMi role is appropriate for each user in your environment:

- "Roles Provided in NNMi" (on page 263)

Then choose how to configure access to the data required for NNMi logins:

1. "Control Access with NNMi Accounts" (on page 266)
   User names, passwords, and role assignments are stored in the NNMi database.
2. "Control Access Using Both Directory Service and NNMi" (on page 268)
   User names must be stored in both the directory service database and the NNMi database. Passwords are stored in the directory service database. Role assignments are stored in the NNMi database. NNMi communicates with the directory service using Lightweight Directory Access Protocol (LDAP).
3. "Control Access with a Directory Service" (on page 271)
   User names, passwords, and role assignments are stored in the directory service. NNMi communicates with the directory service using Lightweight Directory Access Protocol (LDAP).

Which Database Stores the Information?

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

Roles Provided in NNMi

As NNMi administrator, you assign a preconfigured NNMi role to each NNMi user. (For more information, see "Configure Sign-In Access" (on page 262).)

User roles determine access to the NNMi console workspaces, forms, and actions. NNMi provides the following roles:

- Administrator
- Operator Level 2
- Operator Level 1
- Guest

You cannot create additional roles or change the names of the roles that NNMi provides.

Caution: Do not use the `system` role. NNMi provides the `system` role for accessing NNMi the first time during installation and for command line access (see "Set Up Command Line Access" (on page 275)).

NNMi provides a special `Web Service Client` role to provide access for software that is integrated with NNMi (for example, see "HP RAMS MPLS WAN Configuration (NNMi Advanced)" (on page 869)).

For details about roles, see the following topics:

- "Determine which NNMi Role to Assign" (on page 263) (controls access to views and forms)
- "Control Menu Access" (on page 264) (NNMi administrators control which roles can access a small subset of Action menu items)
- "Configure Basic Settings for a Node Group Map" (on page 282) (for each Node Group Map, the *Minimum Role for Saving Map Layout* attribute setting controls the minimum user role required for saving the layout after the user repositions nodes on the map)

Determine which NNMi Role to Assign

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

As NNMi administrator, you can change the following aspects of role definitions:

- "Control Menu Access" (on page 264) (restrict access to certain NNMi Actions menu items and Tools menu items - provide tighter security than those enforced by the default settings) See also "Configure Launch Actions" (on page 878) for more information about adding options to the NNMi Actions menu.
- "Configure Basic Settings for a Node Group Map" (on page 282). (for each Node Group Map, the *Minimum Role for Saving Map Layout* attribute setting controls the minimum user role required for saving the layout after the user repositions nodes on the map)
- "Set Up Command Line Access" (on page 275) (control access to NNMi command line commands)

The following table lists the roles required to access NNMi workspaces. You cannot modify role settings for workspaces. See About Workspaces for more information about workspaces. See Views Provided by NNMi for more information about the views provided in each workspace.

Access to Workspaces

Workspaces	Guest Role	Level 1 Operator Role	Level 2 Operator Role	Administrator
All views in the Incident workspaces	Yes	Yes	Yes	Yes
All views in the Topology workspaces	Yes	Yes	Yes	Yes
All views in the Monitoring workspace	Yes	Yes	Yes	Yes
All views in the Troubleshooting workspace	Yes	Yes	Yes	Yes
All views in the Inventory workspace	Yes	Yes	Yes	Yes
All views in the Management Mode workspace			Yes	Yes
All views in the Configuration workspace				Yes

The following table provides some examples of how NNMi roles control permission for modifications to certain forms. You cannot modify role settings for forms.

Access to Forms (some examples)

Forms	Guest Role	Level 1 Operator Role	Level 2 Operator Role	Administrator
Node forms	Read-Only	Read-Write except Management Mode field which is Read-Only	Read-Write	Read-Write
Interface forms	Read-Only	Read-Write except Management Mode field which is Read-Only	Read-Write	Read-Write
IP Address forms	Read-Only	Read-Write except Management Mode field which is Read-Only	Read-Write	Read-Write
IP Subnet forms	Read-Only	Read-Write except Management Mode field which is Read-Only	Read-Write	Read-Write
Incident forms	Read-Only	Read-Write	Read-Write	Read-Write
Node Group forms	Read-Only	Read-Only	Read-Only	Read-Write
Configuration Forms				Read-Write

Control Menu Access

Access to the Tools and Actions menu items is controlled by user role. (See "Determine which NNMi Role to Assign" (on page 263) for additional information about role limitations.)

Note the following:

- All menu items are visible to users, but an *Access Denied* message displays when any user with insufficient privileges tries to use a menu item. For example, both Level 1 or Level 2 Operators are denied access to the Communication Settings action.
- You can restrict access to certain URL actions (provide tighter security than those enforced by the default settings). See "Configure Menu Item Context Basic Details" (on page 877) for more information about configuring actions.

Role Based Access the Tools Menu:

Access to the NNMi Tools menu items is determined by user role.

NNMi Tools Menu Access Limitations

Tools Menu Item	Default Role
Find Node	Guest
Find Attached Switch Port	Operator Level 2
Incident Actions Log	Administrator
Load MIB	Administrator
MIB Browser	Operator Level 2
NNMi Self-Monitoring Graphs	Administrator
NNMi Status	Operator Level 1
NNMi System Health Report	Adminstrator
Restore All Default View Settings	Guest
Signed In Users	Administrator
Sign In/Sign Out Audit Log	Administrator
Status Distribution Graphs	Operator Level 2
Trap Analytics (iSPI NET only)	Administrator
Upload Local MIB File	Adminstrator
Visio Export (iSPI NET only)	Operator Level 2

Role Based Access the Actions Menu:

Access to the NNMi Actions menu is determined by user role.

URL Action Access Limitations

Action Menu Item	Default Role
Browse MIB	Operator Level 2
Communication Settings	Administrator
Configuration Poll	Operator Level 2
Graphs	Operator Level 1
List Supported MIBs	Operator Level 2
Management Mode	Operator Level 2
Monitoring Settings	Operator Level1
Ping (from server)	Operator Level 1
Show All Incidents	Operator Level 1
Show All Open Incidents	Operator Level 1
Show Attached End Nodes	Operator Level 2
Show Members	Operator Level 1
Status Details	Operator Level 1
Status Poll	Operator Level 2
Traceroute (from server)	Operator Level 1
Telnet...(from client)	Operator Level 2

See Investigate and Diagnose Network Problems for more information about these actions.

Control Access with NNMi Accounts

To configure NNMi to store user names, passwords, and role assignments in the NNMi database, use the following instructions.

Which Database Stores the Information?

User Name	Password	Role Mapping
NNMi	NNMi	NNMi

Tip: If you prefer to configure NNMi logins to use data stored in the directory service database, see "Control Access Using Both Directory Service and NNMi" (on page 268) or "Control Access with a Directory Service" (on page 271). If you are not using LDAP or the Active Directory Service to manage NNMi users, you can also use the nnmprincipalconfig.ovpl command to add, delete, or modify user names, passwords, and role assignments.

To grant NNMi access to a user:

1. Navigate to the **Account Mapping** form.
   a. From the workspaces navigation panel, select the **Configuration** workspace.
   b. Select **User Interface Configuration**.
   c. Select the **User Accounts** tab.
   d. Do one of the following:
      - To create new configuration, click the New icon, and continue.
      - To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
2. Locate the **Account** attribute, and click the Lookup icon. (See "Using the Account Mapping Form" for more information.)
   - To create new account, click the New icon and provide the required user name and password. See "Using the User Account Form" for more information.) Then click **Save and Close** to return to the **Account Mapping** form.

     Tip: The user name you just created is added to the User Principals view.
   - To select an NNMi user configuration (user name and password), click the Quick Find icon and make a selection.
3. Locate the **Role** attribute.

   Select a predefined NNMi role from the drop-down menu. See "Roles Provided in NNMi" (on page 263) for more information.

   Note: If you accidentally assign two different NNMi roles to the same NNMi user name, NNMi uses the higher level role.
4. Click **Save and Close** to save your changes and return to the **User Accounts** form.

Related Topics

"Change Password, Name, or Role Assignment" (on page 267)

"Disable a User's Access to NNMi (User Principals)" (on page 272)

"Restore the Administrator Role" (on page 275)

Change Password, Name, or Role Assignment

If configuring NNMi to store user names, passwords, and role assignments in the NNMi database, use the following instructions.

Which Database Stores the Information?

User Name	Password	Role Mapping
NNMi	NNMi	NNMi

Note: If configuring NNMi to use the directory service database, see "Change Role Assignment for a Directory Service User Name" (on page 270) or "Control Access with a Directory Service" (on page 271).

Only NNMi administrators can change account details.

Note: If you are not using LDAP or the Active Directory Service to manage NNMi users, you can also use the nnmprincipalconfig.ovpl command to add, delete, or modify user names, passwords, and role assignments.

To change an NNMi user name:

You must "Disable a User's Access to NNMi (User Principals)" (on page 272), and then recreate the account mapping (see "Control Access with NNMi Accounts" (on page 266)).

Note: If you disable NNMi access for a user who is currently signed into the NNMi console, the change does not take effect until the user signs in next time.

To change an NNMi password:

1. Navigate to the **User Accounts** view.
   a. From the Workspaces navigation panel, select the **Configuration** workspace.
   b. Select **User Interface Configuration**.
   c. Select the **User Accounts** tab.
2. Click the Open icon in the row representing the account you want to edit.
3. To change the user's password, locate the **Account** attribute:
   a. Click the Lookup icon and select **Open** to access the User Account form.
   b. Edit the **Password** value. Type up to 40 alpha-numeric characters, punctuation, spaces, and underline characters.
   c. Click **Save and Close** to return to the **Account Mapping** form.
4. Click **Save and Close**. NNMi immediately implements your changes.

   Note: If you change the password for a user who is currently signed into the NNMi console, the change does not take effect until the user signs in next time.

To change an NNMi role assignment:

1. Navigate to the **User Accounts** view.
   a. From the Workspaces navigation panel, select the **Configuration** workspace.
   b. Select **User Interface Configuration**.
   c. Select the **User Accounts t**ab.
2. Click the Open icon in the row representing the account you want to edit.
3. To change the user's NNMi role assignment, locate the **Role** attribute.

   Click the drop-down list and select the new role for the current NNMi user. See "Determine which NNMi Role to Assign" (on page 263) for more information about NNMi user roles and their privileges.
4. Click **Save and Close**.

   Note: If you change the role for a user who is currently signed into the NNMi console, the change does not take effect until the user signs in next time.

Control Access Using Both Directory Service and NNMi

To configure NNMi to store role assignments in the NNMi database, but rely on your directory service for user names and passwords, use the following instructions. NNMi communicates with the directory service

using Lightweight Directory Access Protocol (LDAP).

Which Database Stores the Information?

User Name	Password	Role Mapping
Both	Directory Service	NNMi

Tip: If you prefer to configure NNMi logins to only use data stored in the directory service database, see "Control Access with a Directory Service" (on page 271).

To enable NNMi to communicate with your environment's directory service:

1. Modify the `ldap.properties` file. See the "Integrating NNMi with a Directory Service through LDAP" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

   Note: To make changes to a user name or password, you must now use the appropriate process for making changes to your environment's directory service software and make changes in NNMi (see "Change Role Assignment for a Directory Service User Name" (on page 270)).

2. **If you are switching from previously storing passwords in the NNMi database, do this step:**

   Note: The password information is no longer required to be in the NNMi database.

   a. Navigate to the **User Principals** view:

      i. From the workspace navigation panel, select the **Configuration** workspace.

      ii. Select **User Interface Configuration**.

      iii. Select the **User Principals** tab.

   b. Delete all objects in the User Principals view:

      i. Select all objects by clicking ☑ in the column heading row.

      ii. Click the ✕ button in the view toolbar.

   c. Navigate to the **User Accounts** view:

      i. From the workspace navigation panel, select the **Configuration** workspace.

      ii. Select **User Interface Configuration**.

      iii. Select the **User Accounts** tab.

   d. Verify that there are no remaining objects in this view. (When you deleted Principal objects, any associated account objects were automatically removed.)

   e. Continue with the next set of instructions.

3. **Establish the NNMi role mappings for directory service users**:

   a. Navigate to the **User Accounts** view:

      i. From the workspace navigation panel, select the **Configuration** workspace.

      ii. Select **User Interface Configuration**.

      iii. Select the **User Accounts** tab.

   b. Repeat this step for each NNMi user.

Associate each user name from your environment's directory service with a predefined NNMi role.

Click the New icon. See "Using the Account Mapping Form".

- Locate the **Account** attribute.

  Click the New icon. Enter the user name exactly as it appears in your environment's directory service database (case-sensitive). See "Using the Principal Form" (on page 272).

  Tip: The user name you just created is added to the User Principals view.

- Locate the **Role** attribute.

  Select a predefined NNMi role from the drop-down menu. See "Roles Provided in NNMi" (on page 263) for more information.

- Click **Save and Close** to save your role configuration changes and return to the **User Accounts and Roles** view.

c. Access the **Incident Browsing** workspace.

d. Open the **All Incidents** view.

Sort this view using the Assigned To (**AT**) column.

Verify that any *assigned* Incidents are associated with a valid user from the directory service database.

- If the user names previously stored in the NNMi database are the same (case-sensitive) as those defined in your environment's directory service, no changes are required for Incident assignments.
- If the user names have changed, reassign the Incidents to a valid user name (see Assign an Incident).

Change Role Assignment for a Directory Service User Name

If configuring NNMi to store user names and passwords in your environment's directory service, but to store role assignments in the NNMi database, use the following instructions. NNMi communicates with the directory service using Lightweight Directory Access Protocol (LDAP).

Which Database Stores the Information?

User Name	Password	Role Mapping
Both	Directory Service	NNMi

Note: To configure NNMi to use directory service for role assignments, see "Control Access with a Directory Service" (on page 271).

Only NNMi administrators can change the role assigned to each authorized directory service user name.

To change an NNMi role assignment:

1. Navigate to the **User Accounts** view.
   a. From the Workspaces navigation panel, select the **Configuration** workspace.
   b. Select **User Interface Configuration**.
   c. Select the **User Accounts** tab.

2. Click the Open icon in the row representing the mapping you want to edit.
3. Locate the **Role** attribute.

   Click the drop-down list and select the new NNMi role for the current user. See "Determine which NNMi Role to Assign" (on page 263) for more information about NNMi user roles and their privileges.

   Note: If you accidentally assign two different NNMi roles to the same NNMi user name, NNMi uses the higher level role.
4. Click **Save and Close**.

   Note: If you change the role for a user who is currently signed into the NNMi console, the change does not take effect until the user signs in next time.

Control Access with a Directory Service

To configure NNMi to rely on your environment's directory service for role assignments, user names, and passwords, use the following instructions. NNMi communicates with the directory service using Lightweight Directory Access Protocol (LDAP).

Which Database Stores the Information?

User Name	Password	Role Mapping
Directory Service	Directory Service	Directory Service

Tip: If you prefer to configure NNMi to store the role assignments in the NNMi database, see "Control Access Using Both Directory Service and NNMi" (on page 268).

To enable NNMi to communicate with your environment's directory service:

1. Modify the `ldap.properties` file. For more information, see the "Integrating NNMi with a Directory Service through LDAP" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

   Note: To make changes to NNMi access (user name, password, or NNMi role assignment), you must now use the appropriate process for making changes to your environment's directory service software.
2. **If you are switching from previously storing this data in the NNMi database:**
   a. Navigate to the **User Principals** view:
      i. From the workspace navigation panel, select the **Configuration** workspace.
      ii. Select **User Interface Configuration**.
      iii. Select the **User Principals** tab.
   b. Delete all objects in the view:
      i. Select all objects by clicking in the column heading row.
      ii. Click the button in the view toolbar.
   c. Navigate to the **User Accounts** view:
      i. From the workspace navigation panel, select the **Configuration** workspace.
      ii. Select **User Interface Configuration**.
      iii. Select the **User Accounts** tab.

d. Verify that there are no remaining objects in this view. (When you deleted Principal objects, any associated Account Mapping and User Account objects were automatically removed.)

e. Access the **Incident Browsing** workspace.

f. Open the **All Incidents** view.

Sort this view using the Assigned To (**AT**) column.

Verify that any *assigned* Incidents are associated with a valid user from the directory service database.

- If the user names previously stored in the NNMi database are the same (case-sensitive) as those defined in your environment's directory service, no changes are required for Incident assignments.
- If the user names have changed, reassign the Incidents to a valid user name (see Assign an Incident).

Disable a User's Access to NNMi (User Principals)

To deny a user's access to the console, delete their user configuration settings from the NNMi database.

Note: If you configured NNMi to store *role assignments* in your environment's directory service database (not the NNMi database), ignore this topic. To disable a user's access to NNMi, use the appropriate process required by your environment's directory service software (see "Control Access with a Directory Service" (on page 271)).

Caution: If you delete the last NNMi user assigned to the `Administrator` role, no one can access the Configuration workspace. See "Restore the Administrator Role" (on page 275) for more information about how to recover from this mistake.

To deny a user's access to NNMi:

1. Navigate to the **User Principals** view.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **User Interface Configuration**.
   c. Select the **User Principals** view.
2. Click **Show View in New Window**.

   NNMi displays the table view in a new window.
3. Check the selection box that precedes the row containing their user name.
4. Click the Delete icon.

   The user's configuration is automatically removed from *both* the User Principals view and the User Accounts view.

Tip: Access the Incident Browsing workspace. Open the All Incidents view. Sort this view using the Assigned To (AT) column. Reassign all Incidents associated with any user you deleted (see Assign an Incident).

Using the Principal Form

Principal object instances are automatically created when you configure access to the NNMi console according to the following scenarios:

- "Control Access with NNMi Accounts" (on page 266)
- "Control Access Using Both Directory Service and NNMi" (on page 268)

Note: If NNMi role assignments are stored in your environment's directory service database, this form is not needed. See "Control Access with a Directory Service" (on page 271).

To change a user name:

You must delete the Principal object and create a new one.

- "Change Password, Name, or Role Assignment" (on page 267)
  When user names, passwords, and role assignments are stored in the NNMi database.

  Which Database Stores the Information?

User Name	Password	Role Mapping
NNMi	**NNMi**	NNMi

- "Change Role Assignment for a Directory Service User Name" (on page 270)
  When user names and passwords are controlled by the directory service, and role assignments are stored in the NNMi database.

  Which Database Stores the Information?

User Name	Password	Role Mapping
Both	**Directory Service**	NNMi

To delete a Principal object (deny access to the NNMi console):

See "Disable a User's Access to NNMi (User Principals)" (on page 272).

When you delete a Principal object, the user configuration is removed from *both* the User Principals view and the User Accounts and Roles view.

Tip: If you delete an NNMi user, access the Incident Browsing workspace. Open the All Incidents view. Sort this view using the Assigned To (AT) column. Reassign all Incidents associated with any user you deleted (see Assign an Incident).

Troubleshoot NNMi Access

NNMi provides several tools to help you troubleshoot and monitor NNMi access:

- "View the Users who are Signed In to NNMi" (on page 273)
- "Audit NNMi User Activity" (on page 274)
- "Restore the Administrator Role" (on page 275)
- "Restore the System Role" (on page 275)

View the Users who are Signed In to NNMi

Use the **Tools** → **Signed in Users** menu option to view a list of the NNMi users who are currently signed in to NNMi. This tool is useful when you want to determine which users and systems are available. For example, you might want to view the users who are signed in before shutting down a system.

To see the list of users who are currently signed in to NNMi:

Select **Tools** → **Signed In Users**.

NNMi displays the number of users currently signed in to NNMi as well as each user name, IP address of the client that is running the NNMi console, and the time in which the user signed in to NNMi.

Audit NNMi User Activity

NNMi tracks a history of sign-in and sign-out activity for each NNMi user. This auditing information also includes a variety of information about user activity since the NNMi management server was last restarted.

NNMi stores the audit log files in the following directory:

- **Windows**:
  %NnmDataDir%\log\nnm\

  signin.0.0.log
- **UNIX**:
  /var/opt/OV/log/nnm/

  signin.0.0.log

Note: Log files are consecutively numbered. A new file is created each time you restart the NNMi management server. For example, <*logFileName*>.1.0.log and <*logFileName*>.2.0.log.

To see the most recent sign-in audit report:

1. A tool is available to NNMi administrators. In the console menu bar, select **Tools → Sign In/Out Audit Log**.
2. The log provides a variety of information about recent account activity. For example:

```
Sign In/Sign Out Audit Log
Jun 14, 2007 10:53:01.926 AM [ThreadID:719] com.hp.ov.ui.util.
SignInOutAuditLog logSignIn:

INFO: Successful Sign In
User: system
Role: Administrator (ADMIN)
Remote Host: <node IP address>
Remote Port: 1549
Locale: en_US
Sign In/Out Audit Since 6/14/07 9:33 AM
=======================================
Currently Signed In:
#1: system <node IP address> 6/14/07 10:53 AM (last access 6/14/07 10:53 AM)
No users currently signed out.
```

To configure the behavior of sign-in information in the audit log files:

1. In a text editor, open the logging.properties file:
   - **Windows**:
     %NnmDataDir%\shared\nnm\conf\ovjboss\logging.properties
   - **UNIX**:
     /var/opt/OV/shared/nnm/conf/ovjboss/logging.properties
2. *Optional*. To disable sign-in and sign-out logging in the signin.0.0.log file, set SignInOutAuditLog.level to OFF:

   com.hp.ov.ui.util.SignInOutAuditLog.level = OFF

3. *Optional*.To enable sign-in and sign-out logging in the `signin.0.0.log` file, set `SignInOutAuditLog.level` to `CONFIG`:

   ```
   com.hp.ov.ui.util.SignInOutAuditLog.level = CONFIG
   ```

4. *Optional*. To disable sign-in and sign-out logging in the `nnmui.0.0.log` file, set `SIgnInOutAuditLog.useParentHandlers` to `false`:

   ```
   com.hp.ov.ui.util.SIgnInOutAuditLog.useParentHandlers = false
   ```

5. Save and close the `logging.properties` file.
6. Before NNMi implements the change, you must follow the directions in the logging.properties reference page (**Help** → **Documentation Library** → **Reference Pages**, in the *File Formats* category).

Restore the Administrator Role

If you have accidentally configured NNMi so that no one is assigned to the `Administrator` role (preventing anyone from being able to access the Configuration workspaces), use the `system` user account to correct the problem.

Sign into the console using the password that was configured for the `system` account when NNMi was installed.

If you do not remember the password assigned to the `system` account, use the `nnmchangesyspw.ovpl` command to reset the `system` password. See nnmchangesyspw.ovpl for more information.

Note: If you are still unable to sign into the console, verify that the `nms-roles.properties` file is in good working order. See "Restore the System Role" (on page 275) for more information.

Restore the System Role

NNMi provides an `nms-roles.properties` file that stores the `system` role assignment. This file is located in the following directory:

- **Windows**:
  `%NnmInstallDir%\nonOV\jboss\nms\server\nms\conf\props\nms-roles.properties`
- **UNIX**:
  `/opt/OV/nonOV/jboss/nms/server/nms/conf/props/nms-roles.properties`

You should not need to ever modify this file.

To verify the contents of this file:

1. With a text editor, open the `nms-roles.properties` file.
2. Verify that the following required line is present:

   system=admin

3. Save and close the file.

Set Up Command Line Access

NNMi limits access to command line interface commands in one of two ways:

- Requiring user name and password.
- Requiring `system` role.

See **Help** → **Documentation Library** → **Reference Pages** for a list of command line commands. Check the appropriate Reference Page to determine which strategy applies.

Requiring User Name and Password

If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access" (on page 275) for more information.

Requiring System Role

During installation, a special `system` user account is used to access NNMi for the first time. Thereafter, that special account should be used only to use some command line interface commands and to "Restore the Administrator Role" (on page 275).

Command line interface commands that required the `system` user account must be issued from the NNMi management server, and you must have *read* access to the following files on the NNMi management server:

1. `nms-users.properties`
2. `nms-roles.properties`

When you run a command line interface command that requires the `system` user account and you do not specify a user name and password, note the following:

- If you are logged in to the operating system as root user, NNMi is able to access the `system` account information and runs the command line interface command using the `system` credentials.
- If you are logged in to the operating system with a user name other than root and your user name is not configured to access the `system` credentials, you are not able to run the command line interface command.

Note: If you want a user other than `root` to access command line interface commands that require the `system` role, use the nnmsetcmduserpw.ovpl command to reconfigure the User Account name and password assigned to the NNMi `system` role.

Caution: Any user with `read` access to the `nms-users.properties` and `nms-roles.properties` files can potentially change the NNMi `system` user account password. (UNIX: by default, only the `root` user has `read` access to these files. Windows: by default, *any user name that is associated with the Administrators group* has `read` access to these files.)

To configure `read` access to the `nms-users.properties` and `nms-roles.properties` files, follow the operating system instructions for changing file access permissions.

- **Windows:**
  `%NnmInstallDir%\nonOV\jboss\nms\server\nms\conf\props\nms-roles.properties`

  `%NnmInstallDir%\nonOV\jboss\nms\server\nms\conf\props\nms-users.properties`
- **UNIX:**
  `/opt/OV/nonOV/jboss/nms/server/nms/conf/props/nms-roles.properties`

  `/opt/OV/nonOV/jboss/nms/server/nms/conf/props/nms-users.properties`

See "Restore the Administrator Role" (on page 275) and "Restore the System Role" (on page 275) for more information about the `system` user account's role and password.

Communicate to Your Team

After configuring user passwords and roles, communicate the following information to your team:

- "Open the Console" (on page 277)
- "Sign Into the NNMi Console" (on page 278)
- "Sign Out from the Console" (on page 278)

Open the Console

Provide each user with the following information:

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/`

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

When your NNMi management server has more than one fully-qualified domain name, NNMi chooses one during the installation process. There are two ways to find out which domain name NNMi is using in your network environment:

- Click **Help → System Information** and navigate to the **Server** tab. Locate the **Official Fully Qualified Domain Name (FQDN)** attribute value.
- Use the `nnmofficialfqdn.ovpl` command. See the nnmofficialfqdn.ovpl Reference Page.

To determine the current port number configuration, look at the line for `boss.http.port` in the nms-local.properties file (see table for the location of this file). See the nnm.ports Reference Page for more information.

Determine the NNMi Console Port Number

Operating System	Identify Current Port Number
Windows	`%NnmDataDir%\conf\nnm\props\nms-local.properties`
UNIX	`/var/opt/OV/conf/nnm/props/nms-local.properties`

Communicate the following browser requirements for your team to use the NNMi console:

- Use Microsoft Internet Explorer 7.0 or later or Mozilla Firefox 2.0 or later.
- Pop-ups, cookies, and JavaScript must be enabled.
- Each user's screen resolution must be 1024x768 pixels or higher.
- When using Microsoft Internet Explorer as your browser, you can access multiple browser sessions of NNMi. Use a different user name for each browser session.
- When using Mozilla Firefox as your browser, multiple browser sessions all point to the same window.

Note: Users can bookmark the URL for the NNMi console. Use the URL for the NNMi console rather than the NNMi Welcome page. See About the NNMi Console for more information about the NNMi console.

To open the console:

1. Type the following URL (Uniform Resource Locator) into your browser navigation bar:
   `http://<serverName>:<portNumber>/nnm/`
2. Sign in with the following name and password:

   <name you configured>

   <password you configured>

   Note: The sign-in prompt cannot be disabled, but you can include name and password in the URL. See "Launch the Console (showMain)" (on page 957).
3. Click the **Sign In** button. (See "Sign Into the NNMi Console" (on page 278) if you need more information.)
4. The console opens in a new window.
5. *Optional*. Close the NNMi Welcome page.

   Note: If you do not close the NNMi Welcome page or sign out, you can relaunch the console from the NNMi Welcome Page without signing in again.

To refresh the console window:

Click the Refresh icon in the tool bar of any NNMi window.

Sign Into the NNMi Console

After entering the URL for the console, users are prompted for a user name and password.

To sign into the Console:

1. At the **User Name** prompt, enter the assigned user name.
2. At the **Password** prompt, enter the currently assigned password.
3. Click the **Sign In** button.

Each user name is assigned to an NNMi role. The role determines what users can do with the NNMi console. "Determine which NNMi Role to Assign" (on page 263) for more information.

The user name and the associated role appear in the upper right corner of the console as shown in the example below:

Sign Out from the Console

To sign out from the console:

1. Select **File** → **Sign Out**.
2. Click **OK**.

Note the following:

- Sign in is not preserved across user sessions. After signing out, each user must sign in again.
- You must sign out of each browser session that is running NNMi. For example, if you have signed in twice with two different browsers, signing out in one browser does not cause you to lose access in the other browser.
- NNMi automatically signs out any user after four hours of inactivity.

Configure Maps

NNMi enables you to configure the following maps:

- Node Group Map views
- Path View Maps

  Note: The Node Group Overview map provided by NNMi is not configurable.

When configuring Node Group maps, you can do the following:

- Include only the nodes that are important to you.
- Specify which Node Group maps appear in the Topology Maps workspace.
- Specify refresh information.
- View node groups in the context of a relevant background image, such as a map illustrating node locations.
- View node groups in a customized arrangement.

When configuring Node Group map views, you can also specify the role level required to save maps in a customized arrangement. See "Define Node Group Map Settings" (on page 281) for more information.

When configuring Path View maps you specify undiscovered regions of your network by creating a `PathConnections.xml` file that defines the path between the undiscovered nodes. See"Configure a Path View Map" (on page 291) for more information.

You can also specify the maximum number of nodes to display on a map. See "Define Default Map Settings" (on page 279) for more information.

Related Topics

"Node Group Map Settings Form" (on page 281)

Node Group Map View

Position Nodes in a Node Group Map

Define Default Map Settings

Default Map Settings define settings for all of your Node Group Maps.

Note: You can override Default Map Setting using the **Node Group Map Settings** Configuration workspace. See "Configure Basic Settings for a Node Group Map" (on page 282) for more information.

To configure Default Map Settings, do the following:

1. Navigate to the **User Interface Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **User Interface Configuration**.
2. Navigate to the **Default Map Settings** tab.
3. Make your configuration choices (see the Default Map Settings Attributes table).
4. Click **Save and Close** to return to the **User Interface Configuration** form.
5. Click **Save and Close** to save and apply your changes.

Default Map Settings Attributes

Attribute	Description
Map Refresh Interval	Specifies the refresh interval for Status Refresh.
Maximum Number of Displayed Nodes	Use this attribute to change the maximum number of nodes to be displayed on a map. Note the following: • If you change the default value to display a large number of nodes at one time, you might need to re-adjust this number if maps are taking longer than expected to display. • In Layer 2 and Layer 3 Neighbor views, NNMi adds nodes one hop at a time. If NNMi finds a large number of nodes in a single hop, the number of nodes might exceed the maximum number specified.
Maximum Number of Displayed End Points	Use this attribute to change the maximum number of end points to be displayed on a map. **Note**: If you change the default value to display a large number of end points at one time, you might need to re-adjust this number if maps are taking longer than expected to display.
Multiconnection Threshold	Use this attribute to change the number of connections that must exist between two objects before NNMi displays the connections as one thick line on a map (known as a multiconnection). When this number of connections is reached, NNMi displays the connections as one thick line on all maps except Path View maps. **Note**: To display the Interface objects and each connection, double-click the line representing the multiconnection.
Indicate Key Incidents	In the Node Group map, users can click the **Indicate Key Incidents** icon in the map view toolbar to toggle this feature on and off. When enabled ☑, when the this Node Group map opens, NNMi enlarges any objects on a Node Group map that are Source Objects for a **Key Incident**[1]. (For example,

[1]Incidents with both: (1) Severity = other than Normal. (2) Correlation Nature = equal to Root Cause, Service Impact, Stream Correlation, Rate Stream Correlation, Info, or None.

Attribute	Description
	when viewing the Node Group map, NNMi enlarges any node on a Node Group map that has an open root cause incident associated with it.) When disabled ☐, when the this Node Group map opens, NNMi does not indicate the objects on a Node Group map that are Source Objects for a **Key Incident**[1]. You can override this setting for a particular Node Group map, when you "Configure Basic Settings for a Node Group Map" (on page 282). When viewing the Node Group map, you can also enable or disable this feature. See Node Group Maps and Key Incident Views for more information.

Define Node Group Map Settings

Node Group Map settings specify the node group and background image to be used in a Node Group map. Map settings include the following:

- Node group name
- Topology Maps Workspace ordering
- Minimum role for saving edited locations for each node in the map
- Refresh interval
- The maximum number of map nodes
- Node connectivity information
- Node Group connectivity information
- Background image information

Node Group Map views are used for a variety of purposes in NNMi:

- Viewing groups of only the nodes that are important to you.
- Viewing Node Groups in the context of a relevant background image.
- Viewing Node Groups in a customized arrangement.

To define Node Group Map Settings, use the "Node Group Map Settings Form" (on page 281).

To view a Node Group Map, use the **Actions** menu from the NNMi main toolbar from either a Node Group or Node Group Map Settings. See Node Group Map for more information.

To view more information about the Node Group from a Node Group map, use the **File → Open Node Group for Map** option to open the Node Group form for the selected Node Group.

Node Group Map Settings Form

Use the Node Group Map Settings form to configure maps based on currently defined Node Groups. Items you configure include the background image and type of connectivity (for example, Layer 2) to be displayed on the map.

[1]Incidents with both: (1) Severity = other than Normal. (2) Correlation Nature = equal to Root Cause, Service Impact, Stream Correlation, Rate Stream Correlation, Info, or None.

Note: NNMi displays the list of Node Group Map Settings that have default configuration changes. If NNMi does not display a list of Node Group Map Settings, this means that NNMi is using the default settings for each Node Group Map. To change the default settings for a Node Group Map, either reposition the nodes on the map of interest and select **Save Layout** from the Node Group Map toolbar or use the Node Group Map Settings form to create a Node Group Map Settings configuration as described below. See Position Nodes on a Node Group Map for more information about using **Save Layout**.

To configure Node Group Map Settings, do the following:

1. Navigate to the **Node Group Map Settings Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **User Interface Configuration**.
   c. Navigate to the **Node Group Map Settings** tab.
   d. Do one of the following:
      - To create a new configuration, click the New icon, and continue.
      - To edit an existing configuration, click the Open icon in the row representing the Node Group Map Settings definition you want to edit, and continue.
2. Make your configuration choices (see table).
3. Click **Save and Close** to save and apply your changes.

Note: You can also access the Node Group Map Settings form from a Node Group Map using the **File** → **Open Node Group Map Settings** option.

Tasks for Configuring Node Group Map Settings

Task	How
"Configure Basic Settings for a Node Group Map" (on page 282)	Use the Basics Settings pane to configure Node Group, Topology Maps, and Refresh Interval information. **Note**: To apply your Topology Maps Ordering configuration changes, sign out of the NNMi console. After restarting the console, your changes should take affect. Possible configuration changes include reordering a Node Group map view or adding a new Node Group map view to the workspace.
"Configure the Connectivity to be Displayed for a Node Group Map" (on page 286)	Use the Connectivity tab to configure the level of node connectivity to be displayed on the Node Group Map. Use this tab to also specify the Node Group connectivity to be displayed and maximum connections to be included on the Node Group map.
"Configure Background Image Information for a Node Group Map" (on page 287)	Use the Background Image tab to configure information about the Background Image to use on the Node Group map.

Configure Basic Settings for a Node Group Map

The Basic Settings configuration determines general information about the Node Group map.

To establish Basic Settings for a Node Group Map:

1. Navigate to the **Node Group Map Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **User Interface Configuration**.
   c. Select the **Node Group Map Settings** tab.
   d. Do one of the following:
      - To create a Node Group Map Settings definition, click the New icon.
      - To edit a Node Group Map Settings definition, click the Open icon in the row that represents the Node Group Map Settings definition you want to edit.
      - To delete a Node Group Map Settings definition, select a row and click the Delete button
2. Establish the appropriate settings to identify Node Group and Refresh Settings information (see tables).
3. Click **Save and Close** to save and apply your changes.

Basic Attributes

Attribute	Description
Node Group	Specifies which parent node group to display in the Node Group Map view. The contents of the parent node group include any nodes and Child Node Groups associated with it. **Note**: NNMi displays any Child Node Groups of the selected parent Node Group as a hexagon on the map. The **Expand Child in Parent Node Group Map** attribute determines how a Child Node Group appears on the Node Group Map. **Expand Child in Parent Node Group Map** is disabled by default. • If the Child Node Group has the **Expand Child in Parent Node Group Map** attribute *disabled*, the Child Node Group appears as a hexagon on the map as shown below: • If any Child Node Group has the **Expand Child in Parent Node Group Map** attribute *enabled*, NNMi instead recursively displays each of the nodes in that Child Node Group on the map. See Node Group Form: Child Node Groups Tab for more information about configuring Child Node Groups.

Attribute	Description
Topology Maps Ordering	Use this attribute to specify the order in which you want the Node Group map to appear in the **Topology Maps** workspace. **Note**: If you do not want this Node Group map to appear in the **Topology Maps** workspace, leave the value blank. See Views Provided by NNMi for more information about the maps provided in the **Topology Maps** workspace. **Note**: To apply your Topology Maps Ordering configuration changes, sign out of the NNMi console. After restarting the console, your changes should take affect. Possible configuration changes include reordering a Node Group map view or adding a new Node Group map view to the Topology Maps workspace.

Attribute	Description
Minimum Role for Saving Map Layout	Controls the minimum user role required to save the layout of repositioned nodes in a Node Group Map. This value also controls the minimum user role for configuring Node Group Map Settings. **Note**: Only a user with the role of Administrator can set the Minimum Role for Saving Map Layout value. Possible values include: • Administrator • Operator Level 2 • Operator Level 1 The default value is *Administrator.* See "Determine which NNMi Role to Assign" (on page 263)for more information about NNMi roles. **Note**: A user with any role can initially reposition nodes on a Node Group Map view. However, unless your user name is assigned to the required minimum role, you cannot save the new node locations on the map. After being saved, these node positions are seen by any user opening this Node Group Map.
Map Refresh Interval	Specify the Refresh Interval you want to use in days, hours, minutes, and seconds. By default, the Refresh Interval is 30 seconds. This interval is used to set the Refresh Status interval for this map if it is used.
Maximum Number of Displayed Nodes	Specifies the maximum number of nodes to be displayed on the Node Group map. **Note**: This number applies to the total number of nodes within the Node Group, including any Child Node Groups displayed on the map.
Maximum Number of Displayed End Points	Specifies the maximum number of end points to be displayed on a map. **Note**: If maps are taking longer than expected to display, you might need to re-adjust this number.
Multiconnection Threshold	Use this attribute to change the number of connections that must exist between two Node Groups before NNMi displays the connections as one thick line (known as a multiconnection) on a Node Group map. Note the following: • The value you enter overrides the Multiconnection Threshold set using the Default Map Settings. • If this setting is blank, NNMi uses the Multiconnection Threshold value configured in Default Map Settings. • When this number of connections is reached, NNMi displays the connections as one thick line. • To display the Interface objects and each connection, double-click the line representing the multiconnection.

Attribute	Description
Indicate Key Incidents	In the Node Group map, users can click the **Indicate Key Incidents** icon in the map view toolbar to toggle this feature on and off. When enabled ☑, when the this Node Group map opens, NNMi enlarges any objects on a Node Group map that are Source Objects for a **Key Incident**[1]. (For example, when viewing the Node Group map, NNMi enlarges any node on a Node Group map that has an open root cause incident associated with it.) When disabled ☐, when the this Node Group map opens, NNMi does not indicate the objects on a Node Group map that are Source Objects for a **Key Incident**[2].
Include in Visio Export	When enabled ☑, NNMi includes this map when exporting all saved Node Group maps using the **Tools → Visio Export (iSPI NET only)→ Saved Node Group Maps** option. When disabled ☐, NNMi does not include this map when exporting all saved Node Group maps using the **Tools → Visio Export (iSPI NET only) → Saved Node Group Maps** option.

Configure the Connectivity to be Displayed for a Node Group Map

The Connectivity Tab of the Node Group Map Settings form enables you to specify the level of connectivity to be displayed on the Node Group map. You also specify the connections that you want to display.

1. Navigate to the **Connectivity tab** of the **Node Group Map Settings** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **User Interface Configuration**.
    c. Select the **Node Group Map Settings** tab.
    d. Do one of the following:
        - To create a Node Group Map Settings definition, click the New icon.
        - To edit a Node Group Map Settings definition, click the Open icon in the row representing the Node Group Map Settings definition you want to edit.
        - To delete a Node Group Map Settings definition, select a row and click the Delete button
    e. d. Select the **Connectivity** tab.
2. Configure the connectivity information for this Node Group Map Settings definition (see table).
3. Click **Save and Close** to save and apply your changes.

[1]Incidents with both: (1) Severity = other than Normal. (2) Correlation Nature = equal to Root Cause, Service Impact, Stream Correlation, Rate Stream Correlation, Info, or None.
[2]Incidents with both: (1) Severity = other than Normal. (2) Correlation Nature = equal to Root Cause, Service Impact, Stream Correlation, Rate Stream Correlation, Info, or None.

Connectivity Attributes

Attribute	Description
Connectivity Type	Connectivity Type determines the type of connectivity to display between nodes in the Node Group Map view. By default, NNMi displays the Layer 2 connectivity between nodes when displaying a Node Group Map view. Possible values include: • **None** - Choose this if you do not want any connectivity displayed on the map. • **Layer 2** - Uses Layer 2 connectivity when displaying devices in a Node Group Map view. This connectivity is used by default when positioning node locations on a Node Group Map. • **Layer 3** - Uses Layer 3 connectivity when displaying devices on a Node Group Map view. See Position Nodes on a Node Group Map for more information.
Add L2 Subnet Connections	If you specify **Layer 3** or **None** as the Connectivity Type, this option specifies that you want to include any subnet connections determined by Subnet Connections Rules. See "Configure Subnet Connection Rules" (on page 152) for more information.
Add L2 User Connection Edits	If you specify **Layer 3** or **None** as the Connectivity Type, specifies that you want to include any Layer 2 connections added using the NNMi nnmconnedit.ovpl command to add or delete connection data. See "Add or Delete a Layer 2 Connection" (on page 176) for more information.
Interface Group	Use this option, if you want to reduce the connectivity endpoints on the Node Group Map. The Interface Group you select defines the Interface Group to which an interface must belong to be used to connect a Node Group to a Node Group or a Node to a Node Group. NNMi displays Layer 2 endpoints that are interfaces in the group. NNMi displays Layer 3 endpoints that are IP addresses associated with interfaces in the group.
Nodes to Node Group	Select this check box if you want Node to Node Group connectivity to appear on the Node Group map. **Note**: By default, this option is not enabled.
Node Groups to Node Groups	Select this check box if you want Node Group to Node Group connectivity to appear on the Node Group map. **Note**: By default, this option is not enabled.

Configure Background Image Information for a Node Group Map

Use the Background Image tab of the Node Group Map Settings form to configure information about the Background Image to use on the Node Group map.

1. Navigate to the **Background Image** tab of the **Node Group Map Settings** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **User Interface Configuration**.
    c. Select the **Node Group Map Settings** tab.

d. Do one of the following:

- To create a Node Group Map Settings definition, click the New icon.
- To edit a Node Group Map Settings definition, click the Open icon in the row representing the Node Group Map Settings definition you want to edit.
- To delete a Node Group Map Settings definition, select a row and click the Delete button

2. Establish the appropriate settings to identify the Background Image information (see table).
3. Click **Save and Close** to save and apply your changes.

Background Image Attributes

Attribute	Description

Background Image	Enter the URL for the background image you want to use for this Node Group Map. You can use a background image provided by NNMi or add your own.
	
	Note: Click **Background Image** to view the map.
	
	Use a Background Image Provided by NNMi
	
	NNMi provides a set of background images that include maps of many countries. If you want to use one of those images, append the location and file name to the URL at which you access the NNMi console. Use the format: `/nnmbg/<file name>`. For example:
	
	`/nnmbg/colorado.gif`
	
	To see all of the available images provided by NNMi, browse to:
	
	`http://<serverName>:<portNumber>/nnmbg/`
	
	`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))
	
	`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console
	
	Use a Background Image You Provide
	
	You can also provide your own images. See "Background Image Sources in Node Group Maps" (on page 290) for more information about where to load the background images you want to use.
	
	To see a list of all the images added to NNMi, access the following URL:
	
	http://<*serverName*>:<*portNumber*>/nnmdocs/images/
	
	To use an image that has been added to NNMi, use the following URL:
	
	`/nnmdocs/images/<file name>`
	
	For example: `/nnmdocs/images/myimage.gif`
	
	Note the following:
	
	• NNMi accepts images that can be loaded by a Web browser. Common file extensions include: .gif, .png and .jpg.
	
	• Image names are case sensitive. All background image file names provided by NNMi

Attribute	Description
	are lowercase. • Do not use http://<*localhost*> in your URL. This implies the image is on your local machine and is not available from other clients. • If using full URLs, all client machines must be able to resolve the DNS hostname of the server on which the images reside. • When you pan and zoom around the map, the background image moves in relation with the other objects on the map. If the image does not display, see "Troubleshoot URLs When Specifying a Background Image" (on page 291) for more information.

Attribute	Description
Background Image Scale	The Background Image Scale attribute applies to the actual background image dimensions when displayed on a Node Group Map. Enter a floating point number greater than zero (0.0) to indicate the ratio at which you want NNMi to scale the background image. For example, the value 1.0 represents a one-to-one ratio, resulting in a background image displayed at actual size. A value of 2.0 represents a two-to-one ratio, resulting in a background image displayed at twice the actual size. **Note**: The default ratio value is 1.0. (This means no scaling is applied.) Use this default value initially. You can adjust it as needed based on the relative size between the image and nodes. See "Scale Background Images in Node Group Maps" (on page 291) for guidelines for scaling the background images you specify.

Background Image Sources in Node Group Maps

When specifying background images to include in Node Group Maps, NNMi enables you to use images provided by NNMi or images that you provide.

The images that NNMi provides include maps of many countries.

To see the available images provided by NNMi:

Browse to: `http://<serverName>:<portNumber>/nnmbg/`

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

To use your own background images:

Place you user-supplied images in the following directory:

Windows:

`%NnmDataDir%/shared/nnm/www/htdocs/images`

Unix:

`/var/opt/OV/shared/nnm/www/htdocs/images`

NNMi accepts images that can be loaded by a Web browser. Common file extensions include: .gif, .png and .jpg.

To see the available images that have been added to NNMi:

Access the following URL: `http://<serverName>:<portNumber>/nnmdocs/images`

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

See "Node Group Map Settings Form" (on page 281) for more information about how to configure Node Group Maps to use background images.

Scale Background Images in Node Group Maps

Scale a specified background image for a Node Group Map using the Background Image Scale attribute. See "Define Node Group Map Settings" (on page 281) for more information.

When you use the maps provided by NNMi, it is recommended that you initially use the default value of 1.0 for the Background Image Scale.

When you use your own images for map backgrounds and you are selecting a scale value, consider the following:

- NNMi renders its nodes 50 by 50 pixels. This means if your image is 500 pixels wide, there is room for 10 nodes across the image.
- To display the image at normal resolution, enter a scale value of 1.0. (This means no scaling occurs.)
- After the image displays on the map, look at the relationship between the node size and the background to determine whether you need to rescale the background image:
  - If the nodes look too large compared to the background, enlarge the image using a scale value greater than 1.0.
  - If the nodes look too small compared to the background, make the image smaller using a scale value less than 1.0.

Troubleshoot URLs When Specifying a Background Image

This topic contains troubleshooting steps to use if your background image does not display.

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

If you used a relative URL (beginning with a slash (/) in the Background Image attribute value:

1. Copy and paste the URL to a browser.
2. Insert `http://<serverName>:<portNumber>` in front of the slash (/).

   `<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

   `<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

If you used an absolute URL (beginning with http://) in the Background Image attribute value:

Copy and paste the URL to a browser.

Configure a Path View Map

Configuring a Path View map is useful when you have two or more areas of your network which are separated by undiscovered devices, such as service provider nodes. NNMi enables you to configure a

Path View map that traverses undiscovered regions of your network. To configure this kind of Path View map, create a `PathConnections.xml` file that defines the following:

- Required. A Start node for each `<CONNECT>` to be included in the Path View map
- *Optional*. A unique identifier for a `<CONNECT>`
- *Optional*. The outbound interface from each Start node per `<CONNECT>`
- Required. Any number of undiscovered nodes you want to be included in the map between each `<CONNECT>`
- *Optional*. An End node for a `<CONNECT>` to be included in the Path View map
- *Optional*. The inbound interface to each End node per `<CONNECT>` specified.

Each time NNMi determines a node in the Path View, NNMi checks whether the node is specified as a Start node in the `PathConnections.xml` file. If the node is specified as a Start node in `PathConnections.xml`, each `<CONNECT>` configured in `PathConnections.xml` is inserted in the Path View map.

Note: *NNMi Advanced*. NNMi can use RAMS data to determine router paths. When RAMS data is used to determine the router paths, NNMi ignores the `PathConnections.xml` file. See Path View with NNMi Advanced for more information.

Note: *NNMi Advanced*. Path View works only with IPv4 addresses. The NNMi Advanced IPv6 address values are not valid choices for Path View. Any devices in your network that are configured with IPv6 addresses cannot be displayed on Path View maps.

To configure a Path View map:

Using the required format, create a `PathConnections.xml` file in the following location:

Windows:
`%NnmDataDir%/shared/ nnm/conf/PathConnections.xml`

UNIX:
`/var/opt/OV/shared/nnm/conf/PathConnections.xml`

The following table describes each of the file elements and its format requirements. (Also see the sample file)

Note: Each segment of the path that you specify using the `<CONNECT>` element is directional. If you want to view the path between two nodes in both directions, make sure you include the Start and End nodes for each direction. You should also include the inbound interface for the Start node. If you do not limit the possible routers by including the inbound interface for the Start node, Path View might find additional routers in the path.

Elements for the Path View Configuration File

Element Descriptions
`<CONNECTIONS>` Required parent element. The file must include only one `<CONNECTIONS>` element.
`<CONNECT>` Specifies a segment of the path. Each `<CONNECT>` designates a start and stop location for the `<CONNECT>`. The file can include more than one `<CONNECT>` element.
`<ID>` `C1`

Element Descriptions

`</ID>`

Optional. Identifies the connection. NNMi uses the ID value you enter when reporting errors for a `<CONNECT>`.

If you do not provide an ID value for the path between a Start and End node, any error message for the `<CONNECT>` displays `Not Applicable` rather than the unique identification value.

```
<START>
    <IP_OR_DNS>xxx.xx.xxx.x</IP_OR_DNS>
     <OUTBOUND_INTERFACE_IFINDEX>x</OUTBOUND_INTERFACE_IFINDEX>
     <NEXT_HOPS>
          <HOP>xxx.xx.xxx.x</HOP>
          <HOP>xxx.xx.xxx.x</HOP>
     </NEXT_HOPS>
</START>
```

Specifies the node where a segment of the path starts. You provide values for the following elements:

- `<IP_OR_DNS>` provides the name or IPv4 address of a node in your network. See "Configure the Node Name Strategy" (on page 140) for more information about node names.
- *Optional*. `<OUTBOUND_INTERFACE_IFINDEX>` designates which of the Start node's interfaces to use for this segment of the path.
- `<NEXT_HOPS>` designates one or more specific IPv4 addresses or nodes that you want to be included in the path.

```
<END>
  <IP_OR_DNS>xxx.xx.xxx.x</IP_OR_DNS>
  <INBOUND_INTERFACE_IFINDEX>x</INBOUND_INTERFACE_IFINDEX>
</END>
```

Specifies the node where the `<CONNECT>` ends. You provide values for the following elements:

- `<IP_OR_DNS>` provides the name or IPv4 address of a node in your network.
- *Optional*.`<INBOUND_INTERFACE_IFINDEX>` designates which of the End node's interfaces to use for this segment of the path.

`</CONNECT>`

Required. Designates the end of the XML code that defines one segment of your path view.

`</CONNECTIONS>`

Required parent element. Designates the end of the XML code that defines your path view.

Click here to view a sample file:

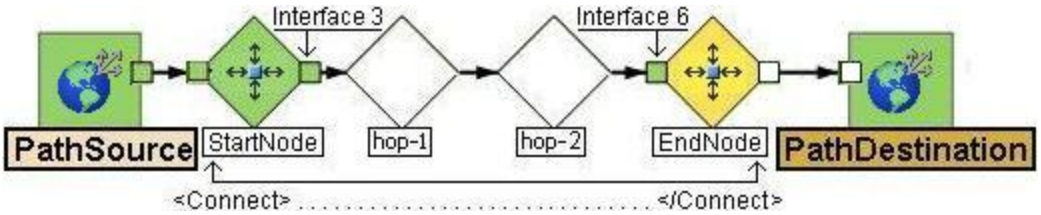
```
<?xml version="1.0" encoding="UTF-8"?>

<CONNECTIONS>
     <CONNECT>
          <ID>
               C1
          </ID>
          <START>
                <IP_OR_DNS>StartNode.xx.xxx.x</IP_OR_DNS>
```

```
            <OUTBOUND_INTERFACE_IFINDEX>3</OUTBOUND_INTERFACE_IFINDEX>
               <NEXT_HOPS>
                    <HOP>hop-1.xxx.xx.xxx</HOP>
                   <HOP>hop-2.xxx.xx.xxx</HOP>
               </NEXT_HOPS>
         </START>
         <END>
               <IP_OR_DNS>EndNode.xxx.xx.xxx</IP_OR_DNS>
               <INBOUND_INTERFACE_IFINDEX>6</INBOUND_INTERFACE_IFINDEX>
         </END>
     </CONNECT>
</CONNECTIONS>
```

When viewing Path View maps that are configured using the `PathConnections.xml` file, note the following:

- If the `<END>` element is not specified, NNMi connects directly to the Destination node to complete the path.
- If the `<END>` element is specified, then the associated `<IP_OR_DNS>` specifies a discovered node as the End node of this segment of your Path View.

Click here to view the sample Path View map generated from the sample file above.

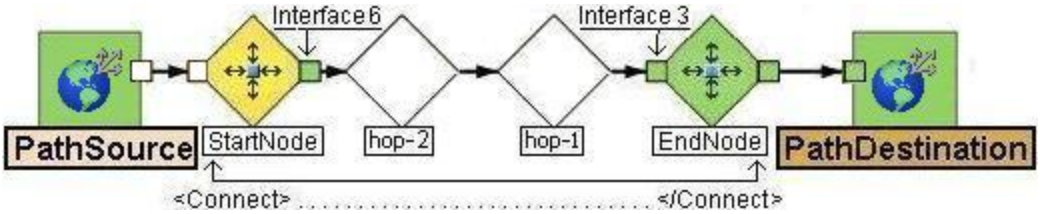


Click here to view a sample file that includes both directions for the sample Path View map above.

Note: In this example, the path is the same in both directions. In many cases, the path might be different in each direction.

```
<?xml version="1.0" encoding="UTF-8"?>

<CONNECTIONS>
   <CONNECT>
         <ID>
             C1
         </ID>
          <START>
               <IP_OR_DNS>StartNode.xx.xxx.x</IP_OR_DNS>
              <OUTBOUND_INTERFACE_IFINDEX>6</OUTBOUND_INTERFACE_IFINDEX>
               <NEXT_HOPS>
                    <HOP>hop-1.xxx.xx.xxx</HOP>
                   <HOP>hop-2.xxx.xx.xxx</HOP>
               </NEXT_HOPS>
          </START>
          <END>
               <IP_OR_DNS>EndNode.xxx.xx.xxx</IP_OR_DNS>
               <INBOUND_INTERFACE_IFINDEX>3</INBOUND_INTERFACE_IFINDEX>
          </END>
     </CONNECT>
</CONNECTIONS>
```

Click here to view the sample Path View map generated from the sample file above after clicking the **Swap Nodes** button.



Configure Default Settings for Line Graph

NNMi enables you to configure default settings for Line Graphs displayed through the Actions menu.

Note: NNMi provides a set of Line Graphs for node and interface objects that are accessible from the Actions menu. As an NNMi administrator you can configure additional Line Graphs using the **Menu Items** tab of the **User Interface Configuration** form. See "Configure SNMP Line Graph Actions" (on page 890) for more information.

To configure default settings for Line Graphs:

1. Navigate to the **Configuration** workspace.
2. Select **User Interface Configuration**.
3. Navigate to the **Default Line Graph Settings** tab.
4. Provide the default settings for all Line Graphs (see the Default Line Graph Settings table).
5. Click **Save and Close** to the **User Interface Configuration** form.
6. Click **Save and Close** to save and apply your changes.

Default Line Graph Settings

Attribute	Description
Default Polling Interval (Seconds)	The Default Polling Interval determines how often the NNMi management server polls for the most recent set of data points to be displayed in a Line Graph. **Note**: This Default Polling Interval does not affect the polling intervals set for the NNMi State Poller. Enter the number of seconds in which NNMi should poll for graph data. If you do not specify an Polling Interval, NNMi determines the best setting for the Polling Interval based on the Maximum Time Range specified. If you do not specify an Polling Interval and you do not specify a Maximum Time Range or if you set the Maximum Time Range to 0 (zero), NNMi determines the best setting for each so the data fits into the Line Graph displayed. When viewing a Line Graph, the user can temporarily change the Polling Interval in a Line Graph. After a graph is re-opened, the Polling Interval returns to this default value. At each Polling Interval, the NNMi management server performs an ad-hoc SNMP query to obtain the most current data.

Attribute	Description
Default Maximum Time Range (Hours)	The maximum time period in hours in which to retain the Line Graph data point sets. When the Maximum Time Range number is reached, NNMi discards the oldest data point sets so that it can display the most recent data for the time range you specify. For example, if you enter 24 hours, when 24 hours has passed, NNMi removes data starting with the initial data point set so that it can display data for the most recent 24-hour interval. Enter a decimal number indicating the maximum number of hours in which to retain the data. If you do not specify a Maximum Time Range or if you specify 0 (zero), NNMi determines the best setting for the Maximum Time Range based on the Polling Interval specified. If you do not specify a Default Maximum Time Range or set the Default Maximum Time Range to 0 (zero), and you do not specify a Default Polling Interval, NNMi determines the best settings for each so the data fits into the Line Graph displayed. You can override this number for an individual graph. See "Configure SNMP Line Graph Actions" (on page 890) for more information.
Default Number of Lines	The Default Number of Lines determines the initial number of lines that are displayed on each Line Graph. **Note**: If more lines than this initial number are available, the user can choose to display additional lines while viewing the graph. You can override this number for an individual graph. See "Configure SNMP Line Graph Actions" (on page 890) for more information.

Configure Menus

As an NNMi administrator, you configure how menu items are nested beneath the Actions menu. See "Create Menu Nesting (in the Actions Menu)" (on page 873) for more information.

Configure Menu Items

The **Menu Items** tab of the **User Interface Configuration** option enables you to make changes or additions to the items available in the Actions menu. For example, you can configure Line Graphs (Graph Action) and additional NNMi Actions (Launch Action) menu items that access in-house tools, Web sites, or a variety of other resources. See "Configure Menu Item Basic Details" (on page 874) for more information.

Configuring Trap Forwarding

NNMi enables you to configure SNMP trap forwarding using the Trap Forward Configuration workspace. This feature is useful when you want to forward traps to a specified destination. For example, you might want to forward certain kinds of traps to one server and forward another set of traps to a different server so they can be managed separately.

When configuring SNMP trap forwarding you perform the following tasks:

- "Configure NNMi Security Settings for SNMPv3 Trap Forwarding" (on page 297)
- "Configure Trap Forwarding Filters" (on page 299)
- "Configure Trap Forwarding Destinations" (on page 301)

Note: See "Manage Incoming SNMP Traps" (on page 427) for information about the criteria NNMi uses to determine when to receive or discard traps.

(*NNMi Advanced - Global Network Management feature*) If you want to forward specific SNMP traps from your NNMi management server (a Regional Manager) to all Global Managers in your Global Network management environment, see "Configure Forward to Global Manager Settings for an SNMP Trap Incident (NNMi Advanced)" (on page 554)

Configure NNMi Security Settings for SNMPv3 Trap Forwarding

Note: If your network environment uses SNMPv2c or SNMPv1 and does not use SNMPv3, skip this task.

If your network environment uses SNMPv3, specify which user-based security model (USM) settings the NNMi management server uses when NNMi acts as an authoritative entity in the following situations:

- Forwarding SNMPv3 traps to other devices in your network environment
- Sending responses to SNMPv3 Inform-Requests

The settings in this form grant permission for NNMi to communicate with the SNMPv3 agent. The SNMPv3 engine identifier and the user-based security settings are required for successful authentication in SNMPv3 protocol.

To configure the NNMi management server as an authoritative entity for SNMPv3:

1. *Prerequisite*: When receiving SNMPv3 data, NNMi must decrypt the data received based on the applicable *SNMP Minimum Security Level* setting in the Communication Configuration workspace. If the applicable SNMPv3 Minimum Security Level = *No Authentication, No Privacy*, NNMi discards all SNMPv3 traps from the device:
   - "Configure Default SNMP and ICMP Protocol Settings" (on page 67)
   - "Communication Region Form" (on page 80)
   - "Specific Node Settings Form (Communication Settings)" (on page 96)
2. Navigate to the **Trap Forward Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Trap Forward Configuration.**
3. Navigate to the **NNMi SNMPv3 Trap Forwarding Security Settings** group.

4. NNMi displays the ID of the engine assigned to the SNMPv3 agent that NNMi uses when forwarding or sending data to other SNMPv3 agents. See the attribute value for Engine Id.
5. Provide the USM information that NNMi uses for authentication and privacy when using SNMPv3 protocol for sending traps or responses to Inform-Requests to other devices in your network environment (see table).
6. Click **Save and Close** to save your changes.

SNMPv3 Engine Assigned to NNMi management server

Attribute	Description
Engine Id	Remote devices must request this SNMPv3 engine ID when sending traps to NNMi. If the SNMPv3 agent sending data to NNMi does not know the correct engine ID, the trap is rejected.

SNMPv3 Settings of the NNMi management server's User-Based Security Model (USM)

Attribute	Description
User Name	The SNMPv3 User Name is the text string used for SNMPv3 requests in your network environment.
Authentication Protocol	The SNMPv3 authentication protocol. Determines whether authentication is required and indicates the type of authentication protocol used. NNMi supports the following protocols: • **HMAC**[1]-**MD5**[2]-96 authentication protocol • **HMAC**[3]-**SHA**[4]-1 authentication protocol
Authentication Passphrase	The SNMPv3 USM authentication passphrase used by the NNMi management server. If required for authentication, provide the appropriate authentication passphrase for the authentication protocol. The length limitations of the authentication passphrase depend on the authentication protocol.
Privacy Protocol	Specify the SNMPv3 USM privacy protocol used by the NNMi management server. This determines whether encryption is required and indicates the type of privacy protocol used. NNMi supports the following privacy protocols: • **DES**[5]-**CBC**[6] Symmetric Encryption Protocol **Note**: To use the following protocol, you must download and install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files. For more information, see the "NNMi Communications" chapter of the HP Network Node Manager i Software Deployment Reference, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals` .

[1]Hash-based Message Authentication Code
[2]Message-Digest algorithm 5
[3]Hash-based Message Authentication Code
[4]Secure Hash Algorithm
[5]Data Encryption Standard
[6]Cipher Block Chaining

Attribute	Description
	• **AES**[1]-128 - Advanced Encryption Standard 128 Protocol **Note**: Leaving this attribute empty means SNMP Minimum Security Level = *No Privacy* for this SNMPv3 configuration.
Privacy Passphrase	Specify the SNMPv3 USM privacy passphrase used by the NNMi management server. If required for privacy, provide the appropriate encryption passphrase for use with the privacy protocol. The length limitations of the privacy passphrase depend on the privacy protocol.

Registration Attributes

Attribute	Description
Last Modified	Date and time the Trap Forward Configuration was last modified.

Configure Trap Forwarding Filters

Pre-requisite: Make sure you have used the NNMi nnmincidentcfg.ovpl command line utility to automatically create or update the incident configurations for the SNMP traps you want to include. See "Load SNMP Trap Incident Configurations" (on page 429) for more information.

Use the Trap Forward Configuration: Trap Forwarding Filters tab to configure a filter expression to specify the SNMP trap Object Identifier (OID) pattern you want to use to determine which SNMP traps NNMi forwards. The traps which pass the filter you specify can then be forwarded to a specified destination using the Trap Forwarding Destinations tab. See "Configure Trap Forwarding Destinations" (on page 301) for more information.

Note: See "Manage Incoming SNMP Traps" (on page 427) for information about the criteria NNMi uses to determine when to receive or discard traps.

To configure Trap Forwarding Filters:

1. Navigate to the **Trap Forward Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Trap Forward Configuration.**
2. Select the **Trap Forwarding Filters** tab.
3. Do one of the following:
   - To create an SNMP Trap Forwarding Filter configuration, click the New icon, and continue.
   - To edit an SNMP Trap Forwarding Filter configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an SNMP Trap Forwarding Filters configuration, click the Delete icon.

[1]Advanced Encryption Standard

4. In the "Trap Forwarding Filter Form" (on page 300) provide the required information.
5. Click **Save and Close** to save your changes and return to the **Trap Forward Configuration** form.

The next time that a trap of this type arrives, NNMi uses the filter you specify to determine whether to forward the trap to a specified destination.

Trap Forwarding Filter Form

Pre-requisite: Make sure you have used the NNMi nnmincidentcfg.ovpl command line utility to automatically create or update the incident configurations for the SNMP traps you want to include. See "Load SNMP Trap Incident Configurations" (on page 429) for more information.

The Trap Forwarding Filters Form enables you to specify the SNMP trap Object Identifier (OID) pattern you want to use to determine which SNMP traps NNMi forwards. The traps which pass the filter you specify can then be forwarded to a specified destination using the Trap Forwarding Destinations tab. See "Configure Trap Forwarding Destinations" (on page 301) for more information.

Note: See "Manage Incoming SNMP Traps" (on page 427) for information about the criteria NNMi uses to determine when to receive or discard traps.

To configure Trap Forwarding Filters:

1. Navigate to the **Trap Forward Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Select **Trap Forward Configuration**.
2. Select the **Trap Forwarding Filters** tab.
3. Do one of the following:
   - To add an SNMP Trap Forwarding Filter configuration, click the New icon, and continue.
   - To edit an SNMP Trap Forwarding Filter configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an SNMP Trap Forwarding Filter configuration, click the Delete icon.
4. Make your configuration choices (see table).
5. Click **Save and Close** to save your changes and return to the **Trap Forward Configuration** form.

SNMP Trap Forwarding Filters Configuration

Attribute	Description
Filter Name	Enter the name you want to use for this SNMP Trap Forwarding Filter configuration. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. No spaces are allowed.
"Filter Form" (on page 300)	Access the Filter Expressions tab to access the Filter form and specify the valid SNMP Object Identifier (OID) pattern to be used for the SNMP trap filter.

Filter Form

Pre-requisite: Make sure you have used the NNMi nnmincidentcfg.ovpl command line utility to automatically create or update the incident configurations for the SNMP traps you want to include. See

"Load SNMP Trap Incident Configurations" (on page 429) for more information.

The Filter Form enables you to specify the SNMP trap Object Identifier (OID) pattern you want to use to filter incoming SNMP traps. The traps which pass the filter you specify can then be forwarded to a specified destination using the Trap Forwarding Destinations tab. See "Configure Trap Forwarding Destinations" (on page 301) for more information.

Note: See "Manage Incoming SNMP Traps" (on page 427) for information about the criteria NNMi uses to determine when to receive or discard traps.

To configure a Trap Forwarding Filter:

1. Navigate to the **Trap Forwarding Filter** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Select **Trap Forward Configuration**.
   c. Select the **Trap Forwarding Filters** tab.
   d. Do one of the following:
      - To create a new configuration, click the New icon.
      - To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
   e. On the form that opens, navigate to the **Filter Expressions** tab.
   f. Locate the **Filter Expressions** table.
   g. Do one of the following:
      - To add a Trap Forwarding Filter, click the New icon.
      - To edit an existing Trap Forwarding Filter, click the Open icon in the row representing the configuration you want to edit.
2. Make your configuration choices (see table).
3. Click **Save and Close** to save your changes and return to the **Trap Forward Configuration** form.

SNMP Trap Forwarding Filter Expression Configuration

Attribute	Description
Trap Object Identifier (OID)	Enter the Trap Object Identifier (OID) pattern you want to use for the SNMP trap filter. Valid values include: • The entire SNMP trap OID value. For example: `.1.3.6.1.6.5.66.7.1225` • The SNMP trap OID value that includes a wildcard as a placeholder for the missing values. For example, to specify only the SNMP trap OID matching prefix: `.1.3.6.1.6.5.66.7.*`

Configure Trap Forwarding Destinations

Pre-requisite: Make sure you have used the NNMi nnmincidentcfg.ovpl command line utility to automatically create or update the incident configurations for the SNMP traps you want to include. See

"Load SNMP Trap Incident Configurations" (on page 429) for more information. You must also create the Trap Forwarding Filters for the SNMP traps you want to forward. See "Configure Trap Forwarding Filters" (on page 299) for more information.

The Trap Forwarding Destinations tab enables you to specify the servers to which you want to forward SNMP traps. Use this tab to also specify the Trap Forwarding Filters to be used for this destination.

(*NNMi Advanced*) If this NNMi management server is a Regional Manager in your environment, see also "Configure Forward to Global Manager Settings for an SNMP Trap Incident (NNMi Advanced)" (on page 554).

Note: See "Manage Incoming SNMP Traps" (on page 427) for information about the criteria NNMi uses to determine when to receive or discard traps.

To configure Trap Forwarding Destinations:

1. Navigate to the **Trap Forward Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Trap Forward Configuration.**
2. Select the **Trap Forwarding Destinations** tab.
3. Do one of the following:
   - To create an SNMP Trap Forwarding Destination configuration, click the New icon, and continue.
   - To edit an SNMP Trap Forwarding Destination configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an SNMP Trap Forwarding Destination configuration, click the Delete icon.
4. In the "Trap Forwarding Destination Form" (on page 302), provide the required information.
5. Do one of the following:
   - To create an SNMP Trap Forwarding Filter configuration, click the New icon, and continue.
   - To edit an SNMP Trap Forwarding Filter configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an SNMP Trap Forwarding Filter configuration, click the Delete icon.
6. In the "Destination Filter Form" (on page 304), provide the required information.
7. Click **Save and Close** to save your changes and return to the **Trap Forward Configuration** form.

The next time a trap that passes the Trap Forwarding Filter arrives, NNMi forwards the trap to the specified Trap Forwarding Destination.

Trap Forwarding Destination Form

Pre-requisite: Make sure you have used the NNMi nnmincidentcfg.ovpl command line utility to automatically create or update the incident configurations for the SNMP traps you want to include. See "Load SNMP Trap Incident Configurations" (on page 429) for more information. You must also create the Trap Forwarding Filters for the SNMP traps you want NNMi to forward. See "Configure Trap Forwarding Filters" (on page 299) for more information.

The Trap Forwarding Destinations form enables you to specify the servers to which you want NNMi to forward SNMP traps.

Note: See "Manage Incoming SNMP Traps" (on page 427) for information about the criteria NNMi uses to determine when to receive or discard traps.

To configure a Trap Forwarding Destination:

1. Navigate to the **Trap Forward Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Select **Trap Forward Configuration**.
2. Select the **Trap Forwarding Destinations** tab.
3. Do one of the following:
   - To add an SNMP Trap Forwarding Destination configuration, click the New icon that precedes the configuration you want to edit, and continue.
   - To edit an SNMP Trap Forwarding Destination configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an SNMP Trap Forwarding Destination configuration, click the Delete icon.
4. Make your configuration choices (see table).
5. Click **Save and Close** to save your changes and return to the **Trap Forward Configuration** form.

SNMP Trap Forwarding Destination Configuration

Attribute	Description
Destination Name	Enter the name you want to use for this SNMP Trap Forwarding Destination configuration. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. No spaces are allowed.
Destination Address	Enter the IP address for the destination server. (*NNMi Advanced*) You can use IPv4 or IPv6 addresses.
Destination Port	Enter the UDP port number for the destination server.
Forwarding Options	• **Default** - NNMi processes the trap prior to forwarding. Click here for more information. NNMi adds two new varbinds to the trap for storing origin address information: ▪ Origin IP Address ▪ Origin IP Address type See "Trap Varbinds Provided by NNMi" (on page 305) for more information. • **SNMPv3 to SNMPv2c Conversion** - NNMi converts an incoming SNMPv3 trap to SNMPv2c. Click here for more information. When converting SNMPv3 traps to SNMPv2c traps, NNMi does the following: ▪ Includes a Context Name varbind - Contains the `contextName` from the original SNMPv3 trap. ▪ Creates a Community Name - The Context Engine ID and SNMPv3 User Name of the original SNMPv3 trap are combined as follows:

Attribute	Description
	`username@contextEngineID`. For example, `ciscoAdmin@8000000b7f3cbec5632b47455e97070c` See "Trap Varbinds Provided by NNMi" (on page 305) for more information. • **Original Trap (UNIX only/IPv4 only)** - NNMi forwards the trap without any changes under certain circumstances. Click here for more information. ▪ Only forwarded from NNMi management servers on UNIX operating systems. ▪ Only forwards traps received-from IPv4 sources and forwarded-to IPv4 destinations.
Specify the Trap Forwarding Filters to Use	Use the Trap Forwarding Filters form to specify the Trap Forwarding Filters configurations to use. These filters determine which traps NNMi forwards to the destination you specify.

Destination Filter Form

Pre-requisite: Make sure you have used the NNMi nnmincidentcfg.ovpl command line utility to automatically create or update the incident configurations for the SNMP traps you want to include. See "Load SNMP Trap Incident Configurations" (on page 429) for more information. You must also create the Trap Forwarding Filters for the SNMP traps you want to forward. See "Configure Trap Forwarding Filters" (on page 299) for more information.

The Trap Forwarding Filter Form enables you to specify the Trap Forwarding Filters that you want to apply for the SNMP traps NNMi forwards to the specified Trap Forwarding Destination.

Note: See "Manage Incoming SNMP Traps" (on page 427) for information about the criteria NNMi uses to determine when to receive or discard traps.

To configure the Trap Forwarding Filters:

1. Navigate to the **Trap Forwarding Filter** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Select **Trap Forward Configuration**.
   c. Select the **Trap Forwarding Destinations** tab.
   d. Do one of the following:
      ◦ To create a new configuration, click the New icon, and continue.
      ◦ To edit an existing configuration, select a row, and click the Open icon in the row representing the configuration you want to edit, and continue.
   e. On the form that opens, navigate to the **FIlter Expressions** tab.
   f. Locate the **Filter Expressions** table.
   g. To create a **Filter Expression**, click the New icon.
2. Make your configuration choices (see table).
3. Click **Save and Close** to save your changes and return to the **Trap Forward Configuration** form.

SNMP Trap Forwarding Filter

Attribute	Description
Filter	Click the Lookup icon, and select Open from the drop-down menu to view information about the selected Filter, if any. Select Quick Find to select the Trap Forwarding Filter you want to use for the current Trap Forwarding Destination.

Trap Varbinds Provided by NNMi

NNMi provides the following varbinds for use when forwarding SNMP traps.

Note: NNMi does not create these varbinds if the Forwarding Options attribute is set to *Original Trap (UNIX only)* when configuring trap forwarding destinations. See "Trap Forwarding Destination Form" (on page 302) for more information.

SNMP Trap Varbinds Provided by NNMi

Name	oid	Type	Description
Origin IP address	.1.3.6.1.4.1.11.2.17.2.19.1.1.3	InetAddress	Contains the IP address (v4 / v6) of the original SNMP notification that generated the trap.
Origin IP Address type	.1.3.6.1.4.1.11.2.17.2.19.1.1.2	InetAddressType	Contains the type of the IP address (v4 / v6) of the Original IP Address varbind. The value "1" indicates IPv4 and "2" indicates IPv6.
Context Name	.1.3.6.1.4.1.11.2.17.2.19.1.1.1	SnmpAdminString	Contains the contextName present in the original SNMPv3 notification. This varbind is present only when NNMi converts an SNMPv3 notification to an SNMPv2c trap. See "Trap Forwarding Destination Form" (on page 302) and "Configure NNMi Security Settings for SNMPv3 Trap Forwarding" (on page 297) for more information.

Configuring Incidents

Incidents are information that NNMi considers important to bring to your attention regarding your network. See "How NNMi Gathers Incidents" (on page 307) for more information.

NNMi provides a set of incident configurations for the following:

- Traps generated from an SNMP agent (SNMPv1, SNMPv2c, or SNMPv3)
- Management incidents that are generated by NNMi
- Events generated by NNM 6.x or 7.x management stations

See "Incident Configurations Provided by NNMi" (on page 312) for more information about the configurations provided.

NNMi provides one centralized location, the incident views, where the management events, SNMP traps, and NNM 6.x or 7.x forwarded events are visible to your team. You control which SNMP traps and NNM 6.x or 7.x events are considered important enough to show up as incidents. You can also configure how incidents that are generated by NNMi are displayed. You and your team can easily monitor the incidents and take appropriate action to preserve the health of your network.

You may choose to modify the incident configurations provided by NNMi or create new incident configurations. To do so, see the following topics:

- "Configuring Trap Forwarding" (on page 297)
- "Configure SNMP Trap Incidents" (on page 436)
- "Configure Management Events" (on page 728)
- "Configure Remote NNM 6.x/7.x Events" (on page 583)
- Using the Incident Configuration form, you can also configure pairwise correlations. See "About Pairwise Configurations" (on page 347) for more information.

Caution: If you make changes to an incident configuration provided by NNMi, those changes are at risk of being overwritten in the future. See Author form for important information.

You can also use the Incident Configuration form to define relationships between multiple incidents by creating deduplication and rate configurations. See "Manage the Number of Incoming Incidents" (on page 341), "Correlate Duplicate Incidents (Deduplication Configuration)" (on page 346), and "Track Incident Frequency (Rate: Time Period and Count)" (on page 347), for more information.

You can use the Incident Configuration form to control how NNMi handles incoming SNMP traps. See "Handle Unresolved Incoming Traps" (on page 432) and "Control which Incoming Traps Are Visible in Incident Views" (on page 431) for more information.

Note: Each time you stop and restart ovjboss, any incidents that have not yet been correlated or persisted are lost. This means that after a restart of ovjboss, an incoming incident might not be correlated as expected. For example, after a restart of ovjboss, a duplicate incident might not be correlated under its original parent incident. Instead, a new parent incident might be generated. See "Stop or Start an NNMi Process" (on page 45) for more information about starting and stopping the ovjboss process.

Manage Incidents Using Incident Configurations

NNMi enables you to control the incidents that are generated and how they are displayed. To help you manage your incidents and incident configurations, you want to understand the following:

- "How NNMi Gathers Incidents" (on page 307)
- "How NNMi Closes Incidents" (on page 310)
- "Incident Configurations Provided by NNMi" (on page 312)

When managing your incidents using Incident Configurations, you can perform the following tasks:

- "Manage the Number of Incoming Incidents" (on page 341)
- "Track Incident Frequency (Rate: Time Period and Count)" (on page 347)
- "Configure an Action for an Incident" (on page 414)
- "Configure Diagnostics for an Incident (NNM iSPI NET)" (on page 421)

How NNMi Gathers Incidents

Incidents are information that NNMi considers important to bring to your attention regarding your network. NNMi gathers incident information from the sources described in the following table.

Incidents Collected by NNMi

Information Source	Description
Causal Engine - Management Events	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 problems and determines the root cause for you, whenever possible, sending incidents to notify you of problems. Any incident generated from a Causal Engine management event has an **Origin** of **NNMi** in your incident views. See Using the Incident Form for more information about incident attributes.
SNMP Traps	Traps are unsolicited SNMP notifications that come from your network devices. The NNMi Causal Engine uses this information as symptoms during its analysis. SNMP traps can also appear as incidents if configured to do so, using the NNMi incident configuration feature. See "Configure SNMP Trap Incidents" (on page 436) for more information.
NNM 6.x and 7.x Events	NNMi can display NNM 6.x and 7.x events that are configured to be forwarded to NNMi.

See "The NNMi Causal Engine and Incidents" (on page 308) for an overview of what the NNMi Causal Engine does with the information collected. See "About the Event Pipeline" (on page 309) for an overview of the event pipeline path each trap or NNMi event takes before NNMi creates an incident. This chronological path guarantees that the data is analyzed in chronological order.

Note: The Causal Engine also sends incident information that it generates through the event pipeline to guarantee the chronological order for determining its root cause incidents.

By default, NNMi includes preconfigured definitions for SNMP traps, NNM 6.x and 7.x events, and the incidents generated by the NNMi Causal Engine. See Incident Views Provided by NNMi for more information.

Related Topics

"Configure SNMP Trap Incidents" (on page 436)

"Configure Management Events" (on page 728)

"Configure Remote NNM 6.x/7.x Events" (on page 583)

"Incident Configurations Provided by NNMi" (on page 312)

"Manage the Number of Incoming Incidents" (on page 341)

The NNMi Causal Engine and Incidents

The Causal Engine extensively evaluates network issues and determines the root cause for you, whenever possible, sending incidents to notify you of problems.

The NNMi Causal Engine defines root cause in terms of symptoms. To do so, it uses a set of rules to define relationships for fault and performance (thresholding) symptoms and root causes. Sources of symptom information include SNMP traps and the monitoring information from the State Poller. See "How NNMi Gathers Incidents" (on page 307) for more information.

The NNMi Causal Engine communicates through incidents in the following ways:

- The Causal Engine generates notifications about problems.
- The Causal Engine closes incidents that are no longer valid (for example, when a "Cold Start" trap is received a short time after a "Node Down" incident was generated because a device was recently rebooted).
- The Causal Engine creates a parent-child relationship between incidents that are all related to one problem (for example, a "Node Down" incident contains a child "Interface Down" incident for each neighboring interface of the node).
- The Causal Engine creates parent-child relationships between incidents that are correlated using the Custom Correlation configuration. See "Configure Custom Correlations" (on page 356) for more information.

The Causal Engine uses the following three stages to help determine and display root cause incidents and their related conclusions.

NNMi Causal Engine Stages

Causal Engine Stages	Description
Condition Listener	Collects symptoms from NNMi processes and services.
Hypothesis engine	Analyzes these symptoms to determine relationships until a root cause is reached.
Blackboard	Based on the information sent by the hypothesis engine, the blackboard updates a device's status and posts any related incidents.

The NNMi Causal Engine analyzes the health of your network and provides the ongoing health status reading for each node, interface, IP address, SNMP agent, and connection. See "The NNMi Causal Engine and Monitoring" (on page 216) for more information.

About the Event Pipeline

Any incident information that appears in your incident views first travels through the event pipeline. The event pipeline guarantees that the incident data is analyzed in chronological order.

Note: Not all information that travels through the pipeline results in an incident.

If at any time an incident does not meet the criteria for a stage in the event pipeline, it is ignored and passed to the next stage in the pipeline. The following table describes the stages contained in the event pipeline.

NNMi Event Pipeline Stages

Event PipelineStages	Description
SNMP Trap and Event Receiver	Accepts all SNMP traps.
pmd Receiver	Accepts NNM events forwarded from remote NNM 6.X and 7.X management stations.
Incident Receiver	Accepts all incident information that comes from the NNMi Causal Engine. See "The NNMi Causal Engine and Incidents" (on page 308) **Note**: The incident information that is received includes any Custom Correlation configurations.
Geo Incident Receiver	Accepts all incident information that comes from Global or Regional Managers.
Type Enforcer	Determines if a configuration exists for this trap, event, or incident. If the incident configuration exists, the type enforcer begins to populate the incident fields according to the configuration. Examples of the incident fields that are populated include **Severity, Origin, Category**, and **Correlation Nature**. If an incident configuration is disabled or does not exist for the incident, NNMi drops the incident.
Resolver	Determines if the incident's Source Node or Source Object (such as interface or card) matches an object in the NNMi database. If available, the resolver populates the incident with the most current Source Node and Source Object attribute values.
Customization	Checks for any of the following incident configurations in the order listed: • Suppression • Enrichment • Dampening
Store Bulk	Collects incidents and stores them.
Notification	Notifies other process and services about a new incident.
Pairwise	Checks for any current pairwise configurations for the incident.
Rate	Checks for any current rate configurations for the incident.
Dedup	Checks for any current deduplication configurations for the incident.
Relate	Performs any additional Causal Engine correlations, including Custom Correlations, and cancels the incident when applicable.

Event PipelineStages	Description
Actions	Performs any automatic actions that the NNMi administrator has configured to be run for one or more incidents. See Using Actions to Perform Tasks for more information.
Rba	(NNM iSPI NET only) The Diagnostics stage. Checks whether Diagnostics should be run on the current incident and submits a execution request to run the Diagnostics report on the device.

How NNMi Closes Incidents

NNMi closes incidents under the following circumstances:

- The incident's configuration is a Pairwise Configuration and both incidents specified in the pair occurred in the order specified. See "About Pairwise Configurations" (on page 347) for more information.
- NNMi determines that the problem that generated the incident is resolved. For example, NNMi closes a Down incident when it generates a Conclusion that indicates the node or device is available for use, has returned to a normal state for a specified threshold, or otherwise no longer needs immediate attention.

  See the tables that follow, beginning with Down Incidents and Associated Conclusions for the list of Conclusions that cause NNMi to set a corresponding Down incident Lifecycle State to Closed.

  The name used to identify each Down incident in the following tables is the Name value used for the Incident Configuration. See"Incident Configurations Provided by NNMi" (on page 312) for the incident configurations that NNMi provides.

An NNMi administrator can also manually change the incident Lifecycle State to Closed. An operator might also be able to change the incident Lifecycle State to Closed if the NNMi administrator chooses to make this Action available.

Note the following:

- The NNMi Causal Engine does not generate Conclusions during initial discovery.
- NNMi only Closes incidents for those objects that have one or more outstanding Conclusions as indicated in the object form's Conclusions tab.

Down Incidents and Conclusion Reasons for Closing Down Incidents

Down Incident	Conclusion Reason for Closing the Down Incident
AddressNotResponding	AddressResponding
BufferOutOfRangeOrMalfunctioning	BufferInRangeAndFunctioning
ConnectionDown	ConnectionUp
ConnectionPartiallyUnresponsive	ConnectionUp
CpuOutOfRangeOrMalfunctioning	CpuInRangeAndFunctioning
CustomPollCritical	CustomPollNormal
CustomPollMajor	CustomPollNormal

Down Incident	Conclusion Reason for Closing the Down Incident
CustomPollMinor	CustomPollNormal
CustomPollWarning	CustomPollNormal
FanOutOfRangeOrMalfunctioning	FanInRangeAndFunctioning
InterfaceDisabled	InterfaceEnabled
InterfaceDown	InterfaceUp
MemoryOutOfRangeOrMalfunctioning	MemoryInRangeAndFunctioning
NodeDown	NodeUp
NodeOrConnectionDown	NodeUp
NonSNMPNodeUnresponsive	NodeUp
PowerSupplyOutOfRangeOrMalfunctioning	PowerSupplyInRangeAndFunctioning
VoltageOutOfRangeOrMalfunctioning	VoltageInRangeAndFunctioning
TemperatureOutOfRangeOrMalfunctioning	TemperatureInRangeAndFunctioning

Down Incidents and Associated Conclusions (*NNMi Advanced*)

Down Incident	Conclusion
AggregatorDegraded	AggregatorUp
AggregatorDown	AggregatorUp
AggregatorLinkDegraded	AggregatorLinkUp
AggregatorLinkDown	AggregatorLinkUp
RrgMultiplePrimary	RrgOnePrimary
RrgMultipleSecondary	RrgOneSecondary
RrgMultipleSecondary	RrgManyExpectedSecondary
RrgNoPrimary	RrgOnePrimary
RrgNoSecondary	RrgOneSecondary
RrgNoSecondary	RrgManyExpectedSecondary

Down Incidents and Associated Conclusions (*HP Network Node Manager iSPI Performance for Metrics Software*)

Down Incident	Conclusion
InterfaceInputDiscardRateHigh	InterfaceInputDiscardRateNominal
InterfaceInputErrorRateHigh	InterfaceInputErrorRateNominal

Down Incident	Conclusion
InterfaceInputUtilizationHigh	InterfaceInputUtilizationNominal
InterfaceInputUtilizationLow	InterfaceInputUtilizationNormal
InterfaceInputUtilizationNone	InterfaceInputUtilizationNominal
InterfaceOutputDiscardRateHigh	InterfaceOutputDiscardRateNominal
InterfaceOutputErrorRateHigh	InterfaceOutputErrorRateNominal
InterfaceOutputUtilizationHigh	InterfaceOutputUtilizationNominal
InterfaceOutputUtilizationLow	InterfaceOutputUtilizationNominal
InterfaceOutputUtilizationNone	InterfaceOutputUtilizationNominal
InterfacePerformanceCritical	InterfacePerformanceClear
InterfacePerformanceWarning	InterfacePerformanceClear

Incident Configurations Provided by NNMi

NNMi provides several incident configurations out-of-the-box. You can review these configurations or modify these configurations to better meet your needs. For example, you might want to customize the message that appears with a particular type of incident, including adding information to the message displayed.

You might also choose to create your own configurations for additional SNMP traps that are important to you.

These out-of-the-box configurations are organized according to the following categories:

"SNMP Trap Incident Configurations Provided by NNMi" (on page 315)

"Management Event Configurations Provided by NNMi" (on page 330)

"Remote NNM 6.x/7.x Event Configurations Provided by NNMi" (on page 328)

"Incident Pair (Pairwise) Configurations Provided by NNM" (on page 348)

Caution: If you make changes to an incident configuration provided by NNMi, those changes are at risk of being overwritten in the future. See Author form for important information.

Custom Incident Attributes Provided by NNMi (for Administrators)

NNMi uses custom incident attributes to attach additional information to incidents.

A subset of CIAs are available for any particular incident. Any relevant CIAs are displayed on the Incident form, in the Custom Attributes tab. There are two categories of possible CIAs:

- SNMP trap varbinds identified by the Abstract Syntax Notation value (ASN.1). Varbinds are defined in MIB files that you can load into NNMi. See "Load SNMP Trap Incident Configurations" (on page 429).
- Custom incident attributes provided by NNMi.

The potential custom incident attributes provided by NNMi are described in the table below.

Custom Incident Attributes Provided by NNMi

Name	Description
cia.address	SNMP agent address.
cia.eventoid	NNM 6.x/7.x object identifier (oid) for the incident.
cia.incidentDurationMs	The time measured in milliseconds between when NNMi detected a problem with one or more network devices to the time the problem was resolved. Use this CIA when you need to track the total time a particular object in the network was down or unavailable. **Note**: This CIA is used only when NNMi's Causal Engine has analyzed and Closed the incident. Any time an incident is closed manually (for example, by the network operator), NNMi does not include cia.incidentDuration.
cia.cardsRemoved	Comma-separated list of removed card names used for formatting the **Card Removed** incident message.
cia.cardsInserted	Comma-separated list of the inserted card names used for formatting the **Card Inserted** incident message.
cia.cardsRemoved	Comma-separated list of removed card names used for formatting the **Card Removed** incident message.
cia.cardsInserted	Comma-separated list of the inserted card names used for formatting the **Card Inserted** incident message.
cia.custompoller.collection	The Name of the associated Custom Poller Collection.
cia.custompoller.lastValue	The last polled value that caused a state change that caused the incident.
cia.custompoller.policy	The Name of the associated Custom Poller Policy.
cia.custompoller.variable.description	The description of the MIB expression being polled.
cia.custompoller.variable.expression	The MIB expression that was collected and the computed value that caused the incident.
cia.custompoller.variable.name	The Name of the MIB expression variable that caused the incident.
cia.custompoller.state	The state of the Custom Polled Instance for this incident.
cia.reasonClosed	The Conclusion information identifying the reason NNMi changed the incident's Lifecycle State to Closed. For example, NNMi might include an Interface Up Conclusion as the reason an Interface Down incident was closed. **Note**: This CIA is used when NNMi's Causal Engine has analyzed and Closed the incident. Software that is integrated with NNMi might also provide values for cia.reasonClosed. Any time an incident is closed manually (for example, by the network operator), NNMi does not include cia.reasonClosed.

Name	Description
cia.remotemgr	Hostname or IP address of the either of the following: • NNM 6.x or 7.x management station that is forwarding the event • (*NNMi Advanced - Global Network Management feature*) NNMi Regional Manager that is forwarding the event
cia.remotetopoid	Topology identifier (topoid) of the NNM 6.x or 7.x event.
cia.snmpoid	SNMP trap object identifier.
cia.timeIncidentDetectedMs	The timestamp in milliseconds when NNMi first detected the problem associated with an incident. **Note**: This CIA is used only when NNMi's Causal Engine has analyzed and Closed the incident. Any time an incident is closed manually (for example, by the network operator), NNMi does not include cia.timeIncidentDetected.
cia.timeIncidentResolvedMs	The time when NNMi determines the problem associated with the incident is resolved. **Note**: This CIA is used only when NNMi's Causal Engine has analyzed and Closed the incident. Any time an incident is closed manually (for example, by the network operator), NNMi does not include cia.timeIncidentResolved.

(*HP Network Node Manager iSPI Performance for Metrics Software*) For network performance monitoring, additional custom incident attributes are provided for your use. Click here for more information.

Custom Incident Attributes Provided for Thresholding (*HP Network Node Manager iSPI Performance for Metrics Software*)

Name	Description
cia.thresholdReason	Configured thresholds have a value of null. Unset thresholds have a value of **No threshold settings defined**. See "Configure Threshold Monitoring for Nodes (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 240) and "Configure Threshold Monitoring for Interfaces (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 228) for the complete list of possible performance threshold results and for information about how to configure performance thresholds.
cia.thresholdParameter	The monitored attribute that is being measured. For example, **Input Utilization**. See "Configure Threshold Monitoring for Nodes (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 240) and "Configure Threshold Monitoring for Interfaces (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 228)for the complete list of possible attributes. This value is selected when configuring performance thresholds.
cia.thresholdLowerBound	The configured value for the low performance threshold. See "Configure Threshold Monitoring for Nodes (HP Network Node

Name	Description
	Manager iSPI Performance for Metrics Software)" (on page 240) and "Configure Threshold Monitoring for Interfaces (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 228)for more information about how to configure performance thresholds.
cia.thresholdUpperBound	The configured value for the high performance threshold. See "Configure Threshold Monitoring for Nodes (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 240) and "Configure Threshold Monitoring for Interfaces (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 228) for more information about how to configure performance thresholds.
cia.thresholdPreviousValue	Results from the previous Performance Polling Interval. For example, the performance threshold results for the Input Error Rate might change from **Nominal** to **High**, based on a change in the thresholdMeasuredValue. See Interface Form for a complete list of possible values.
cia.thresholdCurrentValue	Results from the most recent Performance Polling Interval. For example, **High**. See Interface Form for a complete list of possible values.
cia.thresholdMeasuredValue	The most recent measurement for the performance threshold. The HP Network Node Manager iSPI Performance for Metrics Software software monitors this measurement for threshold violations. This measurement is the average of all measurements taken during the last polling interval (determined by the NNMi State Poller).
cia.thresholdMeasurementTime	The time at which the threshold was reached for a performance threshold. For example, if the threshold for the Input Error Rate is 6.0, and the thresholdMeasuredValue is 6.0, the time at which the thresholdMeasuredValue become equal to 6.0 is stored in this custom incident attribute. The time appears in ISO 8601 format.

These CIAs are used in a variety of ways:

- In SNMP trap configurations. See "Configure SNMP Trap Incidents" (on page 436).
- In remote NNM 6.x/7.x events. See "Configure Remote NNM 6.x/7.x Events" (on page 583).
- In management events. See "Configure Management Events" (on page 728).
- In automatic actions. See "Configure an Action for an Incident" (on page 414).
- In correlation configurations. See "Manage the Number of Incoming Incidents" (on page 341).
- In Launch Action definitions (access through the Actions menu). See "Control the Actions Menu" (on page 872).

SNMP Trap Incident Configurations Provided by NNMi

Caution: If the **Author** value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. See Author form for important information.

NNMi provides the SNMP trap incident configurations described in the following table.

You might also choose to create your own configurations for additional SNMP traps that are important to you.

SNMP Trap Configurations Provided by NNMi

Incident Configuration Name	Description	Deduplication Configuration	Rate Configurat
BGPBackward Transition	Generated when the BGP Finite State Machine moves from a higher numbered state to a lower numbered state.	Name CIA	
BGPEstablished	Generated when the BGP Finite State Machine enters the ESTABLISHED state.	Name CIA	
CempMemBufferNotify	Signifies that a cempMemBufferPeak object has been updated in the buffer pool.	Name	
CiscoChassisAlarmOff	Signifies that the agent entity has detected the chassisTempAlarm, chassisMinorAlarm, or chassisMajorAlarm object in this MIB has transitioned to the off(1) state.	Name CIA	
CiscoChassisAlarmOn	Signifies that the agent entity has detected the chassisTempAlarm, chassisMinorAlarm, or chassisMajorAlarm object in this MIB has transitioned to the on(2) state.	Name CIA	
CiscoChassisChangeNotification	Agent detects any hot-swap component change or changes in the chassis.	Name CIA	
CiscoColdStart	Occurs when a Cisco Agent is powered up.		5 within a time perioc 5 minutes
CiscoDemand NeighborLayer2Change	Sent to the manager whenever the D-channel of an interface changes state.	Name CIA IF index	
CiscoEnvMonFanNotification	Indicates at least one of the fans in the fan array has failed.		
CiscoEnvMonFanStatusChange Notification	Indicates a state change for a device being monitored by ciscoEnvMonFanState.		
CiscoEnvMonRedundantSupplyNotifcation	Indicates the redundant power supply failed.		
CiscoEnvMonSuppStatusChangeNotification	Indicates a change in the state of a device being monitored by ciscoEnvMonSupplyState.		
CiscoEnvMonTemperatureNotification	Indicates the temperature measured at a given testpoint is outside the normal range for the testpoint, For example, it is at the warning, critical, or shutdown stage.		
CiscoEnvMonTempStatusChangeNotification	Indicates a change in the state of a device being monitored by ciscoEnvMonTemperatureState.		

Incident Configuration Name	Description	Deduplication Configuration	Rate Configuration
CiscoEnvMonVoltageNotification	Indicates the voltage measured at a given testpoint is outside the normal range for the testpoint. For example, it is at the warning, critical, or shutdown stage.		
CiscoEnvMonVoltStatusChangeNotification	Indicates a change in the state of a device being monitored by ciscoEnvMonVoltageState.		
CiscoFRUInserted	Indicates a Field Replaceable Unit (FRU) was inserted into the source node.		
CiscoFRURemoved	Indicates a Field Replaceable Unit (FRU) was removed from the source node.		
CiscoLinkDown	Occurs when the Cisco agent detects an interface has gone down.	Name CIA IF index	
CiscoLinkUp	Occurs when the Cisco agent detects an interface has come back up.	Name CIA IF index	
CiscoModuleDown	Signifies that the SNMP Agent has detected that the card has gone down.	Name CIA Module Index	
CiscoModuleStatusChange	Indicates the Operational State of the card has changed.		
CiscoModuleUp	Signifies that the SNMP Agent has detected that the card has come back up.	Name CIA Module Index	
CiscoRFProgressionNotif	Notification sent by the active Card (for example Card Active), whenever its Redundancy Framework (RF) state changes or the RF state of the second card in the Card Redundancy Group changes.		
CiscoRFSwatcNotif	Sent by the newly Active Card (for example Card Active). Indicates that a card state has been switched to a different state.		
CiscoUnrecognizedFRU	Indicates the Field Replaceable Unit (FRU) has a product identification that is not recognized.		
CiscoVlanPortStatusChange	Generated by a device when the value of vlanTrunkPortDynamicStatus object has been changed.	Name CIA	
CiscoWarmStart	Occurs when an Cisco agent is reconfigured.		5 within a time period of 5 minutes
HSRPStateChange	Sent when an HSRP interface transitions to or from an Active or Standby state in a particular HSRP Group.	Name CIA	

Incident Configuration Name	Description	Deduplication Configuration	Rate Configurat
IetfVRRPStateChange	Sent when a standard VRRP interface transitions to or from a Master State in a particular VRRP Group. This trap is used by the standard VRRP protocol. It corresponds to the vrrpTrapNewMaster trap name.	Name CIA	
OSPFIfStateChange	Signifies that there has been a change in the state of a nonvirtual OSPF interface.	Name CIA	
OSPFNbrStateChange	Signifies that there has been a change in the state of a nonvirtual OSPF neighbor.	Name CIA	
OSPFVirtIfStateChange	Signifies that there has been a change in the state of an OSPF virtual interface.	Name CIA	
RMONFallingAlarm	Sent when an RMON device falls below a preconfigured threshold.	Name CIA RMON Alarm Variable	
Rc2kTemperature	Signifies the SNMPv2c entity acting in an agent role, has detected the chassis is overheating.		
RcAggLinkDown	Signifies the operational state of the Multi-Link Trunk (MLT) Aggregator changed from Up to Down.		
RcAggLinkUp	Signifies the operational state of the Multi-Link Trunk (MLT) Aggregator changed from Down to Up.		
RcChasFanDown	Signifies the SNMPv2c entity, acting in an agent role, has detected that the rcChasFanOperStatus object for one of its power supply units is about to transition to the Down state.		
RcChasFanUp	Signifies the SNMPv2c entity, acting in an agent role, has detected that the rcChasFanOperStatus object for one of its power supply units is about to transition to the Up state.		
RcChasPowerSupplyDown	Signifies the rcChasPowerSupplyOperStatus object for one of its power supply units is about to transition to the Down state.		
RcChasPowerSupplyUp	Signifies the SNMPv2c entity, acting in an agent role, has detected that the rcChasPowerSupplyOperStatus object for one of its power supply units is about to transition to the Up state.		
Rcn2kTemperature	Signifies that the SNMPv2c entity, acting in		

Incident Configuration Name	Description	Deduplication Configuration	Rate Configuration
	an agent role, has detected the chassis is overheating.		
RcnAggLinkDown	Signifies the operational state of the Multi-Link Trunk (MLT) Aggregator Link changed from Up to Down.		
RcnAggLinkUp	Signifies the operational state of the Multi-Link Trunk (MLT) Aggregator Interface has changed from Down to Up.		
RcnChasFanDown	Signifies the SNMPv2c entity, acting in an agent role, has detected that the rcChasFanOperStatus object for one of its power supply units is about to transition into the Down state.		
RcnChasFanUp	Signifies the SNMPv2c entity, acting in an agent role, has detected that the rcChasFanOperStatus object for one of its power supply units is about to transition into the Up state.		
RcnPowerSupplyDown	Signifies the SNMPv2c entity, acting in an agent role, has detected that the rcChasPowerSupplyOperStatus object for one of its power supply units is about to transition into the Up state.		
RcnPowerSupplyUp	Signifies the SNMPv2c entity, acting in an agent role, has detected that the rcChasPowerSupplyOperStatus object for one of its power supply units is about to transition into the Up state.		
RcnSmltIsLinkDown	Signifies the rcChasPowerSupplyOperStatus object for one of its power supply units is about to transition into the Down state.		
RcnSmltIsLinkUp	Signifies the rcChasPowerSupplyOperStatus object for one of its power supply units is about to transition into the Up state.		
RcSmltIsLinkDown	Signifies the operational state of the Split Multi-Link Trunk (SMLT) Aggregator Link is transitioning from Up to Down.		
RcSmltIsLinkUp	Signifies the operational state of the Split Multi-Link Trunk (SMLT) Aggregator Link is transitioning from Down to Up.		
RcVrrpStateChange	Sent when a Rapid City (RC) Nortel interface transitions to or from a Master state in a particular VRRP Group. This trap is used by the Rapid City (RC) Nortel propriety VRRP		

Incident Configuration Name	Description	Deduplication Configuration	Rate Configurat
	protocol. It corresponds to the rcVrrpTrapNewMaster trap name.		
RMONFallingAlarm	Sent when an RMON device falls below a preconfigured threshold.		
RMONRiseAlarm	Sent when an RMON device exceeds a preconfigured threshold.		
SNMPColdStart	Signifies that the sending protocol entity is reinitializing itself such that the agent's configuration or the protocol entity implementation may be altered.	Name CIA	5 within a time perioc 5 minutes
SNMPLinkDown	Signifies that the sending protocol entity recognizes a failure in one of the communication links represented in the agent's configuration.	Name CIA IF index	
SNMPLinkUp	Signifies that the sending protocol entity recognizes that one of the communication links represented in the agent's configuration has come up.	Name CIA IF index	
SNMPWarmStart	Signifies that the sending protocol entity is reinitializing itself such that neither the agent configuration nor the protocol entity implementation is altered.	Name CIA	5 within a time perioc 5 minutes
STPNewRoot	Indicates that the sending agent has become the new root of the Spanning Tree.	Name CIA	
STPTopologyChange	Sent by a bridge when any of its configured ports transitions from the Learning state to the Forwarding state, or from the Forwarding state to the Blocking state.	Name CIA	

NNMi Advanced. SNMP Trap Incident Configurations for HP Route Analytics Management Software (RAMS)

Incident Configuration Name	Description	Deduplication Configuration	Rate Configuration
RexAdjStateDown	Signifies the adjacency went down.		
RexAdjStateFlap	Signifies the adjacency's flap count (rexEventCount) in the duration given by rexCountDuration has become greater than or equal to rexEventThreshold. Both adjacency up and adjacency down count as flaps. For example:		

Incident Configuration Name	Description	Deduplication Configuration	Rate Configuration
	An adjacency going down and coming up increments the flap count by two.		
RexAdjStateUp	Signifies the adjacency came up.		
RexASPathChange	Signifies the AS path to a route has changed.		
RexBgpRedundChange	Signifies a change in the number of next hops available for reaching a prefix		
RexBgpVpnReachByCustGain	Signifies the routes in the Customer announced by PE that are up and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of routes in the Customer that are up and not baselined • The percentage of participating routes in the Customer that are up and not baselined		
RexBgpVpnReachByCustLoss	Signifies the routes in the Customer announced by PE that are down and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of routes in the Customer that are down and not baselined • The percentage of participating routes in the Customer that are down and not baselined		

Incident Configuration Name	Description	Deduplication Configuration	Rate Configuration
RexBgpVpnReachByRtGain	Signifies the routes in the Route Target announced by PE that are up and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of routes in the Route Target that are up and not baselined • The percentage of participating routes in the Route Target that are up and not baselined		
RexBgpVpnReachByRtLoss	Signifies the routes in the Route Target announced by PE that are down and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of routes in the Route Target that are down and not baselined • The percentage of participating routes in the Route Target that are down and not baselined		
RexPathChange	Indicates the a path attributes such as metric, number of hops, intermediate hops from a source router to a IP prefix or NSAP address have changed.		
RexPeeringStateDown	Indicates a peering between a router and RAMS has gone down		
RexPeeringStateFlap	Indicates a peering between a router and RAMS has gone down.		
RexPeeringStateUp	Indicates a peering between a router and RAMS has come up.		
RexPrefixDrought	Signifies a particular BGP Peer Rib has decreased significantly from the Baseline Size as a percentage of the baseline		
RexPrefixFlood	Signifies a particular BGP Peer Rib has increased significantly from the Baseline Size as a percentage of the baseline.		

Incident Configuration Name	Description	Deduplication Configuration	Rate Configuration
RexPrefixStateDown	Indicates the prefix(rexDstPrfx,rexDstMask) announced by Router(rexSrcRtrSysID, rexSrcRtrIP, rexSrcRtrName) has gone down.		
RexPrefixStateFlap	Indicates the prefix (rexDstPrfx,rexDstMask) flap count (rexEventCount) in the duration given by rexCountDuration becomes greater than or equal to rexEventThreshold. Both prefix up and prefix down count as flaps. For example: A prefix going down and coming up increments the flap count by two.		
RexPrefixStateUp	Indicates the prefix(rexDstPrfx,rexDstMask) announced by Router(rexSrcRtrSysID, rexSrcRtrIP, rexSrcRtrName) has come up.		
RexRtrConnected	Indicates the first adjacency of a router becomes full duplex. This means the neighbor sends an LSA and the previously isolated router sends an LSA across that adjacency.		
RexRtrIsolated	Signifies a router has become isolated from the rest of the topology as all of its duplex connections it has to other routers which are not overloaded with respect to a particular routing protocol have gone down.		
RexRtrStateFlap	Signifies the router's flap count (rexEventCount) in the duration given by rexCountDuration has become greater than or equal to rexEventThreshold. Both router isolation and router connection count as flaps. For example: A router getting isolated and then connected increments the flap count by two.		
RexTest	This trap is sent for test purposes		

Incident Configuration Name	Description	Deduplication Configuration	Rate Configuration
RexVpnPEParticipationByCustGain	Signifies the Provider Edges (PEs) participating in the Customer that are up and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of PEs that are up and not baselined • The percentage of participating PEs that are up and not baselined		
RexVpnPEParticipationByCustLoss	Signifies the Provider Edges (PEs) participating in the Customer that are down and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of PEs that are down and not baselined • The percentage of participating PEs that are down and not baselined		
RexBgpVpnReachByRtGain	Signifies the routes in the Route Target announced by PE that are up and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of routes in the Route Target that are up and not baselined • The percentage of participating routes in the Route Target that are up and not baselined		
RexVpnPEParticipationByRtLoss	Signifies the PEs participating in the Route Target (RT) that are down and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of PEs that are down and not baselined • The percentage of participating PEs that are down and not baselined		

Incident Configuration Name	Description	Deduplication Configuration	Rate Configuration
RexVpnReachByCustPEGain	Signifies the routes in the Customer announced by PE that are up and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of routes in the Customer that are up and not baselined • The percentage of participating routes in the Customer that are up and not baselined		
RexVpnReachByCustPELoss	Signifies the routes in the Customer announced by PE that are down and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of routes in the Customer that are down and not baselined • The percentage of participating routes in the Customer that are down and not baselined		
RexVpnReachByCustPrefixDown	Signifies that the prefix has become unreachable in Customer.		
RexVpnReachByCustPrefixUp	Signifies that the prefix has become reachable in Customer.		
RexVpnReachByRtPEGain	Signifies the routes in the Route Target announced by PE that are up and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of routes in the Route Target that are up and not baselined • The percentage of participating routes in the Route Target that are up and not baselined		

Incident Configuration Name	Description	Deduplication Configuration	Rate Configuration
RexVpnReachByRtPELoss	Signifies the routes in the Route Target announced by PE that are down and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of routes in the Route Target that are down and not baselined • The percentage of participating routes in the Route Target that are down and not baselined		
RexVpnReachByRtPrefixDown	Signifies the prefix has become unreachable in RT.		
RexVpnReachByRtPrefixUp	Signifies that the prefix has become reachable in RT.		
RexVpnSiteExpectedAnncdPfxLoss	Signifies that there is a decrease in the number of prefixes announced by the Vpn/Site pair.		
RexVpnSiteExpectedRcvdPfxLoss	Signifies that there is a decrease in the number of prefixes received by the Vpn/Site pair.		
RexVpnSitePrefixStateDown	Signifies the prefix(rexDstPrfx,rexDstMask) announced by Router(rexSrcRtrSysID, rexSrcRtrIP, rexSrcRtrName) in VPN(rexVpnName) and site(rexSiteName), has gone down.		
RexVpnSitePrefixStateFlap	Signifies the prefix (rexDstPrfx,rexDstMask) flap count (rexEventCount) in the duration given by rexCountDuration becomes greater than or equal to rexEventThreshold. The prefix is announced by Router (rexSrcRtrSysID, rexSrcRtrIP, rexSrcRtrName) in VPN (rexVpnName) and site (rexSiteName). Both prefix up and prefix down count as flaps. For example: A prefix going down and coming up increments the flap count by two.		

Incident Configuration Name	Description	Deduplication Configuration	Rate Configuration
RexVpnSitePrefixStateUp	Signifies the prefix (rexDstPrfx,rexDstMask) has come up. The prefix is announced by Router (rexSrcRtrSysID, rexSrcRtrIP, rexSrcRtrName) in VPN (rexVpnName) and site (rexSiteName).		
RexVpnSiteUnexpectedAnncdPfxGain	Signifies there is an increase in the number of prefixes announced by the Vpn/Site pair.		
RexVpnSiteUnexpectedRcvdPfxGain	Signifies there is an increase in the number of prefixes received by the Vpn/Site pair.		
TrafficHighLinkUtilization	Indicates the traffic volume has exceeded a specified threshold on a link. Specify the threshold as an absolute number in kilobytes per second or in terms of percentage of link capacity.		
TrafficLinkCoSUtilization	Indicates the traffic volume has exceeded a specified threshold for a CoS queue on a link. Specify the threshold as an absolute number in kilobytes per second or in terms of a percentage of link capacity.		
TrafficLowLinkUtilization	Indicates the traffic volume has fallen below a specified threshold on a link. Specify the threshold as an absolute number in kilobytes per second or in terms of percentage of link capacity.		
TrafficQuantityAlert	A generic trap for all non-link related traffic alerts. Specify the threshold as an absolute number in kilobytes per second or in terms of percentage of link capacity.		

To see or modify these SNMP trap incident configurations:

1. Navigate to the Incident Configuration view.
    a. In the Workspace navigation panel, select the **Configuration** workspace.
    b. Select the **Incident Configuration** view.
2. Select the **SNMP Trap Configuration** tab.

3. Click the Open icon in the row representing the configuration you want to see or edit.
4. When you are finished, click **Save and Close**.

Remote NNM 6.x/7.x Event Configurations Provided by NNMi

Caution: If the **Author** value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. See Author form for important information.

NNMi provides the remote NNM 6.x or 7.x incident configurations described in the following table.

Remote NNMi Out-of-the-Box Incident Configurations

Incident Configuration Name	Description	Deduplication Configuration	Rate Configuration
OvStationNormal	Generated when the status of an NNM 6.x or 7.x management station changes to normal/up.		
OvStationCritial	Generated when the status of an NNM 6.x or 7.x management station changes to down/critical.		
OvNodeWarning	Generated when NNMi detects the status of a node has become up (some or all interfaces in the node are up).		
OVNodeMajor	Generated when NNMi detects the status of a node has become up (some or all interfaces in the node are up).		
OvNodeMarginal	Generated when NNMi detects the status of a node has become up (some or all interfaces in the node are up).		
OvNodeUp	Generated when NNMi detects the status of a node has become up (some or all interfaces in the node are up).		
OvNodeDown	Generated when NNMi detects the status of a node has become down (all interfaces in the node are down).		
OvIfUp	Generated when NNMi detects the status of an interface has come up, normally by responding to an ICMP Echo (ping) request.		
OvIfDown	Generated when NNMi detects the status of an interface has come up, normally by responding to an ICMP (ping) request.		
OvMessage	Generated by a user to display an event message in the event browser.		
OvIfIntermittent	Generated when NNMi detects the		

Incident Configuration Name	Description	Deduplication Configuration	Rate Configuration
	status of an interface has gone down and up multiple times.		
OvApaAddressUp	Generated by the NNMi Causal Engine when it detects that the address is responding to polls.		
OvApaIfUp	Generated by the NNMi Causal Engine when it detects that the interface is responding to polls.		
OvApaNodeUp	Indicates a node's status went from Down to Up.		
OvApaConnUp	Indicates a connection's status went from Down to Up.		
OvApaAggPortUp	Indicates the OperStatus for the logical aggregate port connection is Up.		
OvApaAggPortDown	Indicates the OperStatus for the logical aggregate port connection is Down.		
OvApaAggPortDegraded	Indicates the OperStatus for one of the physical port connections in the aggregate connection is Down		
OvApaAggPortConnUp	Indicates that an aggregate port connection between two nodes is responding to polls and no interfaces are down on either side of the connection.		
OvApaAggPortConnDown	Indicates an aggregate port connection between two nodes is not responding to polls and all interfaces may be down on both sides of the connection.		
OvApaAddressDown	Indicates a node's address status went from Up to Down.		
OvApaIfDown	lindicates a node's interface status went from Up to Down.		
OvApaNodeDown	Indicates a node's status went from Up to Down.		
OvApaConnDown	Indicates a connection's status went from Up to Down.		
OvAPaIfIntermittent	Indicates an interface's status has gone Down and Upmultiple times.		
OvApaAddressIntermittent	Indicates a node's address status has gone Down and Up multiple times.		

Incident Configuration Name	Description	Deduplication Configuration	Rate Configuration
OvApaConnIntermittent	Indicates a network's connection status has gone Down and Up multiple times.		
OvApaNodeIntermittent	Indicates a node's status has gone Down and Up multiple times.		
OvApaNodeSNMPNotResponding	Indicates an SNMP agent is not responding to queries.		
OvApaAggPortNotDegraded	Indicates all of the physical port connections in the aggregate connection are Up.		
OvApaIfRemoved	Indicates an interface has been removed.		
OvApaBoardUp	Indicates a node's board status has gone from Down to Up.		
OvApaBoardDown	Indicates a node's board status has gone from Up to Down.		
OvApaBoardRemoved	Indicates a node's board has been removed.		

To see or modify these Remote NNM 6.x and 7.x trap incident configurations:

1. Navigate to the **Incident Configuration** view.
    a. In the Workspace navigation panel, select the **Configuration** workspace.
    b. Select the **Incident Configuration** view.
2. Select the **Remote NNM 6.x and 7.x Events** tab.
3. Click the Open icon in the row representing the configuration you want to see or edit.
4. When you are finished, click **Save and Close**.

Management Event Configurations Provided by NNMi

Caution: If the **Author** value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. See Author form for important information.

Deduplication is not configured for out-of-the-box management events. See "Correlate Duplicate Incidents (Deduplication Configuration)" (on page 346) for information about how to configure deduplication.

NNMi provides the incident configurations for management events. Click here for more information.

To see or modify these management event incident configurations:

1. Navigate to the **Incident Configuration** view.
    a. In the workspace navigation panel, select the **Configuration** workspace.
    b. Select the **Incident Configuration** view.

2. Select the **Management Event Configuration** tab.
3. Select the configuration you want to see or modify.
4. Click the Open to see or change the configuration.
5. When you are finished, click **Save and Close**.

Management Event Configurations Provided by NNMi

Incident Configuration Name	Description	Rate Configuration
AddressNotResponding	Indicate an address is not responding to ICMP. Reasons an address might not respond include: • Its node is down • A device, such as a router, has been mis-configured so that some addresses cannot be reached	5 events in 5 minutes
AggregatorDegraded	Indicates one or more (but not all) physical interfaces that are part of the Aggregator Interface are not operational.	
AggregatorDown	Indicates the operational status of the Aggregator Interface is down (if monitored), or all of the corresponding physical interfaces are Down.	
AggregatorLinkDegraded	Indicates any Aggregator Interface is operationally down on either node, when there is a connection between two Aggregator Interfaces.	
AggregatorLinkDown	Indicates the Aggregator Interface on either side of an Aggregator Link connection is down.	
BufferOutOfRangeOrMalfunctioning	Indicates the buffer pool is exhausted or cannot meet demand.	
CardDisabled	Indicates that the card has been disabled by the device administrator.	
CardDown	Indicates the card is not responding to polls.	
CardRemoved	Indicates the card was removed from a device.	
CardInserted	Indicates a card was inserted into a device.	
CardUndeterminedState	Indicates the card reported a non-normal state for some unspecified reason.	
ConnectionDown	Indicate that both (or all) ends of a connection are not responding to SNMP queries.	5 events in 5 minutes

Incident Configuration Name	Description	Rate Configuration
ConnectionPartiallyUnresponsive	Indicates that a connection is partially unresponsive. Reasons for this include that an undiscovered device in the connection is down.	
CpuOutOfRangeOrMalfunctioning	Indicates any of 5 second, 1 minute, or 5 minute utilization averages is too high.	
CrgFailover	Indicates the role of a primary card (for example, Card Active) has moved from one card to the other in a Card Redundancy Group. The Card Redundancy Group is routing packets properly.	3 events in 30 minutes
CrgMultiplePrimary	Indicates NNMI has identified multiple primary cards (for example, Card Active) in the Card Redundancy Group. This typically indicates the communication between the cards in the group is malfunctioning.	
CrgNoPrimary	Indicates NNMi is unable to identify a primary card (for example, Card Active) in the Card Redundancy Group. This typically indicates one of the following: • One card, or both cards, are down • NNMi has identified only secondary cards (for example Standby cards) in the group • Communication between cards in the group is malfunctioning.	
CrgNoSecondary	Indicates NNMi cannot identify a secondard card (for example Card Standby) in the Card Redundancy Group. This typically indicates the following: • One of the two cards in the group is down. • NNMi has identified the other card as primary (for example, Card Active). • The Card Redundancy Group is functioning properly	
CustomPollCritical	Indicates that a Polling Instance associated with the Custom Poller Collection is in a Critical State.	
CustomPollMajor	Indicates that a Polling Instance associated with the Custom Poller Collection is in a Major State.	

Incident Configuration Name	Description	Rate Configuration
CustomPollMinor	Indicates that a Polling Instance associated with the Custom Poller Collection is in a Minor State.	
CustomPollWarning	Indicates that a Polling Instance associated with the Custom Poller Collection is in a Warning State.	
DuplicateCorrelation	Provided as a template for configuring deduplication for an incident to specify which attribute values NNMi must match to verify that an incident is a duplicate.	
FanOutOfRangeOrMalfunctioning	Indicates the specified fan is not operating correctly.	
ForwardIncidentRateExceeded	(*NNMi Advanced*) Indicates that the volume of messages entering a Regional Manager's Global Network Management message queue has exceeded the configured rate limits.	
ImportantNodeorConnectionDown	Indicates a node is not responding to an ICMP or SNMP query. It also indicates that only one neighbor is down so that the NNMi Causal Engine cannot determine whether the node or the connection is down.	
ImportantNodeUnmanageable	Indicates a nodes is not responding to an SNMP query.	
InterfaceDisabled	Indicates the interface has been explicitly disabled by the device administrator.	5 events in 5 minutes
InterfaceDown	Indicates that the interface is not responding to polls.	5 events in 5 minutes
IslandGroupDown	Indicates all nodes in a group of Layer 2 connected nodes do not respond to monitoring polls (for example, ICMP or SNMP). These groups are automatically discovered and contain all of the nodes that can be connected through NNMi topology. Typically, these are groups on one side of a WAN (wide area network) connection.	
LicenseExpired	Indicates that the expiration date has passed for an instant-on or temporary NNMi license key. See "Extend a Licensed Capacity" (on page 1020).	

Incident Configuration Name	Description	Rate Configuration
LicenseMismatch	Indicates that the licensed capacity for NNMi does not match the licensed capacity for one of the following products in your network environment: • An NNMi Integration Enablement • HP Network Node Manager iSPI Network Engineering Toolset Software (NNM iSPI NET) • HP Network Node Manager iSPI Performance for Metrics Software • HP Network Node Manager iSPI Performance for Traffic Software **Note**: The licensed capacity count is cumulative for each licensed product (across all installed license keys for that licensed product). See "Extend a Licensed Capacity" (on page 1020).	
LicenseNodeCountExceeded	Indicates that the number of discovered nodes exceeds the licensed capacity for managed node count. See "Extend a Licensed Capacity" (on page 1020).	
MemoryOutOfRangeOrMalfunctioning	Indicates the Source Node's memory pool is exhausted or cannot meet the demand for use.	
MemoryQueueIncidentRateExceeded	(*NNMi Advanced*) Indicates the rate at which NNMi forwards incidents to the Global Manager has exceeded the maximum allowed. NNMi no longer forwards incidents generated from SNMP traps and NNM 6.x/7.x Remote Events.	
MessageQueueSizeExceeded	Indicates one of the queues connecting the stages for the Event Pipeline is above the configured limits. NNMi determines queue size limits based on memory size.	
ModifiedConnectionDown	Indicates a connection has been disconnected, moved, or both and is not responding to SNMP queries.	
NnmClusterFailover	Indicates the NNMi cluster detected a failure of the active server. NNMi services were started on the standby server.	
NnmClusterLostStandby	Indicates the NNMi cluster active server lost its communication to the standby server.	

Incident Configuration Name	Description	Rate Configuration
NnmClusterStartUp	Indicates the NNMi cluster was started in a state where no active server was already present. Therefore the server was started in the active state.	
NnmClusterTransfer	Indicates the system administrator moved the active state from one server to another. The NNMi services will then start on the new active server.	
NodeDown	Indicates that the NNMi Causal Engine has determined the node is down based on the following analysis: 100% of the addresses assigned to this node are unreachable The SNMP agent installed on this machine is not responding NNMi is communicating with at least two of the neighboring devices. And at least two neighboring devices report problems with connectivity to this node.	5 events in 5 minutes
NodeOrConnectionDown	Indicate a node is not responding to an ICMP or SNMP query. It also indicates that only one neighbor is down so that the NNMi Causal Engine cannot determine whether the node or the connection is down.	
NonSNMPNodeUnresponsive	Indicates that a Non-SNMP node is unresponsive. Reasons for this include: 1) The node is down, or 2) An undiscovered device in the path between the node and the NNMi management server is down.	
PowerSupplyOutOfRangeOrMalfunctioning	Indicates a power supply for the Source Node is not operating correctly.	
RateCorrelation	Provided as a template to measure the number of incoming incidents within a defined time period.	

Incident Configuration Name	Description	Rate Configuration
RrgDegraded	**Note**: This incident occurs only in Router Redundancy Groups using the HSRP protocol and larger than two members. Indicates the following: • The Router Redundancy Group has a primary and secondary device. • The remaining devices in the group are not in an expected protocol-specific state. For example, in HSRP other devices may be expected to be in the "Listen" state. Typically, the protocol-specific communication between routers is malfunctioning. However, the group is routing packets properly.	
RrgFailover	Indicates a primary role (for example, HSRP Active or VRRP Master) moved from one device to another in a Router Redundancy Group. Reasons for this incident include one or more of the following: • A router or interface in the Router Redundancy Group has gone down. • A tracked object (interface or IP address) in the Router Redundancy Group has gone down. The group is routing packets properly.	
RrgMultiplePrimary	Indicates that multiple primary devices (for example, HSRP Active or VRRP Master) are identified in a Router Redundancy Group. Typically, the protocol-specific communication between routers in the group is malfunctioning.	
RrgMultipleSecondary	Indicates that more than one secondary device (HSRP Standby) is identified in a Router Redundancy Group. **Note**: This incident applies only to Router Redundancy Groups using the HSRP protocol. VRRP allows more than one secondary role (VRRP Standby State) . Typically, the protocol-specific communication between routers in the group is malfunctioning.	

Incident Configuration Name	Description	Rate Configuration
RrgNoPrimary	Indicates that no primary device (for example, HSRP Active or VRRP Master) is identified in a Router Redundancy group. This incident typically indicates one of the following: • Too many routers are down. • Protocol-specific communication between routers in the group is malfunctioning.	
RrgNoSecondary	Indicates that no secondary device (for example, HSRP Standby or VRRP Backup) is identified in a Router Redundancy Group. This incident typically indicates the following: • Protocol-specific communication between routers in the group is malfunctioning. • The group is routing packets properly because a single primary device has been identified.	
RrgSecondaryChanged	Indicates that a secondary role (for example, HSRP Standby or VRRP Backup) moved from one device to another in a Router Redundancy Group. This incident typically indicates the following: • An router or interface in the Router Redundancy Group has gone down. • A tracked object (interface or address) has gone down. • The group is routing packets properly.	
SNMPTrapLimitCritical	Indicates the number of SNMP traps persisted in the NNMi database is approaching the maximum allowed limit. After the maximum allowed limit is reached, NNM no longer accepts SNMP traps until the number of SNMP traps within the database is reduced using the nnmtrimincidents.ovpl command.	
SNMPTrapLimitMajor	Indicates the number of SNMP traps persisted in the NNMi database has reached or exceeded 95% of the maximum limit. After the maximum limit is reached, NNMi only accepts traps required for Causal Engine analysis until the number of SNMP traps within the database has been reduced using	

Incident Configuration Name	Description	Rate Configuration
	the nnmtrimincidents.ovpl command.	
SNMPTrapLimitWarning	Indicates the number of SNMP traps persisted in the NNMi database has reached or exceeded 90% of the maximum limit. After the maximum limit is reached, NNMi no longer accepts SNMP traps until the number of SNMP traps within the database is reduced using the nnmtrimincidents.ovpl command.	
TemperatureOutOfRangeOrMalfunctioning	Indicates the specified temperature sensor on the Source Node is too hot or too cold.	
TrapStorm	Indicates a trap storm has occurred.	
VoltageOutOfRangeOrMalfunctioning	Indicates the specified voltage on one of the Source Node's power supplies is out of range.	

(*HP Network Node Manager iSPI Performance for Metrics Software*) For network performance monitoring, the HP Network Node Manager iSPI Performance for Metrics Software software provides additional management event configurations. Click here for more information.

Each of these configurations has a Category value of **Performance,** a Family value of **Interface**, and a Nature of **Root Cause**.

Additional Management Event Configurations (*HP Network Node Manager iSPI Performance for Metrics Software*)

Incident Configuration Name	Description	Rate Configuration
InterfaceInputDiscardRateHigh	Indicates a high input discard rate on the interface. This rate is based on the reported change in the number of input packets on the interface and the discarded packet count.	
IntefaceInputErrorRateHigh	Indicates a high input error rate on the interface. This rate is based on the reported change in the number of input packets on the interface and the packet error count.	
InterfaceInputUtilizationHigh	Indicates a high input utilization on the interface. This percentage is based on the interface speed and the reported change in the number of input bytes on the interface.	
InterfaceInputUtilizationLow	Indicates a low input utilization on the interface. This percentage is based on the interface speed and the reported change in the number of input bytes on the interface.	
InterfaceInputUtilizationNone	Indicates there is no input utilization on the interface. This value is based on the interface speed and the	

Incident Configuration Name	Description	Rate Configuration
	reported change in the number of input bytes on the interface.	
InterfaceOutputDiscardRateHigh	Indicates a high output discard rate on the interface. This rate is based on the reported change in the number of input packets on the interface and the discarded packet count.	
InterfaceOutputErrorRateHigh	Indicates a high output error rate on the interface. This rate is based on the reported change in the number of output packets on the interface and the packet error count.	
InterfaceOutputUtilizationHigh	Indicates a high output utilization on the interface. This percentage is based on the interface speed and the reported change in the number of output bytes on the interface.	
InterfaceOutputUtilizationLow	Indicates a low output utilization on the interface. This percentage is based on the interface speed and the reported change in the number of output bytes on the interface.	
InterfaceOutputUtilizationNone	Indicates there is no output utilization on the interface. This value is based on the interface speed and the reported change in the number of output bytes on the interface.	
InterfacePerformanceCritical	Indicates that the interface performance has reached a Critical severity.	
InterfacePerformanceWarning	Indicates that the interface performance has reached a Warning severity.	

Incident Pair (Pairwise) Configurations Provided by NNM

NNM provides the pairwise configurations described in the following table.

Pairwise Configurations Provided by NNM

Name	Description
CiscoLinkDownUpPair	Cancels a CiscoLinkDown incident with a CiscoLinkUp incident on the same interface for the same SNMP agent address. This configuration is used for known Cisco devices.
CiscoModuleDownUpPair	Not yet implemented.
OvApaAddressDownUpPair	Cancels an NNM 6.x or 7.x Node Down event with a NNM 6.x or 7.x Node Up event from the same SNMP agent address.
OvApaAggPortConnDownUpPair	Cancels an NNM 6.x or 7.x APA Aggregate Port Connection Down event with an NNM 6.x or 7.x APA Aggregate Port Connection Up event.

Name	Description
OvApaAggPortDegradeNotDegradePair	Cancels an NNM 6.x or 7.x APA Aggregate Port Degraded event with an NNM 6.x or 7.x APA Aggregate Port Not Degraded event on the same interface for the same SNMP agent address.
OvApaAggPortDownUpPair	Cancels of an NNM 6.x or 7.x APA Aggregate Port Down event with an NNM 6.x or 7.x APA Aggregate Port Up event on the same interface for the same SNMP agent address.
OvApaBoardDownUpPair	Cancels an NNM 6.x or 7.x APA Board Down event with an NNM 6.x or 7.x APA Board Up event from the same SNMP agent address.
OvApaConnDownUpPair	Cancels an NNM 6.x or 7. x APA Address Down event with an NNM 6.x or 7.x APA Address Up event on the same address for the same SNMP agent address.
OvApaIfDownUpPair	Cancels an NNM 6.x or 7.x APA If Down event with an NNM 6.x or 7.x APA If Up event on the same interface for the same SNMP agent address.
OvApaNodeDownUpPair	Cancels an NNM 6.x or 7.x APA Node Down event with an NNM 6.x or 7.x APA Node Up event from the same SNMP agent address.
OvIfDownUpPair	Cancels an NNM 6.x or 7.x If Down event with an NNM 6.x or 7.x If Up event on the same interface for the same SNMP agent address.
OvNodeDownUpPair	Cancels an NNM 6.x or 7.x Node Down event with an NNM 6.x or 7.x Node Up event from the same SNMP agent address.
RcAggLinkDownUpPair	Cancels an RcAggLinkDown incident with an RcAggLinkUp incident from the same Multi-Link Trunk (MLT) Aggregator ID (from the MIB) and SNMP agent address.
RcChasFanDownUpPair	Cancels an RcChasFanDown incident with an RcChasFanUp incident from the same fan ID (from the MIB) and SNMP agent address.
RcChasPowerSupplyDownUpPair	Cancels an RcChasPowerSupplyDown incident with an RcChasPowerSupplyUp incident from the same power supply ID (from the MIB) and SNMP agent address.
RcSmltIsLinkDownUpPair	Cancels an RcSmltIstLinkDown incident with an RcSmltIstLinkUp incident from the same SNMP agent address.
RcnAggLinkDownUpPair	Cancels an RcnAggLinkDown incident with an RcnAggLinkUp incident from the same Multi-Link Trunk (MLT) Aggregator Link ID (from the MIB) and SNMP agent address.
RcnChasFanDownUpPair	Cancels an RcnChasFanDown incident with an RcnChasFanUp incident from the same fan ID (from the MIB) and SNMP agent address.

Name	Description
RcnChasPowerSupplyDownUpPair	Cancels an RcnChasPowerSupplyDown incident with an RcnChasPowerSupplyUp incident from the same power supply ID (from the MIB) and SNMP agent address.
RcnSmltlsLinkDownUpPair	Cancels an RcnSmltlstLinkDown incident with an RcnSmltlstLinkUp incident from the same SNMP agent address.
SnmpLinkDownUpPair	Cancels an SNMPLinkDown incident with an SNMPLinkUp incident on the same interface for the same SNMP agent address.

To see or modify these incident pair configurations:

1. Navigate to the **Incident Configuration** view.
   a. In the Workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Incident Configuration** view.
2. Select the **Pairwise Configuration** tab.
3. Select the configuration you want to see or modify, and click Open to see or change the configuration.

   In the **Pairwise Configuration** form, click **Help→ Using the Pairwise Configuration form** for more information.
4. When you are finished, click **Save and Close** to save your changes and return to the Incident Configuration form.

Manage the Number of Incoming Incidents

NNMi's Causal Engine reduces the number of incidents by extensively evaluating problems and determining the root cause for you, whenever possible.

To help simplify the diagnosis of network faults, you can configure NNMi to manage the number of incidents that are displayed. To do so, use any of the following methods:

- **Disable the Incident configuration**. In the **Basics** group of the Incident Configuration form, verify that **Enabled** is cleared for each configuration for which you do not want NNMi to generate an Incident.
- **Use NNMi's Management Mode feature to set the Management Mode of the network object to Not Managed or Out of Service**. NNMi discards any incoming traps if the Source Node or Source Object is **Unmanaged**[1]. See "Stop or Start Managing a Node, Interface, Card, Address, or Node Component" (on page 250) for more information.
- **Use the Monitoring Configuration to specify that you do not want NNMi to monitor the network object.** NNMi discards any incoming traps if the Source Node or Source Object is not monitored. See "Configure Monitoring Behavior" (on page 216) for more information.
- **Identify additional criteria for or relationships between incoming incidents**. When these criteria or

[1]Indicates the Management Mode is "Not Managed" or "Out of Service".

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.

These strategies can dramatically reduce the number of incidents and improve the value of the incidents displayed. For example, instead of displaying an entire incident storm typically generated by equipment and link failures, use the deduplication configuration to specify only the most meaningful incidents, and correlate the rest as children. Then it is faster and easier to identify the network problem. See "Establish Criteria or Relationships for Incoming Incidents" (on page 342) for more information.

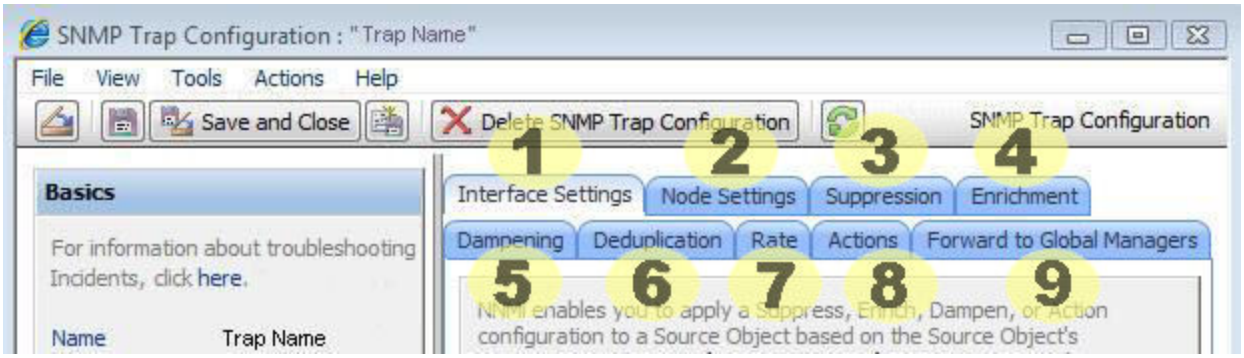
Related Topics

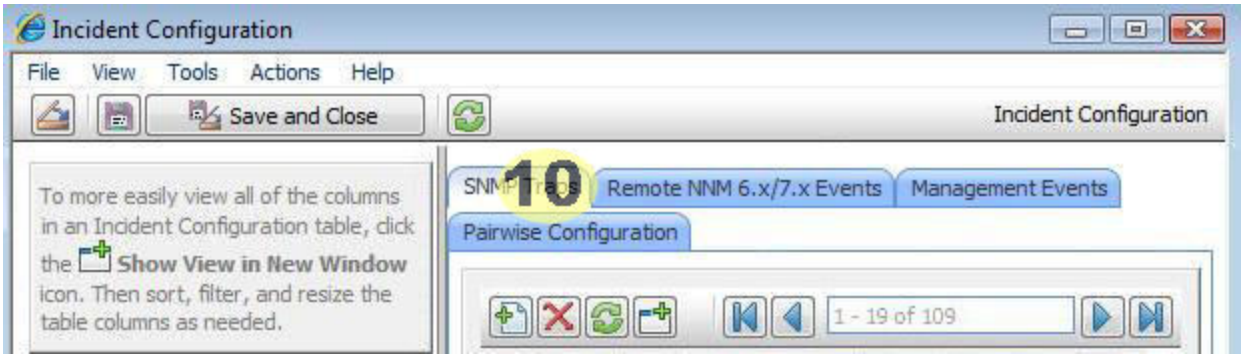
"Configure Management Events" (on page 728)

"Configure SNMP Trap Incidents" (on page 436)

Establish Criteria or Relationships for Incoming Incidents

Using NNMi, you can establish the criteria or relationships for the incoming incidents using any of the incident configurations shown in the following diagram and explained in the table below. You can choose to use them as is, edit them, or create your own configurations. You can also create your own Custom Correlations. See "Configure Custom Correlations" (on page 356) for more information about Custom Correlations.





Overview of Incident Configuration Options

	Configuration Option	When to Use	Example
1	Interface Settings	Select this tab to specify that you want to configure Suppression, Enrichment, Dampening, and Actions for a specified Interface Group.	Change the Severity and Message of an incident

	Configuration Option	When to Use	Example
			configuration for a specified Interface Group. Dampen an Interface Down incident only for the interfaces in a specified Interface Group that you know will be intermittently unavailable.
2	Node Settings	Select this tab to specify that you want to configure Suppression, Enrichment, Dampening, Actions, and Diagnostic Selections for a specified Node Group.	Change the Severity and Message of an incident configuration for the nodes in a specified Node Group.
3	Suppression	Select this tab when you want to discard an incident before it appears in an incident view.	Discard an incident if it is in response to a particular status change notification trap.
4	Enrichment	Select this tab when you want to fine tune any of the following for a selected incident configuration: • Category • Family • Severity • Priority • Correlation Nature • Message • Assigned To • Add a node or interface Custom Attribute to an incident	Change the Severity and Message of an incident configuration.
5	Dampening	Select this tab to delay (dampen) the following for an incident configuration: • Appearance within Incident views in the NNMi Console • Execution of Incident Actions • Execution of Diagnostics (NNM iSPI NET)	Lengthen the Dampen Interval for the Interface Down incident Configuration provided by NNMi. Disable Dampening for the Interface Down Incident Configuration provided by NNMi.

	Configuration Option	When to Use	Example
6	Deduplication	Select this tab to correlate incidents that are identified as duplicates based on one or more Custom Incident Attribute (CIA) or SNMP trap varbind values. To help your operators understand the magnitude or significance of the problem, NNMi tracks the number of duplicates generated. This value is captured as the Duplicate Count attribute. It is incremented on the Duplicate Correlation incident. Its Correlation Nature attribute value is **Dedup Correlation**. NNMi also records the following information related to duplicate incidents: **First Occurrence Time**: Indicates the timestamp of the first occurrence of a duplicate incident. **Last Occurrence Time**: Indicates the timestamp of the latest notification for a set of duplicate incidents. **Count**: Specifies the number of duplicate incidents for the current configuration that NNMi stores at one time. For example, if the Count is 10, after NNMi receives 10 duplicate incidents, NNMi deletes the first (oldest) duplicate incident and keeps the eleventh. (NNMi stores ten maximum.) **Note**: A Duplicate Correlation incident inherits the Dampening settings of its Correlated Children. If the Correlated Children are Closed while Dampened, and therefore deleted, NNMi retains the parent Duplicate Correlation incident. See "Dampening Incident Configurations" (on page 356) for more information about Dampening an Incident Configuration.	Identify any CiscoLinkDown incidents as duplicates if the cia_ address value is the same for the incident's Source Object.

	Configuration Option	When to Use	Example
7	Rate	Select this tab to measure the rate of incoming incidents within a defined time period and correlate any incidents that occur within the specified time period. NNMi stores the following information related to rate: **Count**: Indicates the rate at which the incident must occur within the specified timeframe. **Hours, Minutes,** and **Seconds**: Used to measure the time within the rate must occur **First Occurrence Time**: Indicates the time at which the measured rate was reached. **Last Occurrence Time:** Indicates the last time which the incident occurred. NNMi updates the Correlation Notes with the number of incidents that have occurred within the specified time period. For example, 5 in 5 minutes. **Note**: A Rate Correlation incident inherits the Dampening settings of its Correlated Children. If the Correlated Children are Closed while Dampened, and therefore deleted, NNMi retains the parent Rate Correlation incident. See "Dampening Incident Configurations" (on page 356) for more information about Dampening an Incident Configuration.	If a connection is intermittently down three times within 30 minutes; correlate the Connection Down incidents.
8	Actions	Select this tab to 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 (**Registered**).	When an incident is generated (**Registered**), open a trouble ticket. After the incident is **Closed**, close the trouble ticket.
9	Forward to Global Managers	*NNMi Advanced - Global Network Management feature*). Select the Global Manager Forwarding tab when you want to forward specific SNMP traps from your NNMi management server (a Regional Manager) to all Global Managers in your Global Network Management environment.	Forward all CiscoLinkDown incidents to the Global Manager.
10	Pairwise	Select this tab to pair the first occurrence of an incident to another subsequent incident. **Note**: NNMi provides Correlation Notes information only when the Causal Engine has analyzed and Closed the incident. Software that is integrated with NNMi might also provide information identifying the reason an incident was Closed. Any time an incident is Closed manually (for example, by the network operator), NNMi does not provide Correlation Notes information.	Correlate a CiscoLinkDown incident as the Child Incident for a CiscoLinkUp incident.

NNMi also enables you to create your own Custom Correlations as described in the following table.

Custom Correlation Configuration Options

Configuration	Description
Custom Correlations	Enables you to correlate incidents using regular expressions to define the relationships between Parent and Child Incidents. This feature is useful when you want to define a relationship between a number of incidents potentially from different network objects that form a logical set to identify a problem. The set of correlations is considered complete if all of the incidents arrive within a specified time window. When configuring a Custom Correlation, you select the Parent and Child Incident configurations, the time window, and the regular expression that defines the relationship requirements that must be met before the incidents are correlated. See "Configure Custom Correlations" (on page 356) for more information about Custom Correlations.

See "Configuring Incidents" (on page 306) for more information about the Incident Configuration options. See "Load SNMP Trap Incident Configurations" (on page 429) for more information about how to specify which SNMP traps you want to receive by automatically creating or updating an incident configuration for an SNMP trap using a MIB file.

Related Topics

"Configure Management Events" (on page 728)

"Configure SNMP Trap Incidents" (on page 436)

"Configure Remote NNM 6.x/7.x Events" (on page 583)

Correlate Duplicate Incidents (Deduplication Configuration)

The deduplication configuration determines what values NNMi should match to detect when an SNMP trap, management event, or remote NNM 6.x/7.x event is a duplicate.

You can provide the required information within the following contexts:

"Deduplication Comparison Parameters Form (SNMP Trap Incident)" (on page 566)

"Deduplication Comparison Parameters Form (Remote NNM 6.x/7.x Events)" (on page 711)

"Deduplication Comparison Parameters Form (Management Events)" (on page 845)

Deduplication Comparison Parameters Form

Custom Incident Attributes (CIAs) are used as parameter values. Parameter values enable accurate identification of duplicate incidents. There are two categories of CIAs:

- SNMP trap varbind values (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312).

You can provide the required information within the following contexts:

"Deduplication Comparison Parameters Form (SNMP Trap Incident)" (on page 566)

"Deduplication Comparison Parameters Form (Remote NNM 6.x/7.x Events)" (on page 711)

"Deduplication Comparison Parameters Form (Management Events)" (on page 845)

Track Incident Frequency (Rate: Time Period and Count)

Use Rate Configuration to track incident patterns *based on the number of incident reoccurrences within a specified time period*. After the count within the specified time period is reached, NNMi emits a Rate Correlation incident and continues to update the Correlation Notes with the number of occurrences within that rate.

You can provide the required information within the following contexts:

"Configure Rate (Time Period and Count) for an SNMP Trap Incident" (on page 567)

"Configure Rate (Time Period and Count) for a Remote NNM 6.x/7.x Event Incident" (on page 712)

"Configure Rate (Time Period and Count) for a Management Event Incident" (on page 846)

Rate Comparison Parameters Form

Custom Incident Attributes (CIAs) are used as parameter values. Parameter values enable accurate identification of duplicate incidents. There are two categories of CIAs:

- SNMP trap varbind values (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312).

You can provide the required information within the following contexts:

"Rate Comparison Parameters Form (SNMP Trap Incident)" (on page 569)

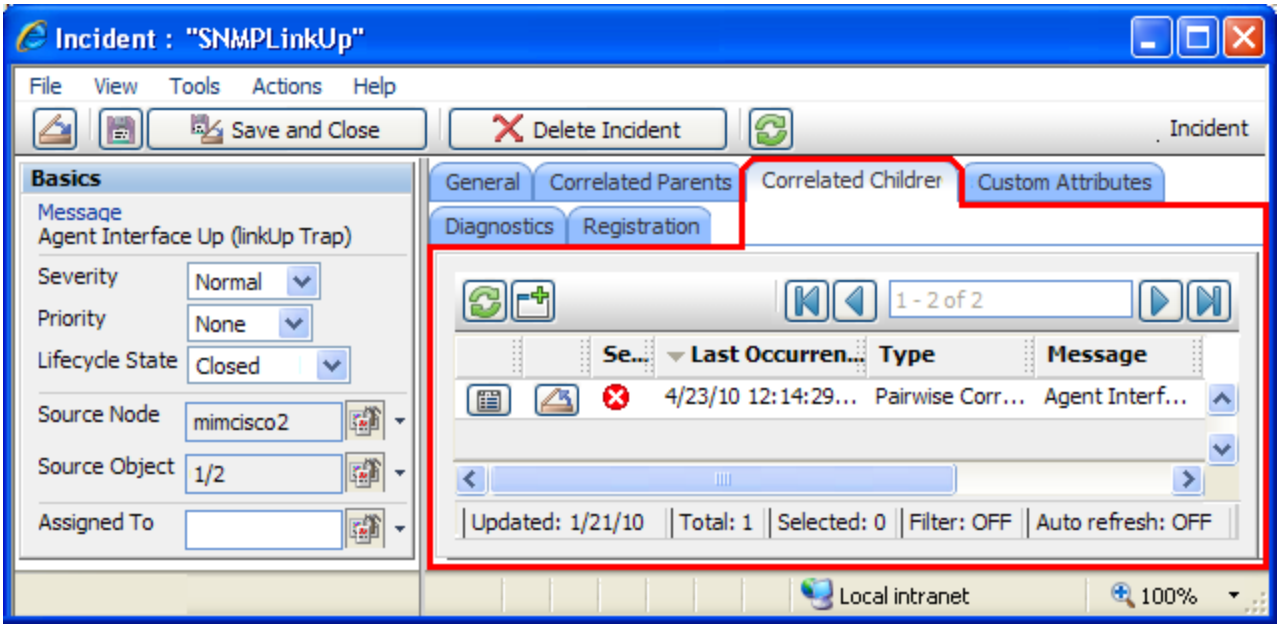
"Rate Comparison Parameters Form (Remote NNM 6.x/7.x Events)" (on page 714)

"Rate Comparison Parameters Form (Management Events)" (on page 848)

About Pairwise Configurations

Often two incidents have a logical relationship to each other, for example, CiscoLinkDown followed by CiscoLinkUp. There is no need for both incidents to take up room in your Incident view. Nesting the two together helps you do your job quickly and efficiently.

Use the Pairwise Configuration to pair up the occurrence of one incident with another subsequent incident. When the second incident in the pair occurs, the first incident becomes a correlated child incident within the parent incident (see illustration below). See "Incident Pair (Pairwise) Configurations Provided by NNM" (on page 348) for ideas.



NNM automatically ensures that the **Source Node** attribute value is identical in both incidents of your defined pair. Some incident pairs require extensive details to verify an accurate match. If *both sample incidents* have custom incident attributes, you can refine the match criteria beyond Source Node. See "Pair Item Configuration Form (Identify Incident Pairs)" (on page 353).

Related Topics:

"Prerequisites for Pairwise Configurations" (on page 350)

"Pairwise Configuration Form (Correlate Pairs of Incidents)" (on page 351)

Incident Pair (Pairwise) Configurations Provided by NNM

NNM provides the pairwise configurations described in the following table.

Pairwise Configurations Provided by NNM

Name	Description
CiscoLinkDownUpPair	Cancels a CiscoLinkDown incident with a CiscoLinkUp incident on the same interface for the same SNMP agent address. This configuration is used for known Cisco devices.
CiscoModuleDownUpPair	Not yet implemented.
OvApaAddressDownUpPair	Cancels an NNM 6.x or 7.x Node Down event with a NNM 6.x or 7.x Node Up event from the same SNMP agent address.
OvApaAggPortConnDownUpPair	Cancels an NNM 6.x or 7.x APA Aggregate Port Connection Down event with an NNM 6.x or 7.x APA Aggregate Port Connection Up event.
OvApaAggPortDegradeNotDegradePair	Cancels an NNM 6.x or 7.x APA Aggregate Port Degraded event with an NNM 6.x or 7.x APA Aggregate Port Not Degraded event on the same interface for the same SNMP agent address.

Name	Description
OvApaAggPortDownUpPair	Cancels of an NNM 6.x or 7.x APA Aggregate Port Down event with an NNM 6.x or 7.x APA Aggregate Port Up event on the same interface for the same SNMP agent address.
OvApaBoardDownUpPair	Cancels an NNM 6.x or 7.x APA Board Down event with an NNM 6.x or 7.x APA Board Up event from the same SNMP agent address.
OvApaConnDownUpPair	Cancels an NNM 6.x or 7. x APA Address Down event with an NNM 6.x or 7.x APA Address Up event on the same address for the same SNMP agent address.
OvApaIfDownUpPair	Cancels an NNM 6.x or 7.x APA If Down event with an NNM 6.x or 7.x APA If Up event on the same interface for the same SNMP agent address.
OvApaNodeDownUpPair	Cancels an NNM 6.x or 7.x APA Node Down event with an NNM 6.x or 7.x APA Node Up event from the same SNMP agent address.
OvIfDownUpPair	Cancels an NNM 6.x or 7.x If Down event with an NNM 6.x or 7.x If Up event on the same interface for the same SNMP agent address.
OvNodeDownUpPair	Cancels an NNM 6.x or 7.x Node Down event with an NNM 6.x or 7.x Node Up event from the same SNMP agent address.
RcAggLinkDownUpPair	Cancels an RcAggLinkDown incident with an RcAggLinkUp incident from the same Multi-Link Trunk (MLT) Aggregator ID (from the MIB) and SNMP agent address.
RcChasFanDownUpPair	Cancels an RcChasFanDown incident with an RcChasFanUp incident from the same fan ID (from the MIB) and SNMP agent address.
RcChasPowerSupplyDownUpPair	Cancels an RcChasPowerSupplyDown incident with an RcChasPowerSupplyUp incident from the same power supply ID (from the MIB) and SNMP agent address.
RcSmltlsLinkDownUpPair	Cancels an RcSmltlstLinkDown incident with an RcSmltlstLinkUp incident from the same SNMP agent address.
RcnAggLinkDownUpPair	Cancels an RcnAggLinkDown incident with an RcnAggLinkUp incident from the same Multi-Link Trunk (MLT) Aggregator Link ID (from the MIB) and SNMP agent address.
RcnChasFanDownUpPair	Cancels an RcnChasFanDown incident with an RcnChasFanUp incident from the same fan ID (from the MIB) and SNMP agent address.
RcnChasPowerSupplyDownUpPair	Cancels an RcnChasPowerSupplyDown incident with an RcnChasPowerSupplyUp incident from the same power supply ID (from the MIB) and SNMP agent address.

Name	Description
RcnSmltlsLinkDownUpPair	Cancels an RcnSmltlstLinkDown incident with an RcnSmltlstLinkUp incident from the same SNMP agent address.
SnmpLinkDownUpPair	Cancels an SNMPLinkDown incident with an SNMPLinkUp incident on the same interface for the same SNMP agent address.

To see or modify these incident pair configurations:

1. Navigate to the **Incident Configuration** view.
   a. In the Workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Incident Configuration** view.
2. Select the **Pairwise Configuration** tab.
3. Select the configuration you want to see or modify, and click Open to see or change the configuration.

   In the **Pairwise Configuration** form, click **Help→ Using the Pairwise Configuration form** for more information.
4. When you are finished, click **Save and Close** to save your changes and return to the Incident Configuration form.

Prerequisites for Pairwise Configurations

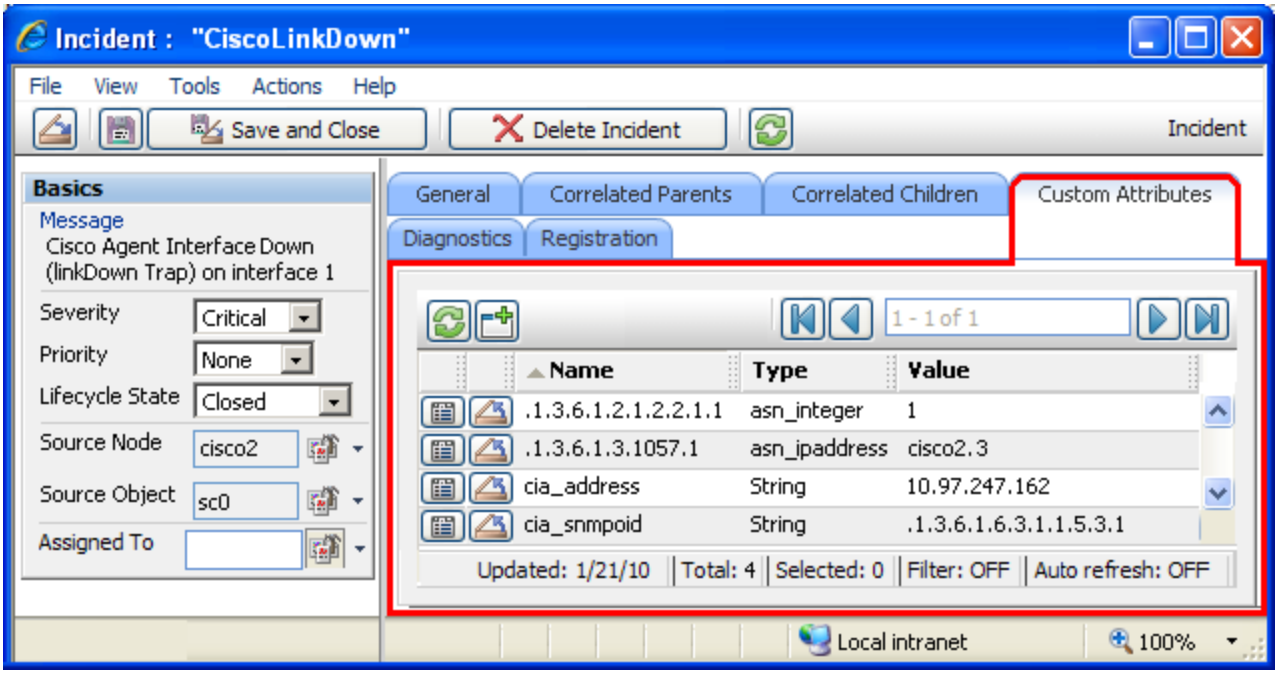
NNMi automatically ensures that the **Source Node** attribute value is identical in both incidents of your defined pair.

If you need to provide more details to accurately identify the logical pair of incidents (from among all possible incidents related to that source node), complete the Optional step 6 below.

Complete the following steps before attempting to set up a Pairwise Configuration:

1. Identify the two incidents or SNMP traps that consist of the logical relationship that makes the pair.
2. Configure those two incidents or traps within NNMi, if they are not already configured:
   - See "Incident Configurations Provided by NNMi" (on page 312).
   - See "Configure SNMP Trap Incidents" (on page 436).
   - See "Configure Remote NNM 6.x/7.x Events" (on page 583).
3. Generate one of each of the two incidents or SNMP traps so you can see an example of each in one of the NNMi Incident views. See "Views Provided by NNMi".
4. Select the first sample incident for the pair, and click Open to display the Incident form.

   Navigate to the Custom Attributes tab. These are the custom incident attributes available to use in step 6, below. See "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312) for more information about Custom Attributes.



5. Repeat the previous step with the second sample incident for the pair.
6. *Optional*. If *both sample incidents* have custom attributes, you can refine the match criteria beyond Source Node. Some incident pairs require extensive details to verify an accurate match. See "Pairwise Configuration Form (Correlate Pairs of Incidents)" (on page 351).

Pairwise Configuration Form (Correlate Pairs of Incidents)

Use the Pairwise Configuration to pair the occurrence of one incident with another subsequent incident. See "About Pairwise Configurations" (on page 347) for more information.

To configure incident pairs:

1. Complete the steps in "Prerequisites for Pairwise Configurations" (on page 350) so you know exactly which two incidents or traps belong to this logical pair.
2. Navigate to the **Pairwise Configuration** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select the **Incident Configuration** view.
    c. Navigate to the **Pairwise Configuration** tab.
    d. Do one of the following:
        - To create a new pair configuration, click the New icon, and continue.
        - To edit an existing pair configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        - To delete a pair configuration, select a row and click the Delete icon.
3. Provide the basic definition of the pair of incidents for this correlation (see table).
4. NNM automatically ensures that the **Source Node** attribute value is identical in both incidents of your

defined pair. Some incident pairs require extensive details to verify an accurate match. If *both sample incidents* have custom incident attributes, you can refine the match criteria beyond Source Node.

Optional. Navigate to the **Pair Items** tab, and provide one or more custom incident attribute sets for NNMi to use as a filter when identifying a valid pair of incidents. See "Pair Item Configuration Form (Identify Incident Pairs)" (on page 353).

Then, click **Save and Close** to save your changes and return to the previous configuration form.

For example:

- If you specify a First In Pair and Second In Pair of .1.3.6.1.2.1.2.2.1.1, the first incident's varbind value for the specified OID must match the second incident's varbind value for the specified OID to confirm a match.
- If you specify two custom attribute sets (one with both First In Pair and Second In Pair set to position 7, and one with both First In Pair and Second In Pair set to position 25), then the values for both custom attributes (varbind position 7 and varbind position 25) in both Incidents must match to confirm the logical pair.

The next time the two incidents in this pair are generated, the first one becomes a Child Incident of the second one. See "About Pairwise Configurations" (on page 347) for an example.

Pairwise Configuration Definition

Attribute	Description
Name	The name is used to identify the pairwise configuration and must be unique. Use a name that will help you to remember the purpose for this pairwise configuration. Maximum length is 64 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. No spaces are allowed.
Enable	In the **Basics** group, verify that **Enable** is selected.
First Incident Configuration	Identify the incident in the pair that would occur first in the logical sequence. Click the Lookup icon and select **Quick Find**. Choose the name of one of the predefined incident configurations. If you cannot find it, see: • See "Incident Configurations Provided by NNMi" (on page 312). • See "Configure SNMP Trap Incidents" (on page 436). • See "Configure Remote NNM 6.x/7.x Events" (on page 583).
Second Incident Configuration	Identify the incident in the pair that would occur second in the logical sequence.Click the Lookup icon and select **Quick Find**. Choose the name of one of the predefined incident configurations. If you cannot find it, see: • See "Incident Configurations Provided by NNMi" (on page 312). • See "Configure SNMP Trap Incidents" (on page 436). • See "Configure Remote NNM 6.x/7.x Events" (on page 583).
Description	*Optional*. Explain the purpose of your pairwise configuration for future reference. Type a maximum of 1024 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.
Author	Indicates who created or last modified the Correlation Rule. See Author form for important information.

Attribute	Description
	Caution: If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. Click the Lookup icon and select Quick Find to access the list of existing Author values or click New to create one.

Pair Item Configuration Form (Identify Incident Pairs)

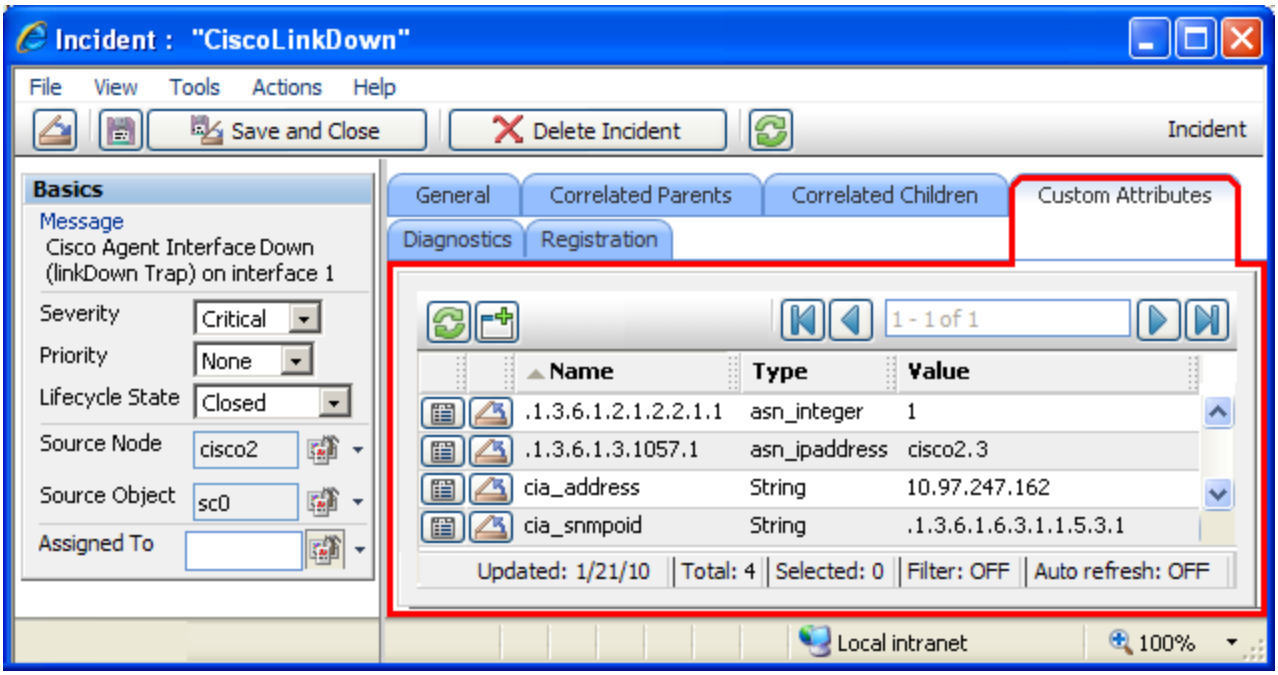
NNMi automatically ensures that the **Source Node** attribute value is identical in both incidents in your defined pair. Some incident pairs require extensive details to verify an accurate match. If *both sample incidents* have custom incident attributes, you can use the Pair Item Configuration form to refine the match criteria beyond Source Node.

Specify one or more values for NNMi to use as a filter when identifying a valid pair of incidents.

You can use any Custom Incident Attributes (CIAs) displayed on the Incident form of the two incidents you are associating into a logical pair. The group of available CIAs depends on which incidents you select. There are two categories of possible CIAs:

- SNMP trap varbinds identified by the Abstract Syntax Notation value (ASN.1) or position. For example, a varbind OID of .1.3.6.1.2.1.2.2.1.1 or a position number of 25.
- Custom attributes provided by NNMi (Name = cia_*). See "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312).

  The group of available CIAs depends on which incident you are configuring (for example, CiscoLinkDown). To see which CIAs are available, navigate to an Incident view, select an instance of that incident-type, click the Open icon and navigate to the Custom Attributes tab. The items listed in the table are the CIAs for that particular incident-type. For example, all CiscoLinkDown incidents would have the same group of CIAs shown in the illustration below.



To configure which attributes NNMi uses to verify incident identity:

1. Complete the steps in "Prerequisites for Pairwise Configurations" (on page 350) so your choices for this Item Pair configuration are displayed in the NNMi console. (Two Incident forms should be open before you proceed to step 2.)
2. Navigate to the **Pair Item Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Incident Configuration** view.
   c. Navigate to the **Pairwise Configuration** tab.
   d. Do one of the following:
      - To create a new pair configuration, click the New icon.
      - To edit a pair configuration, click the Open icon in the row representing the configuration you want to edit.
   e. On the **Pairwise Configuration** form, locate the **Pair Items** tab.
   f. Do one of the following:
      - To create a new pair item configuration, click the New icon.
      - To edit a pair item configuration, click the Open icon in the row representing the configuration you want to edit.
3. Specify the attributes you want NNMi to use to confirm the identity of the pair of incidents (see table).
4. Click **Save and Close** to save your changes and return to the previous form.
5. Repeat steps 1-3 any number of times. The incidents must pass all Pair Item criteria, plus have identical Source Node attribute values.

Pair Item Configuration

Attribute	Description
First In Pair	Type the specification required to confirm the identify of the first incident in this logical pair of incidents. Provide one of the following: • The SNMP trap varbind Abstract Syntax Notation value - ASN.1 (OID) • The SNMP trap varbind position number **Caution**: The varbind position numbers are not visible in the table on the Incident form's Custom Attributes tab. And the rows in that table are sorted by the visible column headings and are not in varbind position order. You must access the vendor-supplied information in the underlying MIB file to determine the appropriate position number for any particular varbind. • The Custom Attribute **Name** value (see "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312) or the Name column in the table on the Incident Form: Custom Attributes Tab of the Incident you are configuring as a member of this logical pair).
Second In Pair	Type the specification required to confirm the identify of the second incident in this logical pair of incidents. Provide one of the following: • The SNMP trap varbind Abstract Syntax Notation value - ASN.1 (OID) • The SNMP trap varbind position number

Attribute	Description
	Caution: The varbind position numbers are not visible in the table on the Incident form's Custom Attributes tab. And the rows in that table are sorted by the visible column headings and are not in varbind position order. You must access the vendor-supplied information in the underlying MIB file to determine the appropriate position number for any particular varbind. • The Custom Attribute **Name** value (see "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312) or the Name column in the table on the Incident Form: Custom Attributes Tab of the Incident you are configuring as a member of this logical pair).

Related Topics

"Incident Pair (Pairwise) Configurations Provided by NNM" (on page 348)

Suppress Incident Configurations

NNMi enables you to suppress incidents based on Interface Group, Node Group, or default Suppression settings. NNMi applies your Suppression settings in the following order. Only the first match applies.

1. Interface Group (SNMP Trap Configuration Form: Interface Settings tab)
2. Node Group (SNMP Trap Configuration Form: Node Settings tab)
3. Enrich configuration settings without specifying an Interface Group or Node Group (SNMP Trap Configuration Form: Enrichment tab)

You can provide the required information within the following contexts:

"Configure Suppression Settings for an SNMP Trap Incident"

"Configure Suppression Settings for a Management Event Incident" (on page 815)

"Configure Suppression Settings for a Remote NNM 6.x/7.x Event Incident" (on page 675)

Enrich Incident Configurations

NNMi enables you to fine tune and enhance incidents based on Interface Group, Node Group, or default Enrichment settings. NNMi applies your Enrichment settings in the following order. Only the first match applies:

1. Interface Group (Interface Settings tab)
2. Node Group (Node Settings tab)
3. Enrich configuration settings without specifying an Interface Group or Node Group (Enrichment tab)

The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature

- Message
- Assigned To

Note: Any configuration you specify for Severity, Priority, or Message overrides those values provided in the Basics information.

You can provide the required information within the following contexts:

"Configure Enrichment Settings for an SNMP Trap Incident" (on page 542)

"Configure Enrichment Settings for a Management Event Incident" (on page 826)

"Configure Enrichment Settings for a Remote NNM 6.x/7.x Event Incident" (on page 684)

Dampening Incident Configurations

NNMi enables you to delay (dampen) the following for an incident configuration:

- Appearance within Incident views in the NNMi Console
- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software* \ *NNM iSPI NET*)

You can provide the required information within the following contexts:

"Configure Dampening Settings for an SNMP Trap Incident" (on page 545)

"Configure Dampening Settings for a Management Event Incident" (on page 829)

"Configure Dampening Settings for a Remote NNM 6.x/7.x Event Incident" (on page 687)

Configure Custom Correlations

For information about each Custom Correlation Configuration tab:

NNMi enables you to correlate groups of incidents under a Parent Incident. This feature is useful when you want to define a relationship between a number of incidents potentially from different network objects that form a logical set to identify a problem. The set of correlations is considered complete if all of the incidents arrive within a specified time window.

When configuring a Custom Correlation, you configure one or both of the following:

Rule	Description
Correlatio Rule	**Tip**: Configure a Correlation Rule when you want to correlate only one type of Child incident Configuration with a Parent Incident Configuration that is generated by NNMi. Use a Correlation Rule to specify the following: • Parent Incident Configuration • Child Incident Configuration • Filters that NNMi should use when selecting the Parent and Child Incident instances for correlation • The time window within which NNMi begins to correlate the incidents. **Note**: If the Parent and Child incidents occur within the Correlation Window Duration specified, NNMi begins to correlate the incidents as soon as they occur.

Rule	Description
	• The regular expression (Correlation Filter) that defines the relationship requirements that must be met before the incidents are correlated The Parent and Child Incident do not have to be the same incident configuration. For example, you can correlate an Address Not Responding incident with an Interface Down incident. See "Correlation Rule Example" (on page 382) for a step-by-step example of how the Subinterface Correlation Rule provided by NNMi was created.
Causal Rule	**Tip**: Configure a Causal Rule when you want to cause NNMi to generate a Parent Incident and you want to correlate one or more Child Incident Configurations under the Parent Incident that you cause to be generated. Use a Causal Rule to specify the following: • Parent Incident Configuration to be generated • One or more Child Incident Configurations to be correlated under the generated Parent Incident • Filters that NNMi should use when selecting the Child Incident instances for correlation • Source Object and Source Node filters to be used to determine the Source Node and Source Object for the generated Parent Incident • The time window that must be met before NNMi correlates the incidents. **Note**: NNMi waits until the Correlation Window Duration has passed before generating the Parent Incident and correlating its Child Incidents. To establish a relationship between multiple Custom Correlations, configure a Causal Rule to generate a Parent Incident that becomes the Child Incident of another Parent Incident. See "Causal Rule Example" for a step-by-step example of creating a Causal Rule.

To configure a Custom Correlation:

1. Navigate to the **Custom Correlation Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Select **Custom Correlation Configuration**.
2. View the configured attributes (see table).
3. Do one of the following, or both:
   - Configure one or more Correlation Rules. See "Configure a Correlation Rule" (on page 358) for more information.
   - Configure one or more Causal Rules. See "Configure a Causal Rule" (on page 385) for more information.
4. Click **Save and Close** to save your changes and return to the previous form.

Custom Correlation Registration Attribute

Attribute	Description
Last Modified	The date and time the Custom Correlation configuration was last modified.

Configure a Correlation Rule

Tip: Configure a Correlation Rule when you want to correlate a Child incident Configuration under a Parent Incident Configuration that is generated by NNMi.

Note: See **Help** → **Documentation Library** → **Release Notes**, and locate the **Support Matrix** link for Correlation Rule limitations.

When correlating groups of incidents under an existing Parent incident, use the Correlation Rules tab to specify the Correlation Rule that defines the Parent Incident, the Child Incident, and the relationship requirements that must be met before the incidents are correlated.

See "Correlation Rule Example" (on page 382) for a step-by-step example of how the Subinterface Custom Correlation Rule provided by NNMi was created.

For information about each Correlation Rules tab:

To configure a Correlation Rule:

1. Navigate to the **Custom Correlation Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Select **Custom Correlation Configuration**.
2. Navigate to the **Correlation Rules** tab.
3. From the **Correlation Rules** table toolbar, do one of the following:
   - To create a Correlation Rule, click the New icon, and continue.
   - To edit a Correlation Rule, click the Open icon in the row representing the Correlation Rule you want to edit, and continue.
   - To delete a Correlation Rule, click the Delete icon.
4. Create your Correlation Rule (see table).
5. Click **Save and Close** to save your changes and return to the previous form.

Correlation Rule Basic Attributes

Attribute	Description
Name	Type a maximum of 64 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. The name is used to identify the Correlation Rule and must be unique. Use a name that will help you to remember the purpose of the Correlation Rule.
Author	Indicates who created or last modified the Correlation Rule. See Author form for important information. **Caution**: If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. Click the Lookup icon and select Quick Find to access the list of existing Author values or click New to create one.
Enabled	If enabled, the NNMi Causal Engine uses the Correlation Rule when evaluating incidents. If disabled, the Correlation Rule is ignored.

Attribute	Description
Parent Incident	Specifies the incident configuration that should be used as the Parent Incident for the Correlation Rule. **Note**: If you want to create a rule to *generate* a Parent Incident configure a Causal Rule. See "Configure a Causal Rule" (on page 385) for more information. **To specify a Parent Incident configuration**: 1. Click the Lookup icon, and do one of the following: ▪ To specify a Parent Incident without making any changes to the Incident configuration, select Quick Find . In the Quick Find dialog, select the Incident of interest. ▪ To create a Parent Incident, select one of the following: ○ New Management Event Configuration ○ New Remote NNM Event Configuration ○ New SNMP Trap Configuration ▪ To modify a Parent Incident, select Open. 2. *Optional*. To create or modify a Parent Incident, enter or modify the attribute values for the selected Incident configuration. See "Configuring Incidents" (on page 306) for more information about the Incident Configuration form. 3. Click **Save and Close** to save your changes and return to the previous form.
Child Incident	Specifies the incident configuration that must match an incoming incident and that should be correlated as the Child Incident for the Custom Correlation. **To specify a Child Incident configuration**: 1. Click the Lookup icon, and select Quick Find. 2. In the Quick Find dialog, select the Incident of interest.
Correlation Window Duration	The time window within which NNMi begins to correlate the incidents. Enter a number for Days, Hours, Minutes, and Seconds. Note the following: • If the Parent and Child incidents occur within the Correlation Window Duration specified, NNMi begins to correlate the incidents as soon as they occur. • If you are relating multiple Custom Correlations, make sure the Correlation Window Duration allows enough time for all of the Parent and Child incidents to be generated.
Description	Use the description field to provide additional information that you would like to store about the current incident configuration. This description applies only to the configuration entry. Type a maximum of 2048 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Configure a Parent Incident Filter for a Correlation Rule

Note: See **Help** → **Documentation Library** → **Release Notes**, and locate the **Support Matrix** link for Parent Incident Filter limitations.

Tip: The Parent Incident Filter is optional, but recommended. Use of a Parent Incident Filter improves NNMi performance by reducing the set of incidents that NNMi processes.

When correlating groups of incidents under a Parent Incident, you can define the requirements for the Parent Incident. The Parent Incident tab enables you to use the Filter Editor to define these requirements. For example, you might want to specify that the Source Node of the Parent Incident be a specific node Name pattern. See Valid Operators in the table that follows for examples of valid Parent Incident Filters.

When specifying the **like** or **not like** operator, use the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information.

Use the Filter Editor Buttons to insert Boolean Operators and to append, insert, and replace expressions in the Filter String. Use the Drag and Drop feature to make changes to the placement of the expressions in your Filter String. Click here for more information about using the Filter Editor for Custom Correlations:

- You can use Custom Incident Attributes, attributes for an incident's Source Node or Source Object, or both to define how matching incidents should be considered for the Correlation Rule. See Valid Attributes for more information.
- When specifying Attribute names and values, NNMi uses the type to determine a match. For example, if the Attribute type is numeric, NNMi does a numeric comparison. If the Attribute type is textual, NNMi does a lexographical string comparison. In all cases, when you use the **like** or **not like** operator, NNMi uses a lexographical string comparison. Click here for more information about Attribute types:
  - `ifIndex` and `ifSpeed` are numeric Attributes.
  - Any Attribute name that begins wtih "is" (`isSnmpInterface`, `isSnmpNode`, `isNnmSystemLocal`) represents a Boolean Attribute.
  - All other Attributes are textual.
- Each set of expressions associated with a Boolean Operator (for example, `AND`) is treated as if it were enclosed in parentheses and evaluated together.
- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The `AND` and `OR` Boolean Operators must contain at least two expressions.
- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Additional Filters Editor. For example, you append to or replace, the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators. See "Add Boolean Operators in the Additional Filters Editor" (on page 193) for more information.

Filter Editor Buttons and Drag and Drop Feature

Button or Feature	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.

Button or Feature	Description
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed Left or Right Expression.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Filter Editor deletes all expressions associated with the Boolean Operator.
Drag and Drop	You can drag any of the following items to a new location in the Filter String: ■ Filter Editor Options: AND, OR, NOT, EXISTS, NOT EXISTS ■ Filter Expression (Attribute, Operator and Value) When moving items in the Filter String, note the following: ■ Click the item you want to move before dragging it to a new location. ■ As you drag a selected item, an underline indicates the target location. ■ If you are moving the selection up, NNMi places the item above the target location. ■ If you are moving the selection down, NNMi places the item below the target location. ■ If you attempt to move the selection to an invalid target location, NNMi displays an error message.

See "Correlation Rule Example" (on page 382) for a step-by-step example of how the Subinterface Correlation Rule provided by NNMi was created.

To configure a Parent Incident Filter:

1. Navigate to the **Custom Correlation Configuration** form:
    a. From the workspace navigation pane, select the **Configuration** workspace.
    b. Select **Custom Correlation Configuration**.
2. Navigate to the **Correlation Rules** tab.
3. From the **Correlation Rules** table toolbar, do one of the following:

- To create a Correlation Rule, click the New icon, and continue.
- To edit a Correlation Rule, click the Open icon in the row representing the configuration you want to edit, and continue.
- To delete a Correlation Rule, click the Delete icon.

4. Navigate to the **Parent Incident Filter** tab.
5. Create your Parent Incident Filter (see Filter Editor Components).

Filter Editor Components

Component	Description
Attribute	The Attribute on which NNMi searches. See Valid Attributes below for a description of valid Attributes.
Operator	Use this Operator to establish the relationship between the Attribute and Expression. See Valid Operators in the table below for the description of each valid Operator.
Expression	Use the Expression to complete the criteria for the Parent Incident configuration. See Valid Expressions below for a description of the components that you might include in your Expression.

6. Click **Save and Close** to save your changes and return to the previous form.

Valid Attributes

Attribute	Description
Attribute	The Attribute on which NNMi searches. Valid attributes other than Source Node attributes depend on the Incident's Source Object. NNMi checks the Source Node as well as the Source Object for any Capability value. Note the following when specifying Attributes: • Boolean Attributes begin with "is" and must contain the value `true` or `false`. • Use the following syntax to specify a Custom Incident Attribute (CIA): `valueOfCia(<CIA_Name>)` **Note**: Check the appropriate Incident form for any valid CIA Names provided by NNMi. For example: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} = 5` • When specifying *CIA_Name* or *CA_Name* you can use the asterisk (*) as a wildcard character to specify one or more characters and a question mark (?) to specify a single character. • If you use attributes that are valid for the Source Node, NNMi uses the Source Node when comparing values. If you use attributes that are valid for the Source Object, NNMi uses the Source Object when comparing values. You cannot use attributes that are valid for the Source Node and Source Object in the same filter. • If the incident does not have a Source Object, a Source Node, or both (for example, the node is not stored in the NNMi database), you must use CIAs in your filter rather than Source Object or Source Node attributes. **Tip**: To check whether the Source Object or Source Node is stored in the NNMi database, open the incident and then select Open from the Lookup menu for the Source

Attribute	Description

Node or Source Object displayed. If a form does not open for the selected object or node, this means the source is not stored in the NNMi database.

- When specifying a Correlation Filter, precede the attribute name with either `parent.` or `child.` to specify from which incident the attribute value should be compared. For example, you might specify `${parent.hostedOn}` or `${child.ifDesc}`.

Possible Source Object choices are as follows:

- Card [click here for a list of attribute values]

  Unique Keys from the Card Form: Capabilities Tab:

  - capability (Unique Key of the Capability)

- Interface [click here for a list of attribute values]

  Use the following syntax to specify a Custom Attribute (CA) for an Interface:

  `valueOfInterfaceCa(<CA_Name>)`

  For example: `${child.valueOfInterfaceCA(Role)} = WAN Connection`

  Values from the Basics Attributes listed on the Interface Form:

  - hostedOn (Hosted On Node)

    Note: You must use the full DNS name for the hostedOn value.

  Values from the Interface Form: General Tab:

  - ifName (name configured for the interface)
  - ifAlias (alias configured for the interface)
  - ifDescr (description configured for the interface)
  - ifIndex (index assigned to the interface)
  - ifSpeed (speed configured for the interface)

    Note: When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**.

  Addresses from the Interface Form: IP Addresses Tab:

  - ipAddress (IP Address associated with the interface)

    Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=**.

  Unique Keys from the Interface Form: Capabilities Tab:

  - capability (Unique Key of the Capability)

  Values from the Basics Attributes on the parent Node Form:

  - isSnmpInterface (Agent Enabled)

  Values from the parent Node Form: General Tab:

  - sysOidInterface (System Object ID)

  Values from the Basics Attributes on the associated Device Profile Form:

Attribute	Description
	▪ devVendorInterface (Device Vendor) ▪ devFamilyInterface (Device Family) • IP Address [click here for a list of attribute values] **Unique Keys from the IP Address Form: Capabilities Tab**: ▪ capability (Unique Key of the Capability) • Node [click here for a list of attribute values] Use the following syntax to specify a Custom Attribute (CA) for a Node: `valueOfNodeCa(<CA_Name>)` For example: `${valueOfNodeCA(Location)} = USA` **Values from the Basics Attributes on the Node Form**: ▪ hostname (Hostname, *case-sensitive*) ▪ mgmtIPAddress (Management Address) ▪ isSnmpNode (Agent Enabled) ▪ isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: ▪ sysName (System Name) ▪ sysContact (System Contact) ▪ sysLocation (System Location) ▪ sysOidNode (System Object ID) **Addresses from the Node Form: IP Addresses Tab**: ▪ hostedIPAddress (Address) Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=** . **Unique Keys from the Node Form: Capabilities Tab**: ▪ capability (Unique Key of the Capability) **Values from the Basics Attributes on the associated Device Profile Form**: ▪ devVendorNode (Device Vendor) ▪ devFamillyNode (Device Family) **Values from the associated entry on the Regional Manager Form: Connection Tab**: ▪ nnmSystemName (Hostname, *case-sensitive*) (*NNMi Advanced*) If the Global Network Management feature is enabled, this attribute value identifies a Regional Manager (NNMi management server).

Attribute	Description

Valid Operator Values

Operator	Description

=	Finds all values equal to the value specified.
	Click here for examples.
	Match any incident with a CIA value of 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1:
	`${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} = 5`
	Match any incident with the Source Object's Capability equal to `com.hp.nnm.capability.card.fru`
	`$(capability) = com.hp.nnm.capability.card.fru`

!=	Finds all values not equal to the value specified. Click here for an example. Match any incident with Device Vendor for the interface (Source Object) not equal to `Cisco`: `${devVendorInterface} != Cisco`
<	Finds all values less than the value specified. Click here for an example. Match any incident with a CIA value of less than 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} < 5`
<=	Finds all values less than or equal to the value specified. Click here for examples. Match any incident with a CIA value of less than or equal to 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} <= 5`
>	Finds all values greater than the value specified. Click here for an example. Match any incident with a CIA value of greater than 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1: ${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} > 5
>=	Finds all values greater than or equal to the value specified. Click here for an example. Match any incident with a Source Object's (interface speed) ifSpeed value of 10Mbps: `${ifSpeed} >= 10000000`
is not null	Finds all non-blank values. Click here for an example. Match any incident with a Source Object's (interface name) ifName attribute that contains a

Operator	Description
	value: `${ifName} is not null`
is null	Finds all blank values. Click here for an example. Match any incident with a Source Object's (interface name) ifName attribute that does not contain a value: `${ifName} is null`

like	Finds matches using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information.
	Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E .
	The period asterisk (.*) characters mean *any number of characters of any type at this location*.
	The period (.) character means *any single character of any type at this location*.
	Click here for an example.
	Match any incident with a Source Object's (interface description) ifDesc attribute that includes `Serial` followed by one or more digits:
	`${ifDesc} like Serial\d+`
	Match any incident with a Source Object's (interface alias) ifAlias attribute that contains `EtherChannel` (for example, `PAgPEtherChannel Group 1`).
	Note: The . (period) indicates any alphanumeric character.
	`${ifAlias} like .*EtherChannel.*`
	Match any incident with a CIA attribute value of `Chassis Fan Tray` followed by a digit and Object Identifier (OID) of `.1.3.6.1.4.1.9.9.13.1.4.1.3`
	Note: To include literal strings in the value, enclose the string value in \Q<*literal_value*>\E as shown in the following example.
	`${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.3\E)} like Chassis Fan Tray \d`

Operator	Description
not like	Finds all matches that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E . The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. Click here for an example. Match any incident with a Source Object's (interface name) ifName value that does not include `rtr`: `${ifName} not like .*rtr.*`

Valid Expressions

Attribute	Description
Expression	The value or pattern for which you want NNMi to search. Note the following: • The expression can include a valid Attribute. • The value or pattern you want to match is case sensitive. • When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**.

Configure a Child Incident Filter for a Correlation Rule

Note: See **Help → Documentation Library → Release Notes**, and locate the **Support Matrix** link for Child Incident Filter limitations.

Tip: The Child Incident Filter is optional, but recommended. Use of a Child Incident Filter improves NNMi performance by reducing the set of incidents that NNMi processes.

When correlating groups of incidents under a Parent incident, you must specify the requirements for the Child Incident. The Child Incident tab enables you to use the Filter Editor to define these requirements. For example, you might want to specify that the Source Node of the Child Incident be a specific Node Name pattern. See Valid Operators in the table that follows for examples of valid Child Incident Filters.

Use the Filter Editor Buttons to insert Boolean Operators and to append, insert, and replace expressions in the Filter String. Use the Drag and Drop feature to make changes to the placement of the expressions in your Filter String. Click here for more information about using the Filter Editor for Custom Correlations:

- You can use Custom Incident Attributes, attributes for an incident's Source Node or Source Object, or both to define how matching incidents should be considered for the Correlation Rule. See Valid Attributes for more information.
- When specifying Attribute names and values, NNMi uses the type to determine a match. For example, if the Attribute type is numeric, NNMi does a numeric comparison. If the Attribute type is textual, NNMi does a lexographical string comparison. In all cases, when you use the **like** or **not like** operator, NNMi uses a lexographical string comparison. Click here for more information about Attribute types:

  - `ifIndex` and `ifSpeed` are numeric Attributes.
  - Any Attribute name that begins wtih "is" (`isSnmpInterface`, `isSnmpNode`, `isNnmSystemLocal`) represents a Boolean Attribute.
  - All other Attributes are textual.
- Each set of expressions associated with a Boolean Operator (for example, `AND`) is treated as if it were enclosed in parentheses and evaluated together.
- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The `AND` and `OR` Boolean Operators must contain at least two expressions.
- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Additional Filters Editor. For example, you append to or replace, the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators. See "Add Boolean Operators in the Additional Filters Editor" (on page 193) for more information.

Filter Editor Buttons and Drag and Drop Feature

Button or Feature	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed Left or Right Expression.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Filter Editor deletes all expressions associated with the Boolean Operator.
Drag and Drop	You can drag any of the following items to a new location in the Filter String: ■ Filter Editor Options: AND, OR, NOT, EXISTS, NOT EXISTS ■ Filter Expression (Attribute, Operator and Value) When moving items in the Filter String, note the following: ■ Click the item you want to move before dragging it to a new location. ■ As you drag a selected item, an underline indicates the target location.

Button or Feature	Description
	■ If you are moving the selection up, NNMi places the item above the target location. ■ If you are moving the selection down, NNMi places the item below the target location. ■ If you attempt to move the selection to an invalid target location, NNMi displays an error message.

See "Correlation Rule Example" (on page 382) for a step-by-step example of how the Subinterface Correlation Rule provided by NNMi was created.

To configure a Child Incident Filter:

1. Navigate to the **Custom Correlation Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Select **Custom Correlation Configuration**.
2. Navigate to the **Correlation Rules** tab.
3. From the **Correlation Rules** table toolbar, do one of the following:
   - To create a Correlation Rule, click the New icon, and continue.
   - To edit a Correlation Rule, click the Open icon in the row representing the Correlation Rule you want to edit, and continue.
   - To delete a Correlation Rule, click the Delete icon.
4. Navigate to the **Child Incident Filter** tab.
5. Create your Child Incident Filter (see the Filter Editor Components below).

   Filter Editor Components

Component	Description
Attribute	The Attribute on which NNMi searches. See Valid Attributes below for a description of valid Attributes.
Operator	Use this Operator to establish the relationship between the Attribute and Expression. See Valid Operators in the table below for the description of each valid Operator. **Note**: When specifying the **like** or **not like** operator, you must use the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information.
Expression	Use the Expression to complete the criteria for the Child Incident configurations. See Valid Expressions below for a description of the components that you might include in your Expression.

6. Click **Save and Close** to save your changes and return to the previous form.

Valid Attributes

Attribute	Description

Attribute	The Attribute on which NNMi searches. Valid attributes other than Source Node attributes depend on the Incident's Source Object. NNMi checks the Source Node as well as the Source Object for any Capability value.
	
	Note the following when specifying Attributes:
	
	• Boolean Attributes begin with "is" and must contain the value `true` or `false`.
	
	• Use the following syntax to specify a Custom Incident Attribute (CIA):
	
	`valueOfCia(<CIA_Name>)`
	
	Note: Check the appropriate Incident form for any valid CIA Names provided by NNMi.
	
	For example: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} = 5`
	
	• When specifying *CIA_Name* or *CA_Name* you can use the asterisk (*) as a wildcard character to specify one or more characters and a question mark (?) to specify a single character.
	
	• If you use attributes that are valid for the Source Node, NNMi uses the Source Node when comparing values. If you use attributes that are valid for the Source Object, NNMi uses the Source Object when comparing values. You cannot use attributes that are valid for the Source Node and Source Object in the same filter.
	
	• If the incident does not have a Source Object, a Source Node, or both (for example, the node is not stored in the NNMi database), you must use CIAs in your filter rather than Source Object or Source Node attributes.
	
	Tip: To check whether the Source Object or Source Node is stored in the NNMi database, open the incident and then select Open from the Lookup menu for the Source Node or Source Object displayed. If a form does not open for the selected object or node, this means the source is not stored in the NNMi database.
	
	• When specifying a Correlation Filter, precede the attribute name with either `parent.` or `child.` to specify from which incident the attribute value should be compared. For example, you might specify `${parent.hostedOn}` or `${child.ifDesc}`.
	
	Possible Source Object choices are as follows:
	
	• Card [click here for a list of attribute values]
	
	Unique Keys from the Card Form: Capabilities Tab:
	
	▪ capability (Unique Key of the Capability)
	
	• Interface [click here for a list of attribute values]
	
	Use the following syntax to specify a Custom Attribute (CA) for an Interface:
	
	`valueOfInterfaceCa(<CA_Name>)`
	
	For example: `${child.valueOfInterfaceCA(Role)} = WAN Connection`
	
	Values from the Basics Attributes listed on the Interface Form:
	
	▪ hostedOn (Hosted On Node)
	
	Note: You must use the full DNS name for the hostedOn value.
	
	Values from the Interface Form: General Tab:

Attribute	Description

- ifName (name configured for the interface)
- ifAlias (alias configured for the interface)
- ifDescr (description configured for the interface)
- ifIndex (index assigned to the interface)
- ifSpeed (speed configured for the interface)

  Note: When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**.

Addresses from the Interface Form: IP Addresses Tab:

- ipAddress (IP Address associated with the interface)

  Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=**.

Unique Keys from the Interface Form: Capabilities Tab:

- capability (Unique Key of the Capability)

Values from the Basics Attributes on the parent Node Form:

- isSnmpInterface (Agent Enabled)

Values from the parent Node Form: General Tab:

- sysOidInterface (System Object ID)

Values from the Basics Attributes on the associated Device Profile Form:

- devVendorInterface (Device Vendor)
- devFamilyInterface (Device Family)

- IP Address [click here for a list of attribute values]

  Unique Keys from the IP Address Form: Capabilities Tab:

  - capability (Unique Key of the Capability)

- Node [click here for a list of attribute values]

  Use the following syntax to specify a Custom Attribute (CA) for a Node:

  `valueOfNodeCa(<CA_Name>)`

  For example: `${valueOfNodeCA(Location)} = USA`

  Values from the Basics Attributes on the Node Form:

  - hostname (Hostname, *case-sensitive*)
  - mgmtIPAddress (Management Address)
  - isSnmpNode (Agent Enabled)
  - isNnmSystemLocal (NNMi Management Server)

  Values from the Node Form: General Tab:

  - sysName (System Name)

Attribute	Description
	■ sysContact (System Contact) ■ sysLocation (System Location) ■ sysOidNode (System Object ID) **Addresses from the Node Form: IP Addresses Tab**: ■ hostedIPAddress (Address) Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=** . **Unique Keys from the Node Form: Capabilities Tab**: ■ capability (Unique Key of the Capability) **Values from the Basics Attributes on the associated Device Profile Form**: ■ devVendorNode (Device Vendor) ■ devFamillyNode (Device Family) **Values from the associated entry on the Regional Manager Form: Connection Tab**: ■ nnmSystemName (Hostname, *case-sensitive*) (*NNMi Advanced*) If the Global Network Management feature is enabled, this attribute value identifies a Regional Manager (NNMi management server).

Attribute	Description

Valid Operator Values

Operator	Description

=	Finds all values equal to the value specified.
	Click here for examples.
	Match any incident with a CIA value of 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1:
	`${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} = 5`
	Match any incident with the Source Object's Capability equal to `com.hp.nnm.capability.card.fru`
	`$(capability) = com.hp.nnm.capability.card.fru`

!=	Finds all values not equal to the value specified. Click here for an example. Match any incident with Device Vendor for the interface (Source Object) not equal to `Cisco`: `${devVendorInterface} != Cisco`
<	Finds all values less than the value specified. Click here for an example. Match any incident with a CIA value of less than 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} < 5`
<=	Finds all values less than or equal to the value specified. Click here for examples. Match any incident with a CIA value of less than or equal to 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} <= 5`
>	Finds all values greater than the value specified. Click here for an example. Match any incident with a CIA value of greater than 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1: ${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} > 5
>=	Finds all values greater than or equal to the value specified. Click here for an example. Match any incident with a Source Object's (interface speed) ifSpeed value of 10Mbps: `${ifSpeed} >= 10000000`
is not null	Finds all non-blank values. Click here for an example. Match any incident with a Source Object's (interface name) ifName attribute that contains a

Operator	Description
	value: `${ifName} is not null`
is null	Finds all blank values. Click here for an example. Match any incident with a Source Object's (interface name) ifName attribute that does not contain a value: `${ifName} is null`

like	Finds matches using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information.
	Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E .
	The period asterisk (.*) characters mean *any number of characters of any type at this location*.
	The period (.) character means *any single character of any type at this location*.
	Click here for an example.
	Match any incident with a Source Object's (interface description) ifDesc attribute that includes `Serial` followed by one or more digits:
	`${ifDesc} like Serial\d+`
	Match any incident with a Source Object's (interface alias) ifAlias attribute that contains `EtherChannel` (for example, `PAgPEtherChannel Group 1`).
	Note: The . (period) indicates any alphanumeric character.
	`${ifAlias} like .*EtherChannel.*`
	Match any incident with a CIA attribute value of `Chassis Fan Tray` followed by a digit and Object Identifier (OID) of `.1.3.6.1.4.1.9.9.13.1.4.1.3`
	Note: To include literal strings in the value, enclose the string value in \Q<*literal_value*>\E as shown in the following example.
	`${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.3\E)} like Chassis Fan Tray \d`

Operator	Description
not like	Finds all matches that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E . The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. Click here for an example. Match any incident with a Source Object's (interface name) ifName value that does not include `rtr`: `${ifName} not like .*rtr.*`

Valid Expressions

Attribute	Description
Expression	The value or pattern for which you want NNMi to search. Note the following: • The expression can include a valid Attribute. • The value or pattern you want to match is case sensitive. • When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**.

Configure a Correlation Filter

Note: See **Help → Documentation Library → Release Notes**, and locate the **Support Matrix** link for Correlation Filter limitations.

When correlating groups of incidents under a Parent incident, you must specify the Correlation Filter that defines the relationship requirements that must be met before the incidents are correlated. The Correlation Filter tab enables you to use the Filter Editor to define these relationship requirements. See Valid Operators in the table that follows for examples of valid Correlation Filters.

Use the Filter Editor Buttons to insert Boolean Operators and to append, insert, and replace expressions in the Filter String. Use the Drag and Drop feature to make changes to the placement of the expressions in your Filter String. Click here for more information about using the Filter Editor for Custom Correlations:

- You can use Custom Incident Attributes, attributes for an incident's Source Node or Source Object, or both to define how matching incidents should be considered for the Correlation Rule. See Valid Attributes for more information.
- When specifying Attribute names and values, NNMi uses the type to determine a match. For example, if the Attribute type is numeric, NNMi does a numeric comparison. If the Attribute type is textual, NNMi does a lexographical string comparison. In all cases, when you use the **like** or **not like** operator, NNMi uses a lexographical string comparison. Click here for more information about Attribute types:
  - `ifIndex` and `ifSpeed` are numeric Attributes.

  - Any Attribute name that begins wtih "is" (`isSnmpInterface`, `isSnmpNode`, `isNnmSystemLocal`) represents a Boolean Attribute.
  - All other Attributes are textual.
- Each set of expressions associated with a Boolean Operator (for example, `AND`) is treated as if it were enclosed in parentheses and evaluated together.
- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The `AND` and `OR` Boolean Operators must contain at least two expressions.
- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Additional Filters Editor. For example, you append to or replace, the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators. See "Add Boolean Operators in the Additional Filters Editor" (on page 193) for more information.

Filter Editor Buttons and Drag and Drop Feature

Button or Feature	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed Left or Right Expression.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Filter Editor deletes all expressions associated with the Boolean Operator.
Drag and Drop	You can drag any of the following items to a new location in the Filter String: ■ Filter Editor Options: AND, OR, NOT, EXISTS, NOT EXISTS ■ Filter Expression (Attribute, Operator and Value) When moving items in the Filter String, note the following: ■ Click the item you want to move before dragging it to a new location. ■ As you drag a selected item, an underline indicates the target location. ■ If you are moving the selection up, NNMi places the item above the target location.

Button or Feature	Description
	■ If you are moving the selection down, NNMi places the item below the target location. ■ If you attempt to move the selection to an invalid target location, NNMi displays an error message.

See "Correlation Rule Example" (on page 382) for a step-by-step example of how the Subinterface Custom Correlation Rule provided by NNMi was created.

To configure a Correlation Filter:

1. Navigate to the **Custom Correlation Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Select **Custom Correlation Configuration**.
2. Navigate to the **Correlation Rules** tab.
3. From the **Correlation Rules** table toolbar, do one of the following:
   - To create a Correlation Rule, click the New icon, and continue.
   - To edit a Correlation Rule, click the Open icon in the row representing the Correlation Rule you want to edit, and continue.
   - To delete a Correlation Rule, click the Delete icon.
4. Navigate to the **Correlation Filter** tab.
5. Create your Correlation Filter (see Filter Editor Components).

   Filter Editor Components

Component	Description
Attribute	The Attribute on which NNMi searches. See Valid Attributes below for a description of valid Attributes.
Operator	Use this Operator to establish the relationship between the Attribute and Expression. See Valid Operators in the table below for the description of each valid Operator.
Expression	Use the Expression to complete the criteria for the required relationship between the parent and child incident configurations. See Valid Expressions below for a description of the components that you might include in your Right Expression.

6. Click **Save and Close** to save your changes and return to the previous form.

Valid Attributes

Attribute	Description
Attribute	The Attribute on which NNMi searches. Valid attributes other than Source Node attributes depend on the Incident's Source Object. NNMi checks the Source Node as well as the Source Object for any Capability value. Note the following when specifying Attributes: • Boolean Attributes begin with "is" and must contain the value `true` or `false`.

Attribute	Description

- Use the following syntax to specify a Custom Incident Attribute (CIA):

  `valueOfCia(<CIA_Name>)`

  Note: Check the appropriate Incident form for any valid CIA Names provided by NNMi.

  For example: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} = 5`

- When specifying *CIA_Name* or *CA_Name* you can use the asterisk (*) as a wildcard character to specify one or more characters and a question mark (?) to specify a single character.
- If you use attributes that are valid for the Source Node, NNMi uses the Source Node when comparing values. If you use attributes that are valid for the Source Object, NNMi uses the Source Object when comparing values. You cannot use attributes that are valid for the Source Node and Source Object in the same filter.
- If the incident does not have a Source Object, a Source Node, or both (for example, the node is not stored in the NNMi database), you must use CIAs in your filter rather than Source Object or Source Node attributes.

  Tip: To check whether the Source Object or Source Node is stored in the NNMi database, open the incident and then select Open from the Lookup menu for the Source Node or Source Object displayed. If a form does not open for the selected object or node, this means the source is not stored in the NNMi database.
- When specifying a Correlation Filter, precede the attribute name with either `parent.` or `child.` to specify from which incident the attribute value should be compared. For example, you might specify `${parent.hostedOn}` or `${child.ifDesc}`.

Possible Source Object choices are as follows:

- Card [click here for a list of attribute values]

  Unique Keys from the Card Form: Capabilities Tab:

  - capability (Unique Key of the Capability)
- Interface [click here for a list of attribute values]

  Use the following syntax to specify a Custom Attribute (CA) for an Interface:

  `valueOfInterfaceCa(<CA_Name>)`

  For example: `${child.valueOfInterfaceCA(Role)} = WAN Connection`

  Values from the Basics Attributes listed on the Interface Form:

  - hostedOn (Hosted On Node)

    Note: You must use the full DNS name for the hostedOn value.

  Values from the Interface Form: General Tab:

  - ifName (name configured for the interface)
  - ifAlias (alias configured for the interface)
  - ifDescr (description configured for the interface)
  - ifIndex (index assigned to the interface)
  - ifSpeed (speed configured for the interface)

Attribute	Description

Note: When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**.

Addresses from the Interface Form: IP Addresses Tab:

- ipAddress (IP Address associated with the interface)

  Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=**.

Unique Keys from the Interface Form: Capabilities Tab:

- capability (Unique Key of the Capability)

Values from the Basics Attributes on the parent Node Form:

- isSnmpInterface (Agent Enabled)

Values from the parent Node Form: General Tab:

- sysOidInterface (System Object ID)

Values from the Basics Attributes on the associated Device Profile Form:

- devVendorInterface (Device Vendor)
- devFamilyInterface (Device Family)

- IP Address [click here for a list of attribute values]

  Unique Keys from the IP Address Form: Capabilities Tab:

  - capability (Unique Key of the Capability)

- Node [click here for a list of attribute values]

  Use the following syntax to specify a Custom Attribute (CA) for a Node:

  `valueOfNodeCa(<CA_Name>)`

  For example: `${valueOfNodeCA(Location)} = USA`

  Values from the Basics Attributes on the Node Form:

  - hostname (Hostname, *case-sensitive*)
  - mgmtIPAddress (Management Address)
  - isSnmpNode (Agent Enabled)
  - isNnmSystemLocal (NNMi Management Server)

  Values from the Node Form: General Tab:

  - sysName (System Name)
  - sysContact (System Contact)
  - sysLocation (System Location)
  - sysOidNode (System Object ID)

  Addresses from the Node Form: IP Addresses Tab:

  - hostedIPAddress (Address)

Attribute	Description
	Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=** . **Unique Keys from the Node Form: Capabilities Tab**: ■ capability (Unique Key of the Capability) **Values from the Basics Attributes on the associated Device Profile Form**: ■ devVendorNode (Device Vendor) ■ devFamillyNode (Device Family) **Values from the associated entry on the Regional Manager Form: Connection Tab**: ■ nnmSystemName (Hostname, *case-sensitive*) (*NNMi Advanced*) If the Global Network Management feature is enabled, this attribute value identifies a Regional Manager (NNMi management server).

Valid Operators

Operator	Description
=	Finds all values equal to the value specified. Click here for an example. Correlate the incidents if the `hostedOn` value for the Source Object of the Child Incident is equal to the `hostedOn` value for the Source Object in the Parent Incident. `${child.hostedOn} = ${parent.hostedOn}`
!=	Finds all values not equal to the value specified. Click here for an example. Correlate the incidents if the `hostedOn` value for the Source Object of the Child Incident is not equal to the `hostedOn` value for the Source Object in the Parent Incident. `${child.hostedOn} != ${parent.hostedOn}`
<	Finds all values less than the value specified.
<=	Finds all values less than or equal to the value specified.
>	Finds all values greater than the value specified.
>=	Finds all values greater than or equal to the value specified.
is not null	Finds all non-blank values.
is null	Finds all blank values.
like	Finds matches using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E . The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*.
not like	Finds all matches that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E . The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*.

Valid Expressions

Attribute	Description
Expression	The value or pattern for which you want NNMi to search.

Attribute	Description
	Note the following: • The expression can include a valid Attribute. • The value or pattern you want to match is case sensitive. • When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**.

Correlation Rule Example

Tip: Use these steps as a guideline for creating your own Correlation Rules.

This example uses the Subinterface Correlation Rule to describe the steps for creating a Correlation Rule. The Subinterface Correlation Rule specifies that Interface Down incidents that occur for subinterfaces should be correlated under the Interface Down incident generated for the main interface. Click here for more information about Custom Correlations.

The NNMi Custom Correlation feature enables you to correlate groups of incidents under a Parent Incident. This feature is useful when you want to define a relationship between a number of incidents potentially from different network objects that form a logical set to identify a problem. The set of correlations is considered complete if all of the incidents arrive within a specified time window. You can correlate incidents under an existing Incident Configuration (Correlation Rule) or create a new Incident Configuration (Causal Rule).

This example uses an existing Incident Configuration as the Parent Incident. See "Causal Rule Example" (on page 410) for an example that generates a new Incident Configuration as the Parent Incident.

To configure the Subinterface Correlation Rule Basics information:

1. Navigate to the **Custom Correlation Configuration** form:
    a. From the workspace navigation pane, select the **Configuration** workspace.
    b. Select **Custom Correlation Configuration**.
2. Navigate to the **Correlation Rules** tab.
3. From the **Correlation Rules** table toolbar, click the New icon.
4. In the **Name** attribute, enter a unique name that will help you to identify the Correlation Rule. In this example, the Correlation Rule Name is **Subinterface**.
5. In the **Author** attribute, enter a name that identifies the person who is creating the Correlation Rule. In this example, **HP Network Node Manager** is the Author name to identify this Correlation Rule as one that NNMi provides.
6. Make sure **Enabled** is checked to indicate the NNMi Causal Engine should use this Correlation Rule when evaluating incidents.
7. To use an existing Parent Incident, do the following:
    a. In the **Parent Incident** Lookup Field, select Quick Find to select from the list of existing incident configurations.
    b. Click the **Select This Item** icon that precedes the incident configuration that must match an incoming incident and that should be correlated as the Parent Incident for the Custom

Correlation.

In the Subinterface Correlation Rule, the **InterfaceDown** incident configuration was selected as the Parent Incident.

8. Select the Incident Configuration that must match an incoming incident and that should be correlated as the Child Incident for the Custom Correlation.

   In the Subinterface Correlation Rule, the **InterfaceDown** incident configuration was also selected as the Child Incident.

9. In the **Correlation Duration Window** attribute, enter the time limit (in days, hours, minutes, and seconds) that must be reached before the incoming incident are correlated. The Subinterface Correlation Rule specifies a Correlation Window Duration of 6 minutes.

10. Use the **Description** attribute to provide additional information that you would like to store about the current incident configuration. This description applies only to the configuration entry.

    The Subinterface Correlation Rule includes the following description: **Correlates sub-interfaces down incidents under the main interface down**.

To configure the Parent Incident Filter:

1. In the Correlation Rule form, navigate to the **Parent Incident Filter** tab.
2. The following Parent Incident Filter specifies that the Correlation Rule applies only to Cisco devices:

   ```
   ${devVendorInterface} = Cisco
   ```

3. The following Parent Incident Filter specifies that the ifDesc value must contain the string `Serial` followed by one or more digits and then a forward slash, followed by zero or more digits:

   ```
   ${ifDesc}like Serial\d+/*\d*
   ```

4. As shown in the following Filter String, the Parent Incident Filters use the Boolean operator AND so that both criteria must be met for the Incident to be selected as a Parent:

   ```
   {${devVendorInterface} = Cisco AND ${ifDesc} like Serial\d+/*\d*)
   ```

   To create this Parent Incident Filter, in the Filter Editor:

   a. Click **And**.
   b. In the **Attribute** field, enter `${devVendorInterface}`.
   c. In the **Operator** field, select **=** from the drop-down menu.
   d. In the **Expression** field, enter **Cisco**.
   e. Click **Append**.
   f. In the **Attribute** field, enter `${ifDesc}`.
   g. In the **Operator** field, select `like` from the drop-down menu.
   h. In the **Expression** field, enter `Serial\d+/*\d*`.
   i. Click **Add**.

To configure the Child Incident Filter:

1. In the Correlation Rule form, navigate to the **Child Incident Filter** tab.
2. The following Child Incident Filter specifies that the Correlation Rule applies only to Cisco devices:

   ```
   ${devVendorInterface} = Cisco
   ```

3. The following Child Incident Filter specifies that the ifDesc value must contain the following sequence of values:

   The string `Serial` followed by one or more digits, then a forward slash, followed by zero or more digits, and then a period followed by one or more digits:

   ```
   ${ifDesc}like Serial\d+/*\d*
   ```

4. As shown in the following Filter String, the Child Incident Filters use the Boolean operator AND so that both criteria must be met for the Incident to be selected as a Child:

   ```
   {${devVendorInterface} = Cisco AND ${ifDesc} like Serial\d+/*\d*)
   ```

   To create this Parent Incident Filter, in the Filter Editor:

   a. Click **And**.

   b. In the **Attribute** field, enter `${devVendorInterface}`.

   c. In the **Operator** field, select **=** from the drop-down menu.

   d. In the **Expression** field, enter **Cisco**.

   e. Click **Append**.

   f. In the **Attribute** field, enter `${ifDesc}`.

   g. In the **Operator** field, select `like` from the drop-down menu.

   h. In the **Expression** field, enter `Serial\d+/*\d*`.

   i. Click **Append**.

To configure the Correlation Filter:

Note: When specifying a Correlation Filter, you must specify whether the attribute is from a Child Incident or Parent Incident using the following syntax: `${child.<`*attribute_name*`>}` or `${parent.<`*attribute_name*`>}`.

1. In the Correlation Rule form, navigate to the **Correlation Filter** tab.

2. To ensure that the Interface Down incidents are generated for the same node, the Subinterface Correlation Rules uses `hostedOn` as the attribute for both the Child and Parent Incidents as shown in the following example filter:

   ```
   ${child.hostedOn}= ${parent.hostedOn}
   ```

   To ensure that the Interfaces are subinterfaces for the main interface, the filter also matches the ifDesc values:

   ```
   ${child.ifDesc}like ${parent.ifDesc}.*
   ```

   As shown in the following Filter String, the Correlation Filter uses the Boolean operator AND so that both criteria must be met for the Incidents to be correlated:

   ```
   ${child.hostedOn} = ${parent.hostedOn}AND ${child.ifDesc} like
   ${parent.ifDesc}.*
   ```

   To create the Correlation Rule filter:

   a. Click **And**.

   b. In the **Attribute** field, enter `${child.hostedOn}`.

   c. In the **Operator** field, select **=** from the drop-down menu.

d. In the **Expression** field, enter ${parent.hostedOn}.

e. Click **Append**.

f. In the **Attribute** field, enter `${child.ifDesc}`.

g. In the **Operator** field, select `like` from the drop-down menu.

h. In the **Expression** field, enter `${parent.ifDesc}.*`.

i. Click **Append**.

3. Click **Save and Close** to save your changes and return to the previous form.

Configure a Causal Rule

Tip: Configure a Causal Rule when you want to cause NNMi to generate a Parent Incident and you want to correlate one or more Child Incident Configurations under the Parent Incident that you cause to be generated.

Note: See **Help → Documentation Library → Release Notes**, and locate the **Support Matrix** link for Causal Rule limitations.

When correlating groups of incidents under a Parent incident, use the Causal Rules tab to specify the following.

- Parent Incident Configuration to be generated
- One or more Child Incident Configurations to be correlated with the generated Parent Incident
- Filters that NNMi should use when selecting the Child Incident instances for correlation
- Source Object and Source Node Filter to be used to determine the Source Node and Source Object for the Parent Incident that is generated
- The time window that must be met before NNMi correlates the incidents

For information about each Causal Rules tab:

To configure a Causal Rule:

1. Navigate to the **Custom Correlation Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Select **Custom Correlation Configuration**.
2. Navigate to the **Causal Rules** tab.
3. From the **Causal Rules** table toolbar, do one of the following:
   - To create a Causal Rule, click the New icon, and continue.
   - To edit a Causal Rule, click the Open icon in the row representing the Causal Rule you want to edit, and continue.
   - To delete a Causal Rule, click the Delete icon.
4. Create your Causal Rule (see table).
5. Click **Save and Close** to save your changes and return to the previous form.

Cause Rule Basic Attributes

Attribute	Description
Name	Type a maximum of 64 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. The name is used to identify the Causal Rule and must be unique. Use a name that will help you to remember the purpose of the Causal Rule.
Author	Indicates who created or last modified the Causal Rule. See Author form for important information. **Caution**: If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. Click the Lookup icon and select Quick Find to access the list of existing Author values or click New to create one.
Enabled	If enabled, the NNMi Causal Engine uses the Causal Rule when evaluating incidents. If disabled, the Causal Rule is ignored.
Parent Incident	Specifies the incident configuration that should be generated as the Parent Incident for the Causal Rule. **To specify a Parent Incident configuration**: 1. Click the Lookup icon, and do one of the following: ▪ To specify a Parent Incident without making any changes to the Incident configuration, select Quick Find . In the Quick Find dialog, select the Incident of interest. ▪ To create a Parent Incident, select one of the following: ○ New Management Event Configuration ○ New Remote NNM Event Configuration ○ New SNMP Trap Configuration ▪ To modify a Parent Incident, select Open. 2. *Optional*. To create or modify a Parent Incident, enter or modify the attribute values for the selected Incident configuration. See "Configuring Incidents" (on page 306) for more information about the Incident Configuration form. 3. Click **Save and Close** to save your changes and return to the previous form.
Correlation Nature	Select the Correlation Nature that you want to assign to the Parent Incident that is generated. **Note**: The Child Incident will have the Correlation Nature of Secondary Root Cause.
Common Child Incident Attribute	Specifies the Incident Attribute that all Child Incidents must have in common for the incident instance to be correlated under the Parent Incident defined for the Causal Rule. **Note**: You cannot specify $(capability) as a Common Child Incident Attribute.

Attribute	Description
Correlation Window Duration	The time window that must be met before NNMi correlates the incidents. Enter a number for Days, Hours, Minutes, and Seconds. Note the following: • NNMi waits until the Correlation Window Duration has passed before generating the Parent Incident and correlating its Child Incidents. • If you are relating multiple Custom Correlations, make sure the Correlation Window Duration allows enough time for all of the Parent and Child incidents to be generated.
Description	Use the description field to provide additional information that you would like to store about the current incident configuration. This description applies only to the configuration entry. Type a maximum of 2048 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Configure a Child Incident for a Causal Rule

The Child Incident tab enables you to specify which Child Incidents should be considered for correlation according to the Causal Rule you are configuring.

For information about each Causal Rules tab:

For information about each Child Incident tab:

To configure a Child Incident for a Causal Rule:

1. Navigate to the **Custom Correlation Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Select **Custom Correlation Configuration**.
2. Navigate to the **Causal Rules** tab.
3. From the **Causal Rules** table toolbar, do one of the following:
   - To create a Causal Rule, click the New icon, and continue.
   - To edit a Causal Rule, click the Open icon in the row representing the Causal Rule you want to edit, and continue.
   - To delete a Causal Rule, click the Delete icon.
4. Create your Causal Rule. (See "Configure a Causal Rule" (on page 385).)
5. Create your Child Incident Configuration (see table).
6. *Optional*. Configure a Child Incident Filter. (See "Configure a Child Incident Filter for a Causal Rule" (on page 389).)
7. *Optional*. Configure a Source Object Filter. (See "Configure a Source Object Filter for a Causal Rule" (on page 397).)
8. *Optional*. Configure a Source Node Filter. (See "Configure a Source Node Filter for a Causal Rule" (on page 404).)
9. Click **Save and Close** to save your changes and return to the previous form.

Causal Rule Basic Attributes

Attribute	Description
Name	Type a maximum of 64 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. The name is used to identify the Child Incident Configuration and must be unique within each Causal Rule. Use a name that will help you to remember the purpose of the Child Incident Configuration.
Child Incident	Specifies the incident configuration that should be used as the Child Incident when evaluating the Causal Rule. **To specify a Child Incident configuration**: 1. Click the Lookup icon, and do one of the following: ▪ To specify a Child Incident without making any changes to the incident configuration, select Quick Find . In the Quick Find dialog, select the Incident of interest. ▪ To create a Child Incident, select one of the following: ○ New Management Event Configuration ○ New Remote NNM Event Configuration ○ New SNMP Trap Configuration ▪ To modify a Child Incident, select Open. 2. *Optional*. To create or modify a Child Incident, enter or modify the attribute values for the selected Incident configuration. See "Configuring Incidents" (on page 306) for more information about the Incident Configuration form. 3. Click **Save and Close** to save your changes and return to the previous form.
Forward Child Custom Incident Attributes	Enter a comma-delimited list of the Custom Incident Attributes you want to appear with the generated Parent Incident. NNMi forwards these values from the Child Incidents that you configure for the Causal Rule.
Optional Child Incident	If ☑ enabled, the NNMi Causal Engine generates the Parent Incident whether this Child Incident occurs. If ☐ disabled, the NNMi Causal Engine only generates the Parent Incident if this Child Incident occurs.
Use Child Incident's Source Object for Parent	If ☑ enabled, indicates you want NNMi to use the Source Object of the Child Incident as the Source Object for the Parent Incident. **Note**: If you enable this option, NNMi ignores any Source Object Filter you configured. If ☐ disabled, indicates you want NNMi to use the Source Object Filter configuration to determine the Parent Incident's Source Node. See "Configure a Source Object Filter for a Causal Rule" (on page 397) for more information. If you do not specify the Source Object to use for the Parent Incident, NNMi uses the Source Object of the first Child Incident that occurs.

Attribute	Description
Use Child Incident's Source Node for Parent	If ☑ enabled, indicates you want NNMi to use the Source Node of the Child Incident as the Source Node for the Parent Incident. **Note**: If you enable this option, NNMi ignores any Source Node Filter you configured. If ☐ disabled, indicates you want NNMi to use the Source Node Filter configuration to determine the Parent Incident's Source Node. See "Configure a Source Node Filter for a Causal Rule" (on page 404) for more information. If you do not specify the Source Node to use for the Parent Incident, NNMi uses the Source Node of the first Child Incident that occurs.

Configure a Child Incident Filter for a Causal Rule

Note: See **Help** → **Documentation Library** → **Release Notes**, and locate the **Support Matrix** link for Child Incident Filter limitations.

The Child Incident Filter tab enables you to create a filter to specify which Child Incidents should be considered for correlation according to the Causal Rule you are configuring.

For information about each Causal Rules tab:

For information about each Child Incident tab:

Use the Filter Editor Buttons to insert Boolean Operators and to append, insert, and replace expressions in the Filter String. Use the Drag and Drop feature to make changes to the placement of the expressions in your Filter String. Click here for more information about using the Filter Editor for Custom Correlations:

- You can use Custom Incident Attributes, attributes for an incident's Source Node or Source Object, or both to define how matching incidents should be considered for the Correlation Rule. See Valid Attributes for more information.
- When specifying Attribute names and values, NNMi uses the type to determine a match. For example, if the Attribute type is numeric, NNMi does a numeric comparison. If the Attribute type is textual, NNMi does a lexographical string comparison. In all cases, when you use the **like** or **not like** operator, NNMi uses a lexographical string comparison. Click here for more information about Attribute types:
  - `ifIndex` and `ifSpeed` are numeric Attributes.
  - Any Attribute name that begins wtih "is" (`isSnmpInterface`, `isSnmpNode`, `isNnmSystemLocal`) represents a Boolean Attribute.
  - All other Attributes are textual.
- Each set of expressions associated with a Boolean Operator (for example, `AND`) is treated as if it were enclosed in parentheses and evaluated together.
- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The `AND` and `OR` Boolean Operators must contain at least two expressions.
- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Additional Filters Editor. For example, you append to or replace, the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators. See "Add Boolean Operators in the Additional Filters Editor" (on page 193) for more information.

Filter Editor Buttons and Drag and Drop Feature

Button or Feature	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed Left or Right Expression.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Filter Editor deletes all expressions associated with the Boolean Operator.
Drag and Drop	You can drag any of the following items to a new location in the Filter String: ■ Filter Editor Options: AND, OR, NOT, EXISTS, NOT EXISTS ■ Filter Expression (Attribute, Operator and Value) When moving items in the Filter String, note the following: ■ Click the item you want to move before dragging it to a new location. ■ As you drag a selected item, an underline indicates the target location. ■ If you are moving the selection up, NNMi places the item above the target location. ■ If you are moving the selection down, NNMi places the item below the target location. ■ If you attempt to move the selection to an invalid target location, NNMi displays an error message.

To configure a Child Incident Filter for a Causal Rule:

1. Navigate to the **Custom Correlation Configuration** form:
    a. From the workspace navigation pane, select the **Configuration** workspace.
    b. Select **Custom Correlation Configuration**.
2. Navigate to the **Causal Rules** tab.
3. From the **Causal Rules** table toolbar, do one of the following:

- To create a Causal Rule, click the New icon, and continue.
- To edit a Causal Rule, click the Open icon in the row representing the Causal Rule you want to edit, and continue.
- To delete a Causal Rule, click the Delete icon.

4. Create your Causal Rule. (See "Configure a Causal Rule" (on page 385).)
5. Create your Child Incident Configuration . (See "Configure a Child Incident for a Causal Rule" (on page 387).)
6. *Optional*. Configure a Child Incident Filter. (See Filter Editor Components).

Filter Editor Components

Component	Description
Attribute	The Attribute on which NNMi searches. See Valid Attributes below for a description of valid Attributes.
Operator	Use this Operator to establish the relationship between the Attribute and Expression. See Valid Operators in the table below for the description of each valid Operator.
Expression	Use the Expression to complete the criteria for the Parent Incident configuration. See Valid Expressions below for a description of the components that you might include in your Expression.

7. *Optional*. Configure a Source Object Filter. (See "Configure a Source Object Filter for a Causal Rule" (on page 397).)
8. *Optional*. Configure a Source Node Filter. (See "Configure a Source Node Filter for a Causal Rule" (on page 404).)
9. Click **Save and Close** to save your changes and return to the previous form.

Valid Attributes

Attribute	Description
Attribute	The Attribute on which NNMi searches. Valid attributes other than Source Node attributes depend on the Incident's Source Object. NNMi checks the Source Node as well as the Source Object for any Capability value. Note the following when specifying Attributes: • Boolean Attributes begin with "is" and must contain the value `true` or `false`. • Use the following syntax to specify a Custom Incident Attribute (CIA): `valueOfCia(<CIA_Name>)` **Note**: Check the appropriate Incident form for any valid CIA Names provided by NNMi. For example: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} = 5` • When specifying *CIA_Name* or *CA_Name* you can use the asterisk (*) as a wildcard character to specify one or more characters and a question mark (?) to specify a single character. • If you use attributes that are valid for the Source Node, NNMi uses the Source Node when comparing values. If you use attributes that are valid for the Source Object, NNMi uses the Source Object when comparing values. You cannot use attributes that are valid for the

Attribute	Description

Source Node and Source Object in the same filter.

- If the incident does not have a Source Object, a Source Node, or both (for example, the node is not stored in the NNMi database), you must use CIAs in your filter rather than Source Object or Source Node attributes.

  Tip: To check whether the Source Object or Source Node is stored in the NNMi database, open the incident and then select Open from the Lookup menu for the Source Node or Source Object displayed. If a form does not open for the selected object or node, this means the source is not stored in the NNMi database.

- When specifying a Correlation Filter, precede the attribute name with either `parent.` or `child.` to specify from which incident the attribute value should be compared. For example, you might specify `${parent.hostedOn}` or `${child.ifDesc}`.

Possible Source Object choices are as follows:

- Card [click here for a list of attribute values]

  Unique Keys from the Card Form: Capabilities Tab:

  - capability (Unique Key of the Capability)

- Interface [click here for a list of attribute values]

  Use the following syntax to specify a Custom Attribute (CA) for an Interface:

  `valueOfInterfaceCa(<CA_Name>)`

  For example: `${child.valueOfInterfaceCA(Role)} = WAN Connection`

  Values from the Basics Attributes listed on the Interface Form:

  - hostedOn (Hosted On Node)

    Note: You must use the full DNS name for the hostedOn value.

  Values from the Interface Form: General Tab:

  - ifName (name configured for the interface)
  - ifAlias (alias configured for the interface)
  - ifDescr (description configured for the interface)
  - ifIndex (index assigned to the interface)
  - ifSpeed (speed configured for the interface)

    Note: When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**.

  Addresses from the Interface Form: IP Addresses Tab:

  - ipAddress (IP Address associated with the interface)

    Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=**.

  Unique Keys from the Interface Form: Capabilities Tab:

  - capability (Unique Key of the Capability)

Attribute	Description
	Values from the Basics Attributes on the parent Node Form: ■ isSnmpInterface (Agent Enabled) **Values from the parent Node Form: General Tab**: ■ sysOidInterface (System Object ID) **Values from the Basics Attributes on the associated Device Profile Form**: ■ devVendorInterface (Device Vendor) ■ devFamilyInterface (Device Family) • IP Address [click here for a list of attribute values] **Unique Keys from the IP Address Form: Capabilities Tab**: ■ capability (Unique Key of the Capability) • Node [click here for a list of attribute values] Use the following syntax to specify a Custom Attribute (CA) for a Node: `valueOfNodeCa(<CA_Name>)` For example: `${valueOfNodeCA(Location)} = USA` **Values from the Basics Attributes on the Node Form**: ■ hostname (Hostname, *case-sensitive*) ■ mgmtIPAddress (Management Address) ■ isSnmpNode (Agent Enabled) ■ isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: ■ sysName (System Name) ■ sysContact (System Contact) ■ sysLocation (System Location) ■ sysOidNode (System Object ID) **Addresses from the Node Form: IP Addresses Tab**: ■ hostedIPAddress (Address) Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=** . **Unique Keys from the Node Form: Capabilities Tab**: ■ capability (Unique Key of the Capability) **Values from the Basics Attributes on the associated Device Profile Form**: ■ devVendorNode (Device Vendor) ■ devFamillyNode (Device Family) **Values from the associated entry on the Regional Manager Form: Connection Tab**:

Attribute	Description
	■ nnmSystemName (Hostname, *case-sensitive*) (*NNMi Advanced*) If the Global Network Management feature is enabled, this attribute value identifies a Regional Manager (NNMi management server).

Attribute	Description

Valid Operator Values

Operator	Description

=	Finds all values equal to the value specified.
	Click here for examples.
	Match any incident with a CIA value of 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1:
	`${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} = 5`
	Match any incident with the Source Object's Capability equal to `com.hp.nnm.capability.card.fru`
	`$(capability) = com.hp.nnm.capability.card.fru`

!=	Finds all values not equal to the value specified. Click here for an example. Match any incident with Device Vendor for the interface (Source Object) not equal to `Cisco`: `${devVendorInterface} != Cisco`
<	Finds all values less than the value specified. Click here for an example. Match any incident with a CIA value of less than 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} < 5`
<=	Finds all values less than or equal to the value specified. Click here for examples. Match any incident with a CIA value of less than or equal to 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} <= 5`
>	Finds all values greater than the value specified. Click here for an example. Match any incident with a CIA value of greater than 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1: ${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} > 5
>=	Finds all values greater than or equal to the value specified. Click here for an example. Match any incident with a Source Object's (interface speed) ifSpeed value of 10Mbps: `${ifSpeed} >= 10000000`
is not null	Finds all non-blank values. Click here for an example. Match any incident with a Source Object's (interface name) ifName attribute that contains a

Operator	Description
	value: `${ifName} is not null`
is null	Finds all blank values. Click here for an example. Match any incident with a Source Object's (interface name) ifName attribute that does not contain a value: `${ifName} is null`

like	Finds matches using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information.
	Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E .
	The period asterisk (.*) characters mean *any number of characters of any type at this location*.
	The period (.) character means *any single character of any type at this location*.
	Click here for an example.
	Match any incident with a Source Object's (interface description) ifDesc attribute that includes `Serial` followed by one or more digits:
	`${ifDesc} like Serial\d+`
	Match any incident with a Source Object's (interface alias) ifAlias attribute that contains `EtherChannel` (for example, `PAgPEtherChannel Group 1`).
	Note: The . (period) indicates any alphanumeric character.
	`${ifAlias} like .*EtherChannel.*`
	Match any incident with a CIA attribute value of `Chassis Fan Tray` followed by a digit and Object Identifier (OID) of `.1.3.6.1.4.1.9.9.13.1.4.1.3`
	Note: To include literal strings in the value, enclose the string value in \Q<*literal_value*>\E as shown in the following example.
	`${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.3\E)} like Chassis Fan Tray \d`

Operator	Description
not like	Finds all matches that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E . The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. Click here for an example. Match any incident with a Source Object's (interface name) ifName value that does not include `rtr`: `${ifName} not like .*rtr.*`

Valid Expressions

Attribute	Description
Expression	The value or pattern for which you want NNMi to search. Note the following: • The expression can include a valid Attribute. • The value or pattern you want to match is case sensitive. • When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**.

Configure a Source Object Filter for a Causal Rule

The Source Filter tab enables you to create a filter to specify which Source Object should be used for the Parent Incident that is generated for this Causal Rule.

Note: Create only one Source Object Filter for a Causal Rule. If you select **Use Child Incident's Source Object for Parent** ☑, NNMi ignores any Source Object Filter you configure.

For information about each Causal Rules tab:

For information about each Child Incident tab:

Use the Filter Editor Buttons to insert Boolean Operators and to append, insert, and replace expressions in the Filter String. Use the Drag and Drop feature to make changes to the placement of the expressions in your Filter String. Click here for more information about using the Filter Editor for Custom Correlations:

- You can use Custom Incident Attributes, attributes for an incident's Source Node or Source Object, or both to define how matching incidents should be considered for the Correlation Rule. See Valid Attributes for more information.
- When specifying Attribute names and values, NNMi uses the type to determine a match. For example, if the Attribute type is numeric, NNMi does a numeric comparison. If the Attribute type is textual, NNMi does a lexographical string comparison. In all cases, when you use the **like** or **not like** operator, NNMi uses a lexographical string comparison. Click here for more information about Attribute types:
  - `ifIndex` and `ifSpeed` are numeric Attributes.

  - Any Attribute name that begins wtih "is" (`isSnmpInterface`, `isSnmpNode`, `isNnmSystemLocal`) represents a Boolean Attribute.
  - All other Attributes are textual.
- Each set of expressions associated with a Boolean Operator (for example, `AND`) is treated as if it were enclosed in parentheses and evaluated together.
- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The `AND` and `OR` Boolean Operators must contain at least two expressions.
- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Additional Filters Editor. For example, you append to or replace, the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators. See "Add Boolean Operators in the Additional Filters Editor" (on page 193) for more information.

Filter Editor Buttons and Drag and Drop Feature

Button or Feature	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed Left or Right Expression.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Filter Editor deletes all expressions associated with the Boolean Operator.
Drag and Drop	You can drag any of the following items to a new location in the Filter String: ■ Filter Editor Options: AND, OR, NOT, EXISTS, NOT EXISTS ■ Filter Expression (Attribute, Operator and Value) When moving items in the Filter String, note the following: ■ Click the item you want to move before dragging it to a new location. ■ As you drag a selected item, an underline indicates the target location. ■ If you are moving the selection up, NNMi places the item above the target location.

Button or Feature	Description
	■ If you are moving the selection down, NNMi places the item below the target location. ■ If you attempt to move the selection to an invalid target location, NNMi displays an error message.

To configure a Source Object Filter for a Causal Rule:

1. Navigate to the **Custom Correlation Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Select **Custom Correlation Configuration**.
2. Navigate to the **Causal Rules** tab.
3. From the **Causal Rules** table toolbar, do one of the following:
   - To create a Causal Rule, click the New icon, and continue.
   - To edit a Causal Rule, click the Open icon in the row representing the Causal Rule you want to edit, and continue.
   - To delete a Causal Rule, click the Delete icon.
4. Create your Causal Rule. (See "Configure a Causal Rule" (on page 385).)
5. Create your Child Incident Configuration . (See "Configure a Child Incident for a Causal Rule" (on page 387).)
6. *Optional*. Configure a Child Incident Filter. (See "Configure a Child Incident Filter for a Causal Rule" (on page 389).)
7. *Optional*. Configure a Source Object Filter. (See the tables that follow, starting with Filter Editor Components).

   Filter Editor Components

Component	Description
Attribute	The Attribute on which NNMi searches. See Valid Attributes below for a description of valid Attributes.
Operator	Use this Operator to establish the relationship between the Attribute and Expression. See Valid Operators in the table below for the description of each valid Operator.
Expression	Use the Expression to complete the criteria for the Parent Incident configuration. See Valid Expressions below for a description of the components that you might include in your Expression.

8. *Optional*. Configure a Source Node Filter. (See "Configure a Source Node Filter for a Causal Rule" (on page 404).)
9. Click **Save and Close** to save your changes and return to the previous form.

Valid Attributes

Attribute	Description

Attribute	The Attribute on which NNMi searches. Valid attributes other than Source Node attributes depend on the Incident's Source Object. NNMi checks the Source Node as well as the Source Object for any Capability value.
	
	Note the following when specifying Attributes:
	
	• Boolean Attributes begin with "is" and must contain the value `true` or `false`.
	
	• Use the following syntax to specify a Custom Incident Attribute (CIA):
	
	`valueOfCia(<CIA_Name>)`
	
	Note: Check the appropriate Incident form for any valid CIA Names provided by NNMi.
	
	For example: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} = 5`
	
	• When specifying *CIA_Name* or *CA_Name* you can use the asterisk (*) as a wildcard character to specify one or more characters and a question mark (?) to specify a single character.
	
	• If you use attributes that are valid for the Source Node, NNMi uses the Source Node when comparing values. If you use attributes that are valid for the Source Object, NNMi uses the Source Object when comparing values. You cannot use attributes that are valid for the Source Node and Source Object in the same filter.
	
	• Interface [click here for a list of attribute values]
	
	Use the following syntax to specify a Custom Attribute (CA) for an Interface:
	
	`valueOfInterfaceCa(<CA_Name>)`
	
	For example: `${child.valueOfInterfaceCA(Role)} = WAN Connection`
	
	Values from the Basics Attributes listed on the Interface Form:
	
	▪ hostedOn (Hosted On Node)
	
	Note: You must use the full DNS name for the hostedOn value.
	
	Values from the Interface Form: General Tab:
	
	▪ ifName (name configured for the interface)
	
	▪ ifAlias (alias configured for the interface)
	
	▪ ifDescr (description configured for the interface)
	
	▪ ifIndex (index assigned to the interface)
	
	▪ ifSpeed (speed configured for the interface)
	
	Note: When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**.
	
	Addresses from the Interface Form: IP Addresses Tab:
	
	▪ ipAddress (IP Address associated with the interface)
	
	Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=** .

Attribute	Description

Unique Keys from the Interface Form: Capabilities Tab:

- capability (Unique Key of the Capability)

Values from the Basics Attributes on the parent Node Form:

- isSnmpInterface (Agent Enabled)

Values from the parent Node Form: General Tab:

- sysOidInterface (System Object ID)

Values from the Basics Attributes on the associated Device Profile Form:

- devVendorInterface (Device Vendor)
- devFamilyInterface (Device Family)

- IP Address [click here for a list of attribute values]

  Unique Keys from the IP Address Form: Capabilities Tab:

  - capability (Unique Key of the Capability)

- Node [click here for a list of attribute values]

  Use the following syntax to specify a Custom Attribute (CA) for a Node:

  `valueOfNodeCa(<CA_Name>)`

  For example: `${valueOfNodeCA(Location)} = USA`

  Values from the Basics Attributes on the Node Form:

  - hostname (Hostname, *case-sensitive*)
  - mgmtIPAddress (Management Address)
  - isSnmpNode (Agent Enabled)
  - isNnmSystemLocal (NNMi Management Server)

  Values from the Node Form: General Tab:

  - sysName (System Name)
  - sysContact (System Contact)
  - sysLocation (System Location)
  - sysOidNode (System Object ID)

  Addresses from the Node Form: IP Addresses Tab:

  - hostedIPAddress (Address)

    Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=** .

  Unique Keys from the Node Form: Capabilities Tab:

  - capability (Unique Key of the Capability)

  Values from the Basics Attributes on the associated Device Profile Form:

Attribute	Description
	■ devVendorNode (Device Vendor) ■ devFamillyNode (Device Family) **Values from the associated entry on the Regional Manager Form: Connection Tab**: ■ nnmSystemName (Hostname, *case-sensitive*) (*NNMi Advanced*) If the Global Network Management feature is enabled, this attribute value identifies a Regional Manager (NNMi management server).

Attribute	Description

Valid Operator Values

Operator	Description
=	Finds all values equal to the value specified. Click here for examples. Match any incident with a CIA value of 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} = 5`
!=	Finds all values not equal to the value specified. Click here for an example. Match any incident with a Source Object value of Interface with Device Vendor value not equal to `Cisco`: `${devVendorInterface} != Cisco`
<	Finds all values less than the value specified. Click here for an example. Match any incident with a CIA value less than 5 and Object Identifier (OID) value of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} < 5`
<=	Finds all values less than or equal to the value specified. Click here for examples. Match any incident with a CIA attribute value less than or equal to 5 and Object Identifier (OID) value of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} <= 5`
>	Finds all values greater than the value specified. Click here for an example. Match any incident with a CIA value greater than 5 and Object Identifier (OID) attribute value of .1.3.6.1.4.1.9.9.106.2.0.1: ${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} > 5
>=	Finds all values greater than or equal to the value specified. Click here for an example. Match any incident with a Source Object attribute value of Interface that has an (interface speed) ifSpeed of 10Mbps: `${ifSpeed} >= 10000000`
is not null	Finds all non-blank values. Click here for an example. Match any incident with a Source Object attribute value of Interface that has an (interface name) ifName value: `${ifName} is not null`

Operator	Description
is null	Finds all blank values. Click here for an example. Match any incident with a Source Object attribute value of Interface that does not have an (interface name) ifName value: `${ifName} is null`

like	Finds matches using wildcard characters and the question mark.
	The asterisk (*) character means *any number of characters of any type at this location*.
	The question mark (?) character means *any single character of any type at this location*.
	Click here for an example.
	Match any incident with a Source Object attribute value of Interface with a (description) ifDesc that begins with `Serial` followed by any number of characters:
	`${ifDesc} like Serial*`
	Match any incident with a Source Object attribute value of Interface with an (interface alias) ifAlias value that begins with `EtherChannel` (for example, `EtherChannel Group 1`).
	`${ifAlias} like EtherChannel*`

not like	Finds all matches that do not have the values specified. The asterisk (*) characters means *any number of characters of any type at this location*. The question mark (?) character means *any single character of any type at this location*. Click here for an example. Match any with a Source Object attribute value of Interface with an (interface name) ifName value that does not begin with `rtr*`: `${ifName} not like rtr*`

Valid Expressions

Attribute	Description
Expression	The value or pattern for which you want NNMi to search. Note the following: • The expression can include a valid Attribute. • The value or pattern you want to match is case sensitive. • When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**.

Configure a Source Node Filter for a Causal Rule

The Source Node Filter tab enables you to create a filter to specify which Source Node should be used for the Parent Incident that is generated for this Causal Rule.

Note: Create only one Source Node Filter for a Causal Rule. If you select **Use Child Incident's Source Node for Parent**☑, NNMi ignores any Source Node Filter you configure.

For information about each Causal Rules tab:

For information about each Child Incident tab:

Use the Filter Editor Buttons to insert Boolean Operators and to append, insert, and replace expressions in the Filter String. Use the Drag and Drop feature to make changes to the placement of the expressions in your Filter String. Click here for more information about using the Filter Editor:

- You can use Custom Incident Attributes, attributes for an incident's Source Node, or both to define how matching incidents should be considered for the Causal Rule. See Valid Attributes for more information.
- When specifying Attribute names and values, NNMi uses the type to determine a match. For example, if the Attribute type is Integer, NNMi does a numeric comparison. If the Attribute type is textual, NNMi does a lexographical string comparison. In all cases, when you use the **like** or **not like** operator, NNMi uses a lexographical string comparison.
- Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The AND and OR Boolean Operators must contain at least two expressions.
- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Additional Filters Editor. For example, you append to or replace, the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators. See "Add Boolean Operators in the Additional Filters Editor" (on page 193) for more information.

Filter Editor Buttons and Drag and Drop Feature

Button or Feature	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed Left or Right Expression.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression.

Button or Feature	Description
	Note: If the Boolean Operator is selected, the Filter Editor deletes all expressions associated with the Boolean Operator.
Drag and Drop	You can drag any of the following items to a new location in the Filter String: ■ Filter Editor Options: AND, OR, NOT, EXISTS, NOT EXISTS ■ Filter Expression (Attribute, Operator and Value) When moving items in the Filter String, note the following: ■ Click the item you want to move before dragging it to a new location. ■ As you drag a selected item, an underline indicates the target location. ■ If you are moving the selection up, NNMi places the item above the target location. ■ If you are moving the selection down, NNMi places the item below the target location. ■ If you attempt to move the selection to an invalid target location, NNMi displays an error message.

To configure a Source Node Filter for a Causal Rule:

1. Navigate to the **Custom Correlation Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Select **Custom Correlation Configuration**.
2. Navigate to the **Causal Rules** tab.
3. From the **Causal Rules** table toolbar, do one of the following:
   - To create a Causal Rule, click the New icon, and continue.
   - To edit a Causal Rule, click the Open icon in the row representing the Causal Rule you want to edit, and continue.
   - To delete a Causal Rule, click the Delete icon.
4. Create your Causal Rule. (See "Configure a Causal Rule" (on page 385).)
5. Create your Child Incident Configuration . (See "Configure a Child Incident for a Causal Rule" (on page 387).)
6. *Optional*. Configure a Child Incident Filter. (See "Configure a Child Incident Filter for a Causal Rule" (on page 389).)
7. *Optional*. Configure a Source Object Filter. (See "Configure a Source Object Filter for a Causal Rule" (on page 397).)
8. *Optional*. Configure a Source Node Filter. (See the tables that follow, starting with Filter Editor

Components.)

Filter Editor Components

Component	Description
Attribute	The Attribute on which NNMi searches. See Valid Attributes below for a description of valid Attributes.
Operator	Use this Operator to establish the relationship between the Attribute and Expression. See Valid Operators in the table below for the description of each valid Operator.
Expression	Use the Expression to complete the criteria for the Parent Incident configuration. See Valid Expressions below for a description of the components that you might include in your Expression.

9. Click **Save and Close** to save your changes and return to the previous form.

Valid Attributes

Attribute	Description

Attribute	The Attribute on which NNMi searches.
	Note the following when specifying Attributes:
	• Boolean Attributes begin with "is" and must contain the value `true` or `false`.
	• Use the following syntax to specify a Custom Incident Attribute (CIA):
	`valueOfCia(<CIA_Name>)`
	Note: Check the appropriate Incident form for any valid CIA Names provided by NNMi.
	For example: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} = 5`
	• When specifying *CIA_Name* or *CA_Name* you can use the asterisk (*) as a wildcard character to specify one or more characters and a question mark (?) to specify a single character.
	• Node [click here for a list of attribute values]
	Use the following syntax to specify a Custom Attribute (CA) for a Node:
	`valueOfNodeCa(<CA_Name>)`
	For example: `${valueOfNodeCA(Location)} = USA`
	Values from the Basics Attributes on the Node Form:
	▪ hostname (Hostname, *case-sensitive*)
	▪ mgmtIPAddress (Management Address)
	▪ isSnmpNode (Agent Enabled)
	▪ isNnmSystemLocal (NNMi Management Server)
	Values from the Node Form: General Tab:
	▪ sysName (System Name)
	▪ sysContact (System Contact)

Attribute	Description
	■ sysLocation (System Location) ■ sysOidNode (System Object ID) **Addresses from the Node Form: IP Addresses Tab**: ■ hostedIPAddress (Address) Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<,** or **<=**. **Unique Keys from the Node Form: Capabilities Tab**: ■ capability (Unique Key of the Capability) **Values from the Basics Attributes on the associated Device Profile Form**: ■ devVendorNode (Device Vendor) ■ devFamillyNode (Device Family) **Values from the associated entry on the Regional Manager Form: Connection Tab**: ■ nnmSystemName (Hostname, *case-sensitive*) (*NNMi Advanced*) If the Global Network Management feature is enabled, this attribute value identifies a Regional Manager (NNMi management server).

Attribute	Description

Valid Operator Values

Operator	Description
=	Finds all values equal to the value specified. Click here for examples. Match any incident with a CIA value of 5 and Object Identifier (OID) value of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} = 5`
!=	Finds all values not equal to the value specified. Click here for an example. Match any incident with a Source Node value that has a Device Vendor value not equal to `Cisco`: `${devVendorNode} != Cisco`
<	Finds all values less than the value specified. Click here for an example. Match any incident with a CIA value less than 5 and Object Identifier (OID) value of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} < 5`
<=	Finds all values less than or equal to the value specified. Click here for examples. Match any incident with a CIA value less than or equal to 5 and Object Identifier (OID) value of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} <= 5`
>	Finds all values greater than the value specified. Click here for an example. Match any incident with a CIA value greater than 5 and Object Identifier (OID) value of .1.3.6.1.4.1.9.9.106.2.0.1: ${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} > 5
>=	Finds all values greater than or equal to the value specified.
is not null	Finds all non-blank values. Click here for an example. Match any incident with a Source Node that has a (system contact name) sysContact value: `${sysContact} is not null`
is null	Finds all blank values. Click here for an example. Match any incident with a Source Node that does not have a (system contact name) sysContact value:

Operator	Description
	`${sysContact} is null`
like	Finds matches using wildcard characters and the question mark. The asterisk (*) character means *any number of characters of any type at this location*. The question mark (?) character means *any single character of any type at this location*. Click here for an example. Match any incident with a Source Node that has a (system location) sysLocation value that begins with Bldg5: `${syslocation} like Bldg5*`
not like	Finds all matches that do not have the values specified. The asterisk (*) characters means *any number of characters of any type at this location*. The question mark (?) character means *any single character of any type at this location*. Click here for an example. Match any incident with a Source Node that has a (system location) sysLocation value that does not begin with `Bldg5`: `${sysLocation} not like Bldg5*`

Valid Expressions

Attribute	Description
Expression	The value or pattern for which you want NNMi to search. Note the following: • The expression can include a valid Attribute. • The value or pattern you want to match is case sensitive.

Causal Rule Example

Tip: Use these steps as a guideline for creating your own Causal Rules.

This example creates a Causal Rule that generates a new CardHealthProblem Parent Incident. It uses the traps described in the following table to determine the following:

• Whether there is a temperature problem or diagnostic failure for a Field Replaceable Unit (FRU) Card module

• Whether the source of the problem is a fan, a power supply, or both.

Trap Descriptions

Trap	Description
FruModuleStatusChange	Indicates a temperature problem (14) or diagnostic failure (11) for the Field Replaceable Unit (FRU) card module
CiscoEnvMonFanNotification	Indicates the problem is related to a fan. The example Causal Rule uses this trap to obtain the name of the fan.

Trap	Description
CiscoEnvMonSuppStatusChangeNotif	Indicates the problem is related to the Power Supply.

Using the Causal Rule described in this example, NNMi generates a new CardHealthProblem Parent Incident when NNMi determines the following:

- The Source Object for the Child Incident is a Field Replaceable Unit (FRU) card.

  Note: NNMi checks for the **com.hp.nnm.capability.card.fru** capability to determine whether the Source Object is an FRU card.

- The FruModuleStatusChange trap returns a value of either 14 (temperature problem) or 11 (diagnostic failure).

To configure the CardHealth Causal Rule Basics information:

1. Navigate to the **Causal Rule** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Select **Custom Correlation Configuration**.
2. Navigate to the **Causal Rules** tab.
3. From the **Causal Rules** table toolbar, click the New icon.
4. In the **Name** attribute, enter a unique name that will help you to identify the Causal Rule. In this example, the Causal Rule Name is **Card Health**.
5. In the **Author** attribute, either enter a name that identifies the person who is creating the Causal Rule or keep the default value **Customer**.
6. Make sure **Enabled** is checked to indicate the NNMi Causal Engine should use this Causal Rule when evaluating incidents.
7. To create a new Incident Configuration for the Parent Incident, in the **Parent Incident** Lookup Field, select New.
8. In the Management Event Configuration form, enter the **Basics** information as follows:
   a. In the **Name** attribute, enter **CardHealthProblem** for the Name value.
   b. Make sure **Enabled** is checked to indicate the NNMi Causal Engine should use this Causal Rule when evaluating incidents.
   c. In the **Categories** Lookup Field, select Quick Find and select **Fault** from the list of incident Categories.
   d. In the **Family** Lookup Field, select Quick Find and then **Card** from the list of incident Families.
   e. In the **Severity** Lookup Field, select Quick Find and then **Critical** from the list of incident Severities.
   f. In the Message Format attribute, enter the following: `Card $.1.3.6.1.2.1.47.1.1.1.1.7.5000 with $.1.3.6.1.4.1.9.9.13.1.4.1.2 and Power Supply not functioning`

NNMi displays the name of the Card using the Object Identifier (OID) value of $.1.3.6.1.2.1.47.1.1.1.1.7.5000. NNMi displays the name of the Fan using the OID value of $.1.3.6.1.4.1.9.9.13.1.4.1.2.

g. Click **Save and Close** to save your changes and return to the **Causal Rule** form.

9. In the **Correlation Nature** select **Root Cause** from the drop-down list.
10. In **Common Child Incident Attribute**, enter **${hostname}**.
11. In the **Correlation Window Duration** attribute, keep the default value of **5** minutes.
12. Use the **Description** attribute to provide additional information that you would like to store about the current incident configuration. This description applies only to the configuration entry.

To configure the first Child Incident (CiscoModuleStatusChange):

1. In the **Causal Rule** form, navigate to the **Child Incidents** tab.
2. Click the New icon to configure the first Child Incident.
3. In the **Name** attribute of the Child Incident Configuration form, enter **FRU Card**.
4. In the **Child Incident** Lookup Field, select Quick Find and then **CiscoModuleStatusChange** from the list of incident configurations.
5. To forward the Card name to the new Parent Incident, in **Forward Child Custom Incident Attributes**, enter **.1.3.6.1.2.1.47.1.1.1.1.7.5000**.
6. Check to enable **Use Child Incident's Source Object for Parent** ☑.
7. Check to enable **Use Child Incident's Source Node for Parent** ☑.

To configure the first Child Incident Filter:

1. In the **Child Incident Configuration** form, navigate to the **Child Incident Filter** tab.

   Next, create the following filter: `(capability = com.hp.nnm.capability.card.fru AND ${valueOfCia(\Q.1.3.6.1.4.1.9.9.117.1.2.1.1.2.5000\E)} = 11) OR ${valueOfCia(\Q.1.3.6.1.4.1.9.9.117.1.2.1.1.2.5000\E)} = 14)`
2. In the **Attribute** field, enter **capability**.
3. In the **Operator** field, select **=** from the drop-down menu.
4. In the **Expression** field, enter **com.hp.nnm.capability.card.fru**.
5. Click **Append**.
6. Select **Insert** from the drop-down menu.
7. Click **AND**.
8. Click to highlight **AND** in the Child Incident Filter Expression.
9. Select **Append** from the drop-down menu.
10. Click **OR**.
11. Click to highlight **OR** in the Child Incident Filter Expression.
12. In the **Attribute** field, enter **${valueOfCia(\Q.1.3.6.1.4.1.9.9.117.1.2.1.1.2.5000\E)}**.
13. In the **Operator** field, select **=** from the drop-down menu.
14. In the **Expression** field, enter **11**.

15. Click **Append**.
16. In the **Attribute** field, enter **${valueOfCia(\Q.1.3.6.1.4.1.9.9.117.1.2.1.1.2.5000\E)}**.
17. In the **Operator** field, select **=** from the drop-down menu.
18. In the **Expression** field, enter **14**.
19. Click to highlight **OR** in the Child Incident Filter Expression.
20. Click **Append**.
21. Click **Save and Close** to return to the **Causal Rule** form.

To configure the second Child Incident (CiscoEnvMonFanNotification):

1. In the **Causal Rule** form, navigate to the **Child Incidents** tab.
2. Click the New icon to configure the second Child Incident.
3. In the **Name** attribute of the **Child Incident Configuration** form, enter **Chassis Fan**.
4. In the **Child Incident** Lookup Field, select Quick Find and then **CiscoEnvMonFanNotification** from the list of incident configurations.
5. To forward the Fan name to the new Parent Incident, in **Forward Child Custom Incident Attributes**, enter **.1.3.6.1.2.1.47.1.1.1.1.7.5000**.

To configure the second Child Incident Filter:

1. In the **Child Incident Configuration** form, navigate to the **Child Incident Filter** tab.

   Next, create the following filter: `(${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.2\E)} = Chassis Fan Tray 1 AND ${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.3\E)} = 3)`
2. In the **Attribute** field, enter **${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.2\E)}**.
3. In the **Operator** field, select **=** from the drop-down menu.
4. In the **Expression** field, enter **Chassis Fan Tray 1**.
5. Click **Append**.
6. Select **Insert** from the drop-down menu.
7. Click **AND**.
8. In the **Attribute** field, enter **${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.3\E)}**.
9. In the **Operator** field, select **=** from the drop-down menu.
10. In the **Expression** field, enter **3**.
11. Click **Append**.
12. Click **Save and Close** to return to the **Causal Rule** form.

To configure the third Child Incident (CiscoEnvMonSuppStatusChangeNotif):

1. In the **Causal Rule** form, navigate to the **Child Incidents** tab.
2. Click the New icon to configure the third Child Incident.
3. In the **Name** attribute of the **Child Incident Configuration** form, enter **Chassis Power**.
4. In the **Child Incident** Lookup Field, select Quick Find and then

CiscoEnvMonSuppStatusChangeNotif from the list of incident configurations.

5. To forward the Fan name to the new Parent Incident, in **Forward Child Custom Incident Attributes**, **enter ${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.2\E)}**.

To configure the third Child Incident Filter:

1. In the **Child Incident Configuration** form, navigate to the **Child Incident Filter** tab.

   Next, create the following filter: `(${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.3\E} = 3} AND ${valueOfCia(\Q.1.3.6.1.4.1.9.9.117.1.2.1.1.2.5000\E} = Power Supply 1, WS-CAC-1300W)`

2. In the **Attribute** field, enter **${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.3\E)}**.
3. In the **Operator** field, select **=** from the drop-down menu.
4. In the **Expression** field, enter **3**.
5. Click **Append**.
6. Select **Insert** from the drop-down menu.
7. Click **AND**.
8. In the **Attribute** field, enter **${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.2\E)}**.
9. In the **Operator** field, select **=** from the drop-down menu.
10. In the **Expression** field, enter **Power Supply 1, WS-CAC-1300W**.
11. Click **Append**.
12. Click **Save and Close** to save your changes and return to the **Causal Rule** form.
13. Click **Save and Close** to save your changes and return to the **Custom Correlation Configuration** form.
14. Click **Save and Close** to save the Custom Correlation Configuration.

See "Correlation Rule Example" (on page 382) for an example of creating a Correlation Rule.

Configure an Action for an Incident

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 (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

You can provide the required information within the following contexts:

"Configure Actions for an SNMP Trap Incident" (on page 571)

"Configure Actions for a Remote NNM 6.x/7.x Event Incident" (on page 716)

"Configure Actions for a Management Event Incident" (on page 850)

Lifecycle Transition Action Form

Use this form to enter the command you want to run when an incident of the type you are configuring is at a particular Lifecycle State. For example, when an incident is generated (**Registered**), you might want to automatically open a trouble ticket or email or page your network operator.

You can provide the required information within the following contexts:

"Lifecycle Transition Action Form (SNMP Trap Incidents)" (on page 572)

"Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" (on page 717)

"Lifecycle Transition Action Form (Management Events)" (on page 851)

Valid Parameters for Configuring Incident Actions (Management Events)

When configuring incident actions, you may want to use incident information as part of the action. NNMi provides the following parameter values. Use these parameters as variables in your Jython or executable files.

Tip: See the Using the Incident Form for more information about the parameter values.

Note: NNMi stores varbind values as custom incident attributes (CIAs).

See "Lifecycle Transition Action Form" (on page 415) for more information about configuring incident actions.

Valid Parameters Visible From an Incident's Form

Parameter Value	Description
$category, $cat	Value of the Category attribute in the Incident form.
$count, $cnt	Value representing the number of Custom Incident Attributes that appear in the Incident form.
$family, $fam	Value from the Family attribute in the Incident form.
$firstOccurrenceTime, $fot	Value from the First Occurrence Time attribute in the incident form.
$lastOccurrenceTime, $lot	Value from the Last Occurrence Time attribute in the incident form.
$lifecycleState, $lcs	Value from the Lifecycle State attribute in the Incident form.
$name	Value of the Name attribute from the incident configuration.
$nature, $nat	Value from the Nature attribute in the Incident form.
$origin, $ori	Value from the Origin attribute in the Incident form.
$originOccurrenceTime, $oot	Value from the Origin Occurrence Time attribute in the incident form.
$priority, $pri	Value from the Priority attribute in the Incident form.
$severity, $sev	Value of the Severity attribute of the Incident form.

Valid Parameters Visible from a Node Form

Parameter Value	Description
$managementAddress, $mga	Value from the Management Address attribute of the incident's source Node's form or SNMP Agent form.
$otherSideOfConnectionManagementAddress, $oma	If the incident's Source Node is part of a Layer 2 connection, this attribute is the value of the Management Address of a node on the other side of the Layer 2 connection.
$sourceNodeLongName, $sln	The fully-qualified DNS name as displayed in the Hostname attribute of the incident's source Node's form.
$sourceNodeName, $snn	Value from the Name attribute of the incident's source Node's form.
$sysContact, $sct	Value from the System Contact attribute of the incident's source Node form: General tab.
$sysLocation, $slc	Value from the System Location attribute of the incident's source Node form: General tab.

Valid Parameters Visible from an Interface Form

Parameter Value	Description
$ifAlias, $ifa	Value from the IfAlias attribute for the interface that is the incident's source object.
$ifConfigDupSetting, $icd	Configured Duplex Setting on the port associated with the interface that is the incident's source object.
$ifDesc, $idc	Value from the ifDesc attribute for the interface that is the incident's source object.
$ifIndex, $idx	Value from the ifIndex attribute for the interface that is the incident's source object.
$ifIpAddr, $iia	IP Address values associated with the interface that is the incident's source object. If multiple IPaddresses are associated with the interface, this parameter returns a comma-separated list.
$ifName, $ifn	Value from the ifName attribute for the interface that is the incident's source object.
$ifPhysAddr, $ipa	Value from the Physical Address attribute for the interface that is the incident's source object.
$ifSpeed, $isp	Value from the ifSpeed attribute for the interface that is the incident's souce object.
$ifType, $itp	Value from the ifType attribute for the interface that is the incident's souce object.

Valid Parameters Visible from a Layer 2 Connection Form

Parameter Value	Description
$otherSideOfConnectionConfigDupSetting, $ocd	If the incident's source Node is part of a Layer 2 connection,

Parameter Value	Description
	this parameter contains the Configured Duplex Setting on the port associated with the interface on the other side of the connection.
$otherSideOfConnectionIfAlias, $oia	If the incident's Source Node is part of a Layer 2 connection, this parameter is the value of the ifAlias of one of the interfaces on the other side of the Layer 2 connection.
$otherSideOfConnectionIfDesc, $odc	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifDescr attribute value for the interface on the other side of the Layer 2 connection.
$otherSideOfConnectionIfIndex, $odx	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifIndex attribute value for the interface on the other side of the connection.
$otherSideOfConnectionIfName, $ofn	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifName attribute value for the interface on the other side of the connection.

Valid Parameters Visible from a VLAN Form

Parameter Value	Description
$impVlanIds, $ivi	Value from the VLAN Id attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Port tab and open the form for the VLAN of interest. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.
$impVlanNames, $ivn	Value from the VLAN Name attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Ports tab of the Interface form. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.

Valid Parameters Not Visible From a Form

Parameter Value	Description
$id	Unique Object Identifier attribute value for the incident (unique across the entire NNMi Database).
$firstOccurrenceTimeMS, $fms	Value from the First Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$lastOccurrenceTimeMS, $lms	Value from the Last Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$oid	Value of the unique object identifier (oid) for the incident configuration that originated from either an SNMP trap, Management Event, or Remote NNM 6.x or 7.x event.

Parameter Value	Description
$otherSideOfConnection, $osc	If the incident's Source Node is part of a Layer 2 connection, this attribute is the following combination of values for the node and one of its interfaces on the other side of the Layer 2 connection: The fully-qualified DNS name of the node appended with the interface Name in the following format: <*fully-qualified DNS name*>[*interface_name*]
$originOccurrenceTimeMS, $oms	Value from the Origin Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$sourceNodeUuid, $snu	Universally Unique Object Identifier attribute value of the source node object for the incident (unique across all databases). This identifier distinguishes the source node object instance from all other node objects.
$sourceObjectClass, $soc	Value of the object class for the object you want to include. Use this parameter when you need to request more details of a class of objects through a web service. Examples of object classes include: `com.hp.ov.nms.model.core.Interface` and `com.hp.ov.nms.model.snmp.SnmpAgent`.
$sourceObjectName, $son	Value from the Name attribute of the source object. For example, an interface object is named according to the MIB ifName. Each ifName varies according to the vendor's conventions. Using the name `4/1` as an example, `4` represents `the board number and 1` represents the port number.
$sourceObjectUuid, $sou	Universally Unique Object Identifier attribute value of the source object for the incident (unique across all databases). This identifier distinguishes the source object instance from all other similar object instances..
$uuid	Universally Unique Object Identifier attribute value of the incident (unique across all databases). This identifier distinguishes the incident object instance from all other incident objects.

Valid Parameters Established in Custom Incident Attributes

Parameter Value	Description
$<position_number>	Value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1` NNMi stores varbind values as Custom Incident Attributes. If you know the varbind position number, use this parameter.
$<CIA_name>	Value of the name that is used for the custom incident attribute. For example, `$mycompany.mycia`. NNMi provides CIA values for configuring Management Events. See Custom Incident Attributes Provided by NNMi for more information about custom incident attributes.
$<CIA_oid>	Value of the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this parameter when you are not certain of a custom incident attribute (varbind) position number.

Parameter Value	Description
$*	Used to indicate you want all of the custom incident attribute values originating as varbinds, to be passed to the action configuration. Each varbind is returned in the following format: `$<CIA_name>:<CIA_value>` in which the custom incident attribute name appears followed by the custom incident attribute value.

The function described in the following table replaces the specified numeric value with the associated text value stored in the CIA.

Note: The associated MIB must have been loaded using the nnmloadmib.ovpl command.

Functions to Generate Values Within Incident Messages

Function	Description
$text($<position_number>)	The <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1`. After the function runs, NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.
$text($<CIA_oid>)	The <CIA_oid> argument specifies the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this argument to the $text function when you are not certain of a custom incident attribute (varbind) position number. After the function runs, NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.

Handling Special Characters in Action Arguments

In some cases, NNMi requires or inserts double quotes or escape characters in arguments that are passed to the Jython file, executable, or shell script using the **Command** attribute.

Note: Shell commands are not allowed in the **Command** attribute. If you need to use shell commands, place them in a shell script file and reference that file from the **Command** attribute.

The following table describes how to handle special characters included as arguments to your Jython files, executables, or shell scripts.

Handling Special Characters in Arguments

Circumstance	Result
If the following special characters are requested as a single argument to a Jython, executable, shell script, or shell command: , ; & > < (space) | =	The argument (containing the special character) must be wrapped in double quotes. For example, **"Hello;World"**.

Circumstance	Result

Request all available CIA name/value pairs for a particular incident
 	
$*
	The $* argument returns a parsed string. For this example, the available CIA name/value pairs are:
	• **$1** = 123
	• **$com.mycompany.mycia** = 012345
	• **$.1.3.6.1.2.1.2.2.1.1** = 1007
	Example Command
	`echoScript.bat $*`
	NNMi returns the following string in response to the command:
	• **Windows:**
	`"1:123,com.mycompany.mycia:012345,.1.3.6.1.2.1.2.2.1.1:1007"`
	• **UNIX:**
	`1:123,com.mycompany.mycia:012345,.1.3.6.1.2.1.2.2.1.1:1007`

Request specific CIA values as an argument to an action command
 	
$<*CIA name, position, or OID*>
	To request specific CIA values, use the $ followed by the CIA name
	Example Command
	`echoScript.bat $1 $com.mycompany.mycia $.1.3.6.1.2.1.2.2.1.1`
	For this example, the CIA name/value pairs are:
	• **$1** = 123
	• **$com.mycompany.mycia** = 012345
	• **$.1.3.6.1.2.1.2.2.1.1** = 1007
	NNMi returns the following string in response to the command:
	• **Windows:**
	`123 012345 1007`
	• **UNIX:**
	`123 012345 1007`

Circumstance	Result

If an invalid CIA name, position, or OID is requested as an argument to an action command	If the trap or event does not contain one or more of the requested CIAs, NNMi passes error messages as arguments.
	UNIX:
	`Invalid or unknown cia position 1`
	`Invalid or unknown cia com.mycompany.mycia`
	`Invalid or unknown cia .1.3.6.1.2.1.2.2.1.1`
	Windows: NNMi encloses each CIA value in double quotes.
	`Invalid or unknown cia "position 1"`
	`Invalid or unknown cia "com.mycompany.mycia"`
	`Invalid or unknown cia ".1.3.6.1.2.1.2.2.1.1"`

Use $* in your incident action scripts	**UNIX:** It is recommended that you do not use $* (shell variable substitution) in your incident action scripts. If you do use $* within the shell script, specifying $* expands into the arguments and are rescanned. This means that blanks in arguments will result in multiple arguments. If you want to use shell variable substitution, use the `"$@"` instead so that blanks in arguments are ignored.
Use arguments to Jython methods	Enclose any argument that is not preceded with a "$" (dollar sign) in double quotes. For example, jythonMethod($Severity, "Hello; World").

Configure Diagnostics for an Incident (*NNM iSPI NET*)

HP Network Node Manager iSPI Network Engineering Toolset Software provides a set of Diagnostics (Flow Definitions) that can be run on the Source Node each time an incident reaches a specified Lifecycle State (for example, as soon as an incident becomes Registered).

These Diagnostics are sets of automated commands specific to one or more device types, including Cisco routers and switches, Cisco switch/routers, and Nortel switches.

See "Configure Device Profiles" (on page 135) for more information about device types . See "Diagnostics (Flows) Provided by NNM iSPI NET" (on page 422) for more information about the Diagnostics provided by NNMi.

Configuring NNMi to automatically gather diagnostic information about the Source Node whenever a specified incident reaches a selected Lifecycle State is a two-step process:

1. Specify the Node Group providing the required information within one of the following contexts:
   - "Configure Node Settings for an SNMP Trap Incident" (on page 488)
   - "Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" (on page 634)
   - "Configure Node Settings for a Management Event Incident" (on page 778)
2. Specify the Diagnostics (Flow Definitions) providing the required information within one of the following contexts:

- "Configure Diagnostics Selections for a Node Group (SNMP Trap Incident) (NNM iSPI NET)" (on page 527)
- "Configure Diagnostics Selections for a Node Group (Remote NNM 6.x/7.x Events)" (on page 673)
- "Configure Diagnostics Selections for a Node Group (Management Events)" (on page 813)

Diagnostic Selections Form (*NNM iSPI NET*)

With HP Network Node Manager iSPI Network Engineering Toolset Software, the Diagnostic Selections form enables you to configure NNMi to automatically gather diagnostic information for the Incident you are configuring. When using this form, you specify the diagnostics you want to run on each applicable node in the specified Node Group.

You can provide the required information within the following contexts:

"Configure Diagnostics Selections for a Node Group (SNMP Trap Incident) (NNM iSPI NET)" (on page 527)

"Configure Diagnostics Selections for a Node Group (Remote NNM 6.x/7.x Events)" (on page 673)

"Configure Diagnostics Selections for a Node Group (Management Events)" (on page 813)

Diagnostics (Flows) Provided by NNM iSPI NET

The HP Network Node Manager iSPI Network Engineering Toolset Software Diagnostics (Flows) are sets of automated commands specific to one or more device types. You can associate these Diagnostics with specific incident configurations. After you associate a Diagnostic with an incident configuration and specify the Lifecycle State for which the Diagnostic should run, the Diagnostic automatically runs on the Source Node for the incident whenever the specified Lifecycle State is reached. See "Configure Diagnostics for an Incident (NNM iSPI NET)" (on page 421) for more information.

NNMi also associates these Diagnostics with each node to which the Diagnostics apply. To view the Diagnostics invoked for each node, open the Node form for any node of interest. See Node Form: Diagnostics Tab for more information.

NNMi provides Diagnostics (Flows) for the following device types:

- Cisco router
- Cisco switch
- Cisco switch/router (see Cisco router and Cisco switch)
- Nortel switch

Cisco Router Diagnostics (Flow Definitions) Provided by NNMi

Name	Description

Cisco Router Baseline Information	Uses a series of show commands to determine the current configuration of a Cisco router. It first displays the router's and NNMi management server's current times. Next, it invokes a series of commands on the router and formats these results on the summary page. Click here for a list of the commands included in this Diagnostic.
	`show version`
	`show protocol`

Name	Description

	`show interface summary`
	`show ip route`
	`show ip protocol`
	`show ip traffic`
	`show vlans`
	`show cdp`
	`show cdp entry`
	`show cdp neighbors`
	`show log`
	`show stacks`

Cisco Show IP Route	Obtains routing information using the `show ip route` command.

Cisco Route To Node Diagnostic	**Note**: This Diagnostic Flow is not associated with an NNMi incident or node object and can only be run from Operations Orchestration Central's Flow Library. Before the Diagnostic runs, you are prompted for access information for the source router and target device node.
	Determines failures of either ping or traceroute to a target node. Uses the router to perform a ping and a traceroute to a target node.
	Click here for a list of commands included in this Diagnostic
	`ping target`
	`traceroute target`

Cisco Interface Diagnostic	Performs a number of diagnostic checks on a specified interface on the Cisco router. Diagnostics performed include whether the link is Down while the interface is Up. The following error counts are checked: • Input errors • CRC errors • Frame errors • Overrun errors • Ignored errors

Cisco Switch Diagnostics (Flow Definitions) Provided by NNMi

Name	Description
Cisco Switch Baseline Information	Uses a series of show commands to determine the current configuration of a Cisco switch. It first displays the switch's and NNMi management server's current times. Next, it invokes a series of commands on the switch and formats these results on the summary page.Click here for a list of the commands included in this Diagnostic.

Name	Description

	`show version`
	`show protocol`
	`show interface summary`
	`show vlans`
	`show cdp`
	`show cdp entry`
	`show cdp neighbors`
	`show log`
	`show stacks`

Cisco Switch Spanning Tree Baseline	Gathers spanning tree protocol and port information from the Cisco switch. The commands run depend on the device's operating system: IOS: show spanning-tree brief CATOS; show spantree

Nortel Switch Diagnostics (Flow Definitions) Provided by NNMi

Name	Description
Nortel Port Diagnostic	Determines statistics, including rate-limit and usage for a specified port on a Nortel switch. This Diagnostic detects rate limit, reception and transmission errors. Similar to Cisco Interface Diagnostic, this flow identifies the following types of errors on the identified port: • FCS errors • Undersized packets • Oversized packets • Collisions • Single collisions • Multiple collisions • Excessive collisions • Deferred packets • Late collisions

Nortel Route to Node Diagnostic	**Note**: This Diagnostic Flow is not associated with an NNMi incident or node object and can only be run from Operations Orchestration Central's Flow Library. Before the Diagnostic runs, you are prompted for access information for the source router and target device node.
	Determines failures of either ping or traceroute to a target node.
	Click here for a list of commands included in this Diagnostic
	`ping target`
	`traceroute target`

Name	Description

Nortel Switch Baseline	Determines the configuration of a Nortel switch. It first displays the switch's and NNMi management server's current times. Next, it invokes a series of commands on the switch and formats the results on the summary page. Click here for a list of commands included in this Diagnostic
	`show sys-info`
	`show interface`
	`show logging config`
	`show ssh global`
	`show stack-info`
	`send show rate-limit`
	`send show vlan`

Nortel Switch Spanning Tree Baseline	Gathers spanning tree protocol and port information from the Nortel switch. Click here for a list of commands included in this Diagnostic
	`show spanning-tree config`
	`show spanning-tree port`
	`show spanning-tree vlans`

Incident Configurations You Might Want to Enable

NNMi enables you to choose whether you want to generate an Incident for any Incident Configuration that is stored in the NNMi database. To do so you use the **Enable** attribute for each Incident Configuration.

Note: You can use the Actions menu from the NNMi console to Enable or Disable one or more Incident Configurations. See "Enable or Disable Configurations" (on page 26) for more information.

By default, not all of the Incident Configurations NNMi provides are enabled.

To determine which Incident Configurations are enabled:

1. Navigate to the **Incident Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
2. Navigate to the Incident Configuration tab of interest (**SNMP Traps**, **Remote NNM 6.x/7.x Events**, or **Management Events**).
3. Click the **Enable** column heading to sort the Incident Configurations according to the **Enable** configuration setting.

   NNMi displays a ✔check in the Enabled column for each Incident Configuration that is enabled.

You might want to enable the following Incident Configurations:

"Generate Interface Disabled Incidents" (on page 426)

"Generate Card Disabled Incidents" (on page 426)

"Generate Card Undetermined State Incidents"

"Generate Performance Threshold Incidents (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 426)

Generate Interface Disabled Incidents

By default, NNMi *does not generate* an incident for interfaces with **Administrative Status** set to **Down**. If you want NNMi to generate incidents for these disabled interfaces, use the following procedures.

To enable the Interface Disabled Management Event incident configuration:

1. Navigate to the Incident Configuration view.
    a. In the Workspace navigation panel, select the **Configuration** workspace.
    b. Select the **Incident Configuration** view.
2. Select the **Management Events** tab.
3. Click the Open icon in the row that represents the Interface Disabled configuration.
4. Click Enable.

Generate Card Disabled Incidents

By default, NNMi *does not generate* an incident for cards with**Administrative Status** set to **Down**. If you want NNMi to generate incidents for these disabled cards, use the following procedures.

To enable the Card Disabled Management Event incident configuration:

1. Navigate to the Incident Configuration view.
    a. In the Workspace navigation panel, select the **Configuration** workspace.
    b. Select the **Incident Configuration** view.
2. Select the **Management Events** tab.
3. Click the Open icon in the row that represents the Card Disabled configuration.
4. Click Enable.

Generate Performance Threshold Incidents (*HP Network Node Manager iSPI Performance for Metrics Software*)

NNMi can generate incidents related to performance thresholds. NNMi does not generate threshold incidents until the NNMi administrator configures the performance thresholds and enables the performance incidents.

To configure NNMi to generate performance threshold incidents:

Prerequisite: Enable performance polling and configure the performance thresholds. See "Configure Threshold Monitoring for Interfaces (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 228) for more information.

1. Navigate to the **Incident Configuration** form:
    a. From the workspace navigation pane, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
2. Select the **Management Events** tab.

3. Click the Open icon in the row representing the configuration you want to enable.

   Select from the following threshold incident configurations:

   InterfaceInputErrorRateHigh

   IntefaceInputDiscardRateHigh

   InterfaceInputUtilizationHigh

   InterfaceInputUtilizationLow

   InterfaceInputUtilizationNone

   InterfaceOutputDiscardRateHigh

   InterfaceOutputErrorRateHigh

   InterfaceOutputUtilizationHigh

   InterfaceOutputUtilizationLow

   InterfaceOutputUtilizationNone

   InterfacePerformanceCritical

   InterfacePerformanceWarning

4. Enable the threshold incident by checking **Enable** in the **Basics** group of the **Management EventConfiguration** form.
5. Click **Save and Close** to save your changes and return to the **Incident Configuration** form.
6. Repeat steps 3 through 6 for each configuration you want to use.

The HP Network Node Manager iSPI Performance for Metrics Software now records the number and frequency of threshold related incidents (exceptions). The HP Network Node Manager iSPI Performance for Metrics Software provides reports to help you establish the root cause of network problems. Access the HP Network Node Manager iSPI Performance for Metrics Software reports with **Actions → Reporting - Report Menu** in the incident, node, or interface views and forms. (See NNM HP Network Node Manager iSPI Performance for Metrics Software Actions.)

Manage Incoming SNMP Traps

NNMi provides several tools that enable you to manage the SNMP traps that are sent through the Event Pipeline and are configured to appear as incidents in the NNMi console. For more information about NNMI's Event Pipeline, see "About the Event Pipeline" (on page 309).

NNMi uses the following criteria to determine whether it *receives or discards incoming* traps:

- If the *incoming* trap's Source Node object or Source Object (such as card or interface) has not yet been discovered, NNMi discards the trap by default.

  Note: The NNMi administrator can change this behavior using the **Trap Handling Settings** when configuring incidents. See "Handle Unresolved Incoming Traps" (on page 432) for additional information. See also "Configure Network Devices to Send SNMP Notifications to NNMi" (on page 428).

- If the Source Node or Source Object of the *incoming* trap has been discovered by NNMi using SNMPv3, NNMi accepts *incoming* traps from SNMPv3, SNMPv2c, or SNMPv1. See "SNMPv3 Settings Form" for information about configuring SNMPv3 settings.

- If the Source Node or Source Object of the *incoming* trap has been discovered by NNMi using SNMPv2c or SNMPv1, NNMi discards *incoming* traps from SNMPv3.
- NNMi discards traps that have no Incident Configuration or with an Incident Configuration set to Disabled.
- If either the Source Node or Source Object has *Management Mode* set to **Not Managed** or **Out of Service** in the NNMi database, NNMi always discards the incoming trap. See "Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " (on page 257).

  NNMi provides the Management Mode workspace so that you can quickly view lists of all nodes, interfaces, cards, addresses, or node components that NNMi is not currently discovering or monitoring. For information about these views:

Note the following:

- If you want the NNMi management server to *forward* SNMPv3 traps to other machines in your network environment, see "Configure NNMi Security Settings for SNMPv3 Trap Forwarding" (on page 297) for additional configuration steps.
- If you want the NNMi management server to *forward* SNMPv2 traps to other machines in your network environment, see "Configuring Trap Forwarding" (on page 297) for additional configuration steps.

When managing your SNMP Traps consider performing the following tasks:

"Configure Network Devices to Send SNMP Notifications to NNMi" (on page 428)

"Load SNMP Trap Incident Configurations" (on page 429)

"Control which Incoming Traps Are Visible in Incident Views" (on page 431)

"Handle Unresolved Incoming Traps" (on page 432)

"Analyze Trap Information (NNM iSPI NET)" (on page 433)

Related Topics:

"Configuring Trap Forwarding" (on page 297)

Configure Network Devices to Send SNMP Notifications to NNMi

An SNMP notification is a message sent from an SNMP agent (SNMPv1, SNMPv2c, or SNMPv3) on a network device to notify a network management system of an event on the network device. For example, an error occurred on the network device and its SNMP agent sent a notification. The notification may either be an acknowledged inform (SNMP InformRequest) or an unacknowledged trap.

An inform is an acknowledged notification sent from one SNMP agent to another with the expectation of a reply from the recipient. If no reply is received, the inform message is resent. A trap is a notification sent from one SNMP agent to another without any expectation of a reply.

Configure SNMP agents in your network environment to send traps to the NNMi management server. Sometimes SNMP agents are configured with a recheck interval, so the trap might be sent to the NNMi management server over and over again until the problem is corrected.

The NNMi Causal Engine analyzes these traps and gathers additional information to determine the root cause. It also provides useful troubleshooting information each time an important SNMP notification is received, including the following information:

- The name or address of the node from which the notification came (Source Node)
- The notification identification (SNMP Object ID)
- Notification-specific variables (varbinds)

When configuring the SNMP agent for each network device, configure the trap-forwarding list (or trap-destination list) to include the NNMi management server's fully-qualified hostname or IP address. Refer to documentation for the SNMP agent for information about how to do this. If the NNMi management server is included on the trap-forwarding list, NNMi receives notice when something goes wrong (even if the device does not show up on your NNMi maps).

Note: For an SNMP notification to be processed by NNMi, it must be configured using the NNMi incident Configuration workspace. Many common SNMP notifications are configured in NNMi by default. See "Configure SNMP Trap Incidents" (on page 436) and "SNMP Trap Incident Configurations Provided by NNMi" (on page 315) for more information.

Load SNMP Trap Incident Configurations

NNMi enables you to automatically create or update an Incident Configuration for an SNMP trap using a MIB file. To load a trap definition using a MIB file, you can use either the command line or NNMi console:

"Load SNMP Trap Incident Configurations from the Command Line" (on page 429)

"Load SNMP Trap Incident Configurations using the Console" (on page 430)

Load SNMP Trap Incident Configurations from the Command Line

The NNMi `nnmincidentcfg.ovpl -load` script provides a way for you to automatically create or update an Incident Configuration for an SNMP trap using a MIB file. To load a MIB file, you can use the following syntax:

If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access" (on page 275) for more information.

```
nnmincidentcfg.ovpl -loadTraps <mib_file> -disableAllTraps true|false -u
<NNMiadminUsername> -p <NNMiadminPassword> | -loadMib <mib_file>
```

Note: See nnmincidentcfg.ovpl for more information, including a complete list of the valid script arguments.

nnmincidentcfg.ovpl Arguments

Argument	Description
-loadTraps <*mib_file*>	Used to load the MIB file <*mib_file*> that you want to use to create or update the incident configuration for an SNMP trap. NNMi uses information from the trap definitions (TRAP-TYPES macro) or notification (NOTIFICATION-TYPES macro) in the MIB file for the required incident configuration.
-disableAllTraps	Specifies whether all trap definitions specified using `-loadTraps` <*mib_file*> should be loaded as disabled. **Note**: The default value is `false`. This means that by default all trap definitions specified in <*mib_file*> are loaded as enabled. Set this parameter to `true` to disable the trap definitions that you are loading.
-u	The NNMi user name. This user must be assigned to at least an NNMi administrator role. **Note**: The user name might be a Principal object stored in the NNMi database or might

Argument	Description
	be from your environment's directory service database (depending on NNMi configuration). See "Configure Sign-In Access" (on page 262) for more information.
-p	The password associated with the NNMi account. If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access" (on page 275) for more information.
-loadMIB <*mib_file*>	Used to load a MIB file for the cases in which the <*mib_file*> specified with -loadTraps has dependencies.

For example, to load the MIB file CISCO-VTP_MIB, you might enter the following:

```
nnmincidentcfg.ovpl -loadTraps "C:\Cisco Mibs\CISCO-VTP-MIB.my"
```

If the incident is already configured, NNMi performs an update based on the MIB file information. If the incident is not configured, NNMi creates a new incident configuration entry for the SNMP trap. See "Configure SNMP Trap Incidents" (on page 436) for information about changing an SNMP trap configuration.

Load SNMP Trap Incident Configurations using the Console

NNMi enables you to load one or more SNMP Trap Incident Configurations from a MIB file using the NNMi console.

To load an SNMP Trap Incident Configuration from the NNMi console:

1. Do one of the following:
   a. Navigate to the MIB view or form. For example, Select **Configuration** → **Loaded MIBs**.
   b. Navigate to the MIB Variable view or form. For example, Select **Inventory** → **MIB Variables**.
2. Select **Tools** → **Load MIB...**

   NNMi displays the following information:

   - Unloaded MIBs (user provided) that are stored on the NNMi management server and that were provided by the NNMi administrator.
   - Unloaded MIBs (NNMi provided) that NNMi has stored on the NNMi management server during installation.
   - Loaded MIBs that are loaded in the NNMi database.
3. Navigate to the Unloaded MIB view of interest. For example, **Unloaded MIBs (User Provided)**.
4. In the MIB column, find the MIB that contains the trap incidents you want to load. For example, **FLOWMGREST-MIB**.

   Note: The MIB must support the TRAP-TYPE or NOTIFICATION-TYPE macro.

5. To view the MIB before loading, in the Actions column, click **Display**.

   NNMi displays the MIB file contents.

6. To load the MIB, in the Actions column, click **Load Incident Configuration**.

   NNMi displays the progress of each trap configuration that is loaded, including the following:

   - The name and location of the MIB file
   - The number of trap incident configurations
   - The name and numeric object identification (OID) of each trap configuration
   - Whether the trap incident configurations successfully loaded
   - Instructions for loading and listing MIB files.

To upload a local MIB file so that it is stored on the NNMi management server and available for loading, see "Upload MIB Files from the Console" (on page 915).

Control which Incoming Traps Are Visible in Incident Views

You can configure devices in your network environment to send traps to the NNMi management server. To configure how NNMi handles those traps, use the incident configurations provided by NNMi, create your own, or both. See "Configure SNMP Trap Incidents" (on page 436) for information about how to configure SNMP traps as incidents. See "SNMP Trap Incident Configurations Provided by NNMi" (on page 315) for information about the incident configurations provided by NNMi.

Note: To establish this communication flow, the SNMP agent (SNMPv1, SNMPv2c, or SNMPv3) must be intentionally configured by the device administrator to send SNMP traps to your NNMi management server.

After you configure an incident for each SNMP trap you want NNMi to display, NNMi stores your incident configurations for SNMP traps in the `allowedOids.conf` file. NNMi uses this file as a positive filter to identify which traps should appear as incidents.

By default, NNMi enables only the following SNMP Trap incident configurations:

- CiscoWarmStart
- CiscoColdStart
- SNMPWarmStart
- SNMPColdStart
- CiscoLinkDown
- CiscoLinkUp
- HSRPStateChange
- IetfVrrpStateChange
- RcVrrpStateChange
- SNMPLinkDown
- SNMPLinkUp
- RcnAggLinkUp
- RcAggLinkUp
- RcnAggLinkDown

- RcAggLinkDown
- RcnSmltlstLinkUp
- RcSmltlstLinkUp
- RcnSmltlstLinkDown
- RcSmltlstLinkDown

See "SNMP Trap Incident Configurations Provided by NNMi" (on page 315) for more information.

Tip: You can configure NNMi to ignore SNMP Traps for objects that are not discovered as part of the NNMi topology. See "Handle Unresolved Incoming Traps" (on page 432) for more information.

To enable or disable an SNMP trap configuration:

1. Navigate to the Incident Configuration view.
    a. In the Workspace navigation panel, select the **Configuration** workspace.
    b. Select the **Incident Configuration** view.
2. Select the **SNMP Traps** tab.
3. Click the Open icon in the row representing the configuration you want to edit.
4. To enable the incident configuration, click Enable ☑.
5. To disable the incident configuration, clear Enable ☐.

Related Topics

"Handle Unresolved Incoming Traps" (on page 432)

"Manage the Number of Incoming Incidents" (on page 341)

Handle Unresolved Incoming Traps

Your network environment might be configured to forward SNMP traps to the NNMi management server.

If the trap's source node or source object *cannot be matched with any object in the NNMi database*, that trap is considered to be *unresolved*. Follow the steps in this procedure to specify whether NNMi retains or discards these traps. For example, if you configure NNMi to discover only devices you specifically list as seeds, you can decide if you want NNMi to process or ignore traps from any other devices.

See "Manage Incoming SNMP Traps" (on page 427) for more information about the additional criteria NNMi uses to determine when to receive or discard traps.

To control how NNMi handles unresolved incoming SNMP Traps:

1. Navigate to the **Incident Configuration** form:
    a. From the workspace navigation pane, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
2. Navigate to the **NNMi Trap Handling Settings**:
    - If you want NNMi to place unresolved SNMP traps into the NNMi database, clear the **Discard Unresolved SNMP Traps** ☐ check box.

Unresolved traps then appear in incident views, but have missing information. For example, the incident might appear as follows:

- NNMi displays the Source Node as an IP address.
- NNMi displays the Source Object as **None**.

- If you want NNMi to ignore any unresolved traps, select the **Discard Unresolved SNMP Traps** ☑ check box.

Tip: To manage the number of SNMP Traps displayed as incidents, see "Control which Incoming Traps Are Visible in Incident Views" (on page 431)

Analyze Trap Information (NNM iSPI NET)

NNMi measures the rate of incoming SNMP traps regardless of Incident Configuration, including the following:

- Traps from each Node.
- Traps for each SNMP Object Identifier (OID).

NNMi monitors the incoming SNMP traffic flow to determine whether the number of traps received within a certain time period exceeds any set threshold. If a threshold is exceeded, NNMi blocks processing of additional traps until the number of traps falls below the threshold set for each time period.

Note: The NNMi administrator can configure threshold values using the nnmtrapconfig.ovpl script.

When analyzing traps, NNMi looks at both the most common traps as well as the most common Source Nodes from which the traps are received. NNMi logs this SNMP trap analytics data to the `trapanalytics.0.0.log` file.

If NNM iSPI NET is available in your network environment, you can obtain reports about incoming SNMP traps according to the criteria described in the Trap Analytics Reports table.

Note the following:

- The time interval and number of Nodes or SNMP OIDs included in the reports and Line Graphs is based on the numbers configured using the nnmtrapconfig.ovpl script. By default, NNMi uses 5 minutes as the time interval and 10 as the top number of Nodes and SNMP OIDs for which information is computed.
- NNMI identifies each trap using its SNMP Object Identifier (OID) number.
- NNMi enables you to open the following graphs, reports, and forms from a Trap Analytics report:
  - Line Graph of the trap rate for all of the Nodes or SNMP OIDs displayed in the report.
  - Line Graph of the trap rate for a selected Node or SNMP OID.
  - SNMP Trap Incident Configuration form, if any, for an SNMP OID.
  - Source IP Address and Node form for a Node.

    Note: The Source Node must be stored in the NNMi database for the links to appear.
  - MIB Variable form, if any, for the selected SNMP OID.

    Note: The MIB Variable must be stored in the NNMi database. To add a MIB Variable by loading a trap, see "Load SNMP Trap Incident Configurations" (on page 429)

- When you access a Line Graph from a report, the Line Graph displays an updated real-time data using the Nodes or SNMP OIDs included in the report. Because the trap rate is constantly changing, the data in the Line Graph will not match the historical trap numbers displayed in the report.
- If an SNMP Trap Incident Configuration exists for a trap, NNMi displays the name of the SNMP Trap Incident Configuration as well as whether the SNMP Trap Incident Configuration is disabled. This feature is useful when you want to make changes to the Incident Configuration. For example, you might want to enable or disable the Incident Configuration.
- NNMi discards traps that have no Incident Configuration or with an Incident Configuration set to Disabled. If you want to be notified whenever an SNMP Trap is recieved that does not have an associated incident configuration, you can configure NNMi to generate an **Undefined SNMP Trap** incident. For more information, see the "NNMi Incidents" chapter of the HP Network Node Manager i Software Deployment Reference, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

See the Trap Analytics Reports table for more information about the links available from each report.

To access the Trap Analytics reports:

1. Select **Tools** → **Trap Analytics (iSPI NET only)**.
2. Select the graph or report of interest. from the **Trap Analytics** submenu.

   NNMi displays the selected report (see Trap Analytics Reports).

Trap Analytics Reports

Report	Description	Links Available from the Report
Recent Top Trap Rate (by Node)	Table view of the Nodes that are most frequently generating traps during the specified time period.	Line Graph of the Nodes that are most frequently generating traps. Recent Top Rate Traps Received (by OID) report Total Traps Received (by Node) Total Traps Received (by OID) Line Graph of the trap rate for the selected Node. Source Node form, if any, for the trap.
Recent Top Trap Rate (by OID)	Table view of the traps that are most frequently generated during the specified time period.	Line Graph of the traps that are most frequently generated. Recent Top Rate

Report	Description	Links Available from the Report
		Traps Received (by Node) report Total Traps Received (by Node) Total Traps Received (by OID) Line Graph of the trap rate for the selected SNMP OID. Incident Configuration form, if any, for the selected SNMP OID. MIB variable form, if any, for the MIB variable that is associated with the SNMP OID.
Total Traps Received (by Node)	Table view of the trap totals since NNMi was last started. This report is organized by traps per Node.	Line Graph of the total number of traps received per Node since NNMi was last started. Total Traps Received (by OID) report Recent Top Trap Rate (by Node) Recent Top Trap Rate (by OID) Line Graph of a selected Node's total traps in real time. Source Node form, if any, for the selected trap.

Report	Description	Links Available from the Report
Total Traps Received (by OID)	Table view of the trap totals since NNMi was last started. This report is organized by traps per SNMP OIDs.	Line Graph of the total number of traps received since NNMi was last started. Total Traps Received (by Node) report Recent Top Trap Rate (by Node) Recent Top Trap Rate (by OID) Line Graph of the selected SNMP OID's total traps in real time. Incident Configuration form, if any, for the selected SNMP OID. MIB variable form, if any, for the MIB variable that is associated with the SNMP OID.
Trap Analysis Log	Log file listing trap information organized by the following criteria: • Trap rate in number of traps per second • The top 10 addresses that are generating traps • The top 10 traps that are being generated This information is recomputed every 5 minutes as configured in nnmtrapconfig.ovpl. Scroll to the bottom to see the latest entry. **Note**: The NNMi administrator can configure threshold values using the nnmtrapconfig.ovpl script. You can also use the `nnmtrapdump.ovpl` command to extract the data in which you are most interested from the `trapanalytics.0.0.log` file. See the nnmtrapdump.ovpl Reference Page for more information (**Help** → **Documentation Library** → **Reference Pages**, in the *User Commands* category).	

Configure SNMP Trap Incidents

Configure incidents that originate from an SNMP trap.

Note: NNMi discards traps that have no Incident Configuration or with an Incident Configuration set to Disabled. If you want to be notified whenever an SNMP Trap is recieved that does not have an associated incident configuration, you can configure NNMi to generate an **Undefined SNMP Trap** incident. For more information, see the "NNMi Incidents" chapter of the HP Network Node Manager i Software Deployment Reference, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

Tip: You can manage the number of SNMP Traps using either of the following methods: 1) "Manage the Number of Incoming Incidents" (on page 341) and 2) "Handle Unresolved Incoming Traps" (on page 432).

Note: See "Manage Incoming SNMP Traps" (on page 427) for information about the criteria NNMi uses to determine when to receive or discard traps.

To configure incidents originating from SNMP traps:

1. Navigate to the **Incident Configuration** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration.**
2. Select the **SNMP Traps** tab.
3. Do one of the following:
    - To create an SNMP trap configuration, click the New icon, and continue.
    - To edit an SNMP trap configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
    - To delete an SNMP trap configuration, select a row, click the Delete icon.
4. In the SNMP Traps form, provide the required information.
5. Click **Save and Close** to save your changes and return to the **Incident Configuration** form.

The next time that a trap of this type arrives, NNMi creates an associated incident to display in the appropriate incident views.

SNMP Trap Configuration Form

Note: See "Manage Incoming SNMP Traps" (on page 427) for information about the criteria NNMi uses to determine when to receive or discard traps.

To configure incidents originating from SNMP traps:

1. Navigate to the **Incident Configuration** form:
    a. From the workspace navigation pane, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
2. Select the **SNMP Traps** tab.
3. Do one of the following:

    Note: If you want to add or edit an SNMP trap configuration, verify that **Enabled** ☑ is selected.

- To add an SNMP trap configuration, click the New icon, and continue.
- To edit an SNMP trap configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
- To delete an SNMP trap configuration, select a row, and click the Delete icon.

4. Make your configuration choices (see table).
5. Click **Save and Close** to save your changes and return to the **Incident Configuration** form.

Tasks for SNMP Trap Configuration

Task	How
"Configure Basic Settings for an SNMP Trap Incident" (on page 438)	Use the **Basics** pane of the SNMP Trap Configuration form.
"Configure Interface Settings for an SNMP Trap Incident" (on page 453)	Use the **Interface Settings** tab of the SNMP Trap Configuration form.
"Configure Node Settings for an SNMP Trap Incident" (on page 488)	Use the **Node Settings** tab of the SNMP Trap Configuration form.
"Configure Suppress Settings for an SNMP Trap Incident"	Use the **Suppression** tab of the SNMP Trap Configuration form.
"Configure Enrichment Settings for an SNMP Trap Incident" (on page 542)	Use the **Enrichment** tab of the SNMP Trap Configuration form.
"Configure Dampening Settings for an SNMP Trap Incident" (on page 545)	Use the **Dampen** tab of the SNMP Trap Configuration form.
"Configure Deduplication for an SNMP Trap Incident" (on page 562)	Use the **Deduplication** tab of the SNMP Trap Configuration form.
"Configure Rate (Time Period and Count) for an SNMP Trap Incident" (on page 567)	Use the **Rate** tab of the SNMP Trap Configuration form.
"Configure Actions for an SNMP Trap Incident" (on page 571)	Use the **Actions** tab of the SNMP Trap Configuration form.

Configure Basic Settings for an SNMP Trap Incident

The Basics settings for an SNMP Trap Incident specifies general information for an incident configuration, including the name, severity, and message.

Note the following:

- NNMi discards traps that have no Incident Configuration or with an Incident Configuration set to Disabled. If you want to be notified whenever an SNMP Trap is recieved that does not have an

associated incident configuration, you can configure NNMi to generate an **Undefined SNMP Trap** incident. For more information, see the "NNMi Incidents" chapter of the HP Network Node Manager i Software Deployment Reference, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

- When configuring SNMP Trap incidents, if you are using SNMPv3 protocol:
  - You must also configure SNMPv3 communication using the Communication Configuration workspace. For more information,
  - If you configured SNMPv3 communication, use the **Actions** → **Communication Settings** to determine the SNMPv3 user name that NNMi will use for any node from which you want to receive SNMP Trap incidents. Make sure the node is configured with this user name when configuring SNMP trap incidents. See "Troubleshooting Communication Settings" (on page 108) for more information.
  - If you configured SNMPv1 or SNMPv2 communication, NNMi does not authenticate the community string for any node from which you want to receive SNMP Trap incidents.
- In the **Basics** group of the **SNMP Trap Configuration** form, verify that **Enable** ☑ is selected for each configuration you want to use.

For information about each SNMP Traps tab:

To configure Basic settings for an SNMP Trap incident:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **SNMP Traps** tab.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete an incident configuration, click the Delete icon.
2. Configure the required Basic settings (see table).
3. Click **Save and Close** to save your changes and return to the Incident Configuration form.

Basics Attributes for SNMP Trap Configuration

Task	How
"Specify the Incident Configuration Name (SNMP Trap Incident)" (on page 441)	Use the **Basics** pane of the **SNMP Trap Configuration** form. Specify a name that helps you to identify the configuration for subsequent use.
"Specify the SNMP Object ID" (on page 441)	Use the **Basics** pane of the **SNMP Trap Configuration** form. NNMi supports SNMPv3, SNMPv2c and SNMPv1 formats.
Specify whether you want to enable this configuration.	In the **Basics** group of the **SNMP Trap Configuration** form, verify that **Enable** ☑ is selected for each configuration you want to use.

Task	How
"Display an SNMP Trap or an NNM 6.x/7.x Event as a Root Cause Incident" (on page 443)	Use the **Basics** pane of the **SNMP Trap Configuration** form.
"Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443)	Use the **Basics** pane of the **SNMP Trap Configuration** form. You can organize your incidents using Category and Family.
"Specify the Incident Severity (SNMP Trap Incident)" (on page 446)	Use the **Basics** pane of the **SNMP Trap Configuration** form. Possible Severity values include: **Normal, Warning, Minor, Major,** and **Critical**.
"Specify Your Incident Message Format (SNMP Trap Incident)" (on page 446)	Use the **Basics** pane of the **SNMP Trap Configuration** form. The message format determines the message to be displayed for the incident.
"Specify a Description for Your Incident Configuration (SNMP Trap Incident)" (on page 453)	Use the **Basics** pane of the **SNMP Trap Configuration** form. Provide a meaningful description.
Specify an Author for Your Incident Configuration (SNMP Trap Incident)	Use the **Basics** pane of the **SNMP Trap Configuration** form to indicate who created or last modified the trap. See Author form for important information. **Caution**: If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. Click the Lookup icon and select Quick Find to access the list of existing Author values or click New to create one.

After you complete the Basic Configuration for the SNMP trap, you can also choose to configure the information described in the following table.

Additional Incident Configurations

Task	How
"Configure Interface Settings for an SNMP Trap Incident" (on page 453)	Select the **Interface Settings** tab to specify an Interface Group to which you want your incident configuration to apply.
"Configure Node Settings for an SNMP Trap Incident" (on page 488)	Select the **Node Settings** tab to specify a Node Group to which you want your incident configuration to apply.
"Configure Suppression Settings for an SNMP Trap Incident" (on page 529)	Select the **Suppression** tab to specify the criteria for discarding incidents that match the selected incident configuration.
"Configure Enrichment Settings for an SNMP Trap Incident" (on page 542)	Select the **Enrichment** tab to specify enhancements for the selected incident configuration.
"Configure Dampening Settings	Select the **Dampen** tab to specify the time interval that must be met before the incident appears in an Incident view.

Task	How
for an SNMP Trap Incident" (on page 545)	
"Correlate Duplicate Incidents (Deduplication Configuration)" (on page 346)	Select the **Deduplication** tab to specify duplicate incidents that you want to be suppressed.
"Track Incident Frequency (Rate: Time Period and Count)" (on page 347)	Select the **Rate** tab to specify a rate for duplicate incidents. After the rate limit is reached, NNMi generates an Incident to notify you of the problem.
"Configure an Action for an Incident" (on page 414)	Select the **Actions** tab to specify actions that should occur automatically when an incident changes its Lifecycle State.

Specify the Incident Configuration Name (SNMP Trap Incident)

When providing the Name for an incident configuration, use the following guidelines:

Name
The name is used to identify the incident configuration and must be unique. Use a name that will help you to remember the purpose or kind of management event, remote NNM 6.x or 7.x event, or SNMP trap for which you are configuring this incident. Name is also used to identify your Pairwise configurations.

Specify the SNMP Object ID

When configuring incidents for an SNMP trap, you are asked to provide the SNMP Object ID values that you want to use to assist you in identifying the trap.

NNMi requires that SNMPv1 trap object identifiers be converted to SNMPv2c format. The SNMP Object IDs must be entered in a format that is recognized by NNMi. Select the type of SNMP trap you want to configure from the list below to learn about the required NNMi format.

Note: In all cases, the value you enter for an SNMP Object ID must be unique.

- "SNMP Object ID Format for SNMP Traps" (on page 441)
- "SNMP Object ID Format for SNMPv1 Generic Traps" (on page 442)
- "SNMP Object ID Format for a Specific SNMPv1 Trap" (on page 442)

SNMPv1 generic traps are standard traps that are commonly used across vendors. The SNMPv1 enterprise specific traps are those traps that are generated by a particular vendor's device. The vendor is also known as the enterprise. Both include a vendor name as part of the set of information stored with each trap.

SNMP Object ID Format for SNMP Traps

NNMi stores all SNMP trap object identifier (*Object ID*) information in SNMPv2c format. (The Object IDs of SNMPv2c and SNMPv3 traps are automatically in this format. NNMi converts SNMPv1 traps to this format.)

To specify an SNMP trap object ID, open the MIB definition file for the device of interest to look up the correct ID. The MIB file includes object identifiers for all of the traps that the configured SNMP agent (SNMPv1, SNMPv2c, or SNMPv3) generates for a particular device.

In the **SNMP Object ID** attribute of the **SNMP Trap Configuration** form or **Remote NNM 6.x/7.x Event Configuration** form, enter the **SNMP Object ID** attribute value for the trap that you want to see in the console incident views.

SNMP Object ID Format for SNMPv1 Generic Traps

NNMi requires SNMPv1 trap object IDs to be converted to SNMPv2c format. The object IDs are converted according to the specifications in Request for Comments (RFC) document 2576: *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*.

When using SNMPv1 format, you can specify either generic or vendor specific traps. SNMPv1 generic traps are standard traps that are commonly used across vendors. The SNMPv1 enterprise specific traps are those traps that are generated by a particular vendor's device. The vendor is also known as the enterprise. Both include a vendor name as part of the set of information stored with each trap.

The six SNMPv1 generic traps have the following SNMP object identifiers that are recognized by SNMPv2c:

1.3.6.1.6.3.1.1.5.1 (coldStart)

1.3.6.1.6.3.1.1.5.2 (warmStart)

1.3.6.1.6.3.1.1.5.3 (linkDown)

1.3.6.1.6.3.1.1.5.4 (linkUp)

1.3.6.1.6.3.1.1.5.5 (authenticationFailure)

1.3.6.1.6.3.1.1.5.6 (egpNeighborLoss)

To configure an SNMP object identifier (SNMP OID) for a generic SNMPv1 trap, specify the SNMP object ID as described in RFC 2576. You also need to include the object identifier for the vendor name (<VendorEnterprise>) as shown below:

<*SNMPv2c generic trap OID*>.<*VendorEnterprise*

The <*vendorEnterprise*> is the object identifier for the vendor that is included with the varbind trap information.

For example, the SNMP object identifier for Cisco warmStart trap would be:

`.1.3.6.1.6.3.1.1.5.2.1.3.6.1.4.1.9`

Note: Cisco's Vendor enterprise object identifier in this example is `.1.3.6.1.4.1.9`

SNMP Object ID Format for a Specific SNMPv1 Trap

NNMi requires SNMPv1 trap object identifiers to be converted to SNMPv2c format. The object IDs are converted according to the specifications in RFC 2576: *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*.

When using SNMPv1 format, you can specify either generic or vendor specific traps. SNMPv1 generic traps are standard traps that are commonly used across vendors. The SNMPv1 enterprise specific traps are those traps that are generated by a particular vendor's device. The vendor is also known as the entrprise. Both include a vendor name as part of the set of information stored with each trap.

When specifying the SNMP object ID for an SNMPv1 specific trap, include the SNMP object ID for the vendor name and for the trap that you want to see in the console incident views.

The value you enter must be in the format:

<*VendorEnterprise*>.0.<*SpecificTrapNumber*>

The <*VendorEnterprise*> is the object identifier for the vendor that is included in the SNMPv1 trap. The <*SpecificTrapNumber*> is the SNMPv1 specific trap identification number that is provided by the vendor.

For example, for an SNMPv1 vendor object id 1.3.6.1.3.1.12.9 and specific trap number 12234, the SNMP object ID would be:

```
1.3.6.1.3.1.12.9.0.12234
```

Display an SNMP Trap or an NNM 6.x/7.x Event as a Root Cause Incident

SNMP trap and NNM 6.x/7.x events normally appear as symptoms rather than as root cause incidents. However, there might be times when you want an SNMP or NNM 6.x/7.x event to appear as a root cause incident. For example, you might want an HSRP state change (cHsrpStateChange, 1.3.6.1.4.1.9.9.106.2.0.1) trap to be listed as a root cause. This trap might occur when the hot standby has gone down indicating the system is at risk if there is a failover.

Note: To reduce "noise" associated with duplicate incidents, NNMi changes the incident Correlation Nature to **Symptom** for any user-defined Root Cause incidents that exceed the rate or deduplication threshold. Any Stream Correlation incident associated with the user-defined Root Cause incident is change to a Root Cause incident. See Stream Correlation Incidents View for more information about Stream Correlations.

To display an SNMP trap or NNM 6.x/7.x Event as a root cause incident:

Select **Root Cause** ☑ in the **SNMP Trap** or **Remote NNM 6.x/7.x Event Configuration** form.

The next time the incident occurs, the Correlation Nature attribute value changes to **Root Cause**.

To no longer display an SNMP trap or NNM 6.x/7.x Event s a root cause incident:

Clear **Root Cause** ☐ in the **SNMP Trap** or **Remote NNM 6.x/7.x Event Configuration** form.

The next time the incident occurs, the Correlation Nature attribute value changes to **Root Cause**.

Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)

When configuring incidents, NNMi provides the Category and Family attributes to help you organize your incidents.

Preconfigured Categories
The Category attribute helps you organize your incidents. Select the category that you want to be associated with this type of incident when it appears in an incident view. Each of the possible Category values is described in the following table.

Incident Categories Provided by NNMi

Category	Description
Accounting	Used to indicate problems related to usage statistics and allocation of costs associated with the billing of time and services provided by devices. This category is not used by NNMi with default configurations, but it is available for incidents you define.
Application Status	Indicates there is a problem with the health of the NNMi software. Examples of these kinds of events include license expiration (see "Extend a Licensed Capacity" (on page 1020)) or that a certain NNMi process or service lost connection to the Process Status Manager (see "Stop or Start an NNMi Process" (on page 45) and "Stop or Start NNMi Services" (on page 49)).
Configuration	Indicates there is a problem with the configuration of a managed device. For example, there is a physical address mismatch.

Category	Description
Fault	Indicates a problem with the network, for example Node Down.
Performance	Indicates a threshold has been exceeded. For example, a utility has exceeded 90 percent.
Security	Indicates there is a problem related to authentication. For example, an SNMP authentication failure.
Status	Indicates some kind of status message. Examples of these kinds of incidents include "SNMP Link Up" or an "HSRP Group status Normal" message.

Note: You can add your own Category entries to NNMi. See "Create an Incident Category (SNMP Trap Incident)" (on page 444) for more information.

Note: You can add your own Family entries to NNMi. See "Create an Incident Family (SNMP Trap Incident)" (on page 445) for more information.

Create an Incident Category (SNMP Trap Incident)

The Category attribute helps you organize your incidents. Create any Category that makes sense to you and your team. For a list of the Category codes provided by NNMi, "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443).

To create a new incident Category:

1. Navigate to the **Incident Category** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Navigate to the **SNMP Traps** tab
   d. Do one of the following:
      - To create an incident configuration, click the New icon.
      - To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit.
   e. In the configuration form, locate the **Category** attribute.
   f. Click the Lookup icon, and select New.
2. Provide the required information (see table).
3. Click **Save and Close** to save your changes and return to the previous form.

Category Code Attributes

Name	Description
Label	Incident category name. For example, Hardware Faults, or Cisco Error. Maximum size is 255 characters. Alpha-numeric, spaces, and underline characters are allowed.
Unique Key	**Caution**: After you click **Save and Close**, this value cannot be changed. Used as a unique identifier when exporting and importing configuration definitions. To ensure

Name	Description

	that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the label value as part of the unique key as shown in the following examples:
	`com.<your_company_name>.nnm.trap_conf.category.<category_label>`
	`com.<your_company_name>.nnm.event_conf.category.<category_label>`
	`com.<your_company_name>.nnm.inci_conf.category.<category_label>`
	The maximum length is 80 characters. Alpha-numeric characters and periods are allowed. Spaces are not allowed.

Create an Incident Family (SNMP Trap Incident)

The Family attribute helps you organize your incidents. Create any Family that makes sense to you and your team. For a list of the Family codes provided by NNMi, "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443).

To create a new incident Family:

1. Navigate to the **Incident Family** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **SNMP Traps** tab.
   d. Do one of the following:
      - To create an incident configuration, click the New icon.
      - To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit.
   e. In the configuration form, locate the **Family** attribute.
   f. Click the Lookup icon, and select New.
2. Provide the required information (see table).
3. Click **Save and Close** to save your changes and return to the previous form.

Family Attributes

Name	Description
Label	Family name. For example, Hardware Faults, or Cisco Error. Maximum size is 255 characters. Any character type is valid.
Unique Key	**Caution**: After you click **Save and Close**, this value cannot be changed. Used as a unique identifier when exporting and importing configuration definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the label value as part of the unique key as shown in the following examples: `com.<your_company_name>.nnm.trapConf.family.<family_label>`

Name	Description
	com.<*your_company_name*>.nnm.eventConf.family.<*family_label*> com.<*your_company_name*>.nnm.inciConf.family.<*family_label*> The maximum length is 80 alpha-numeric characters, periods allowed, no spaces allowed.

Specify the Incident Severity (SNMP Trap Incident)

The incident severity represents the seriousness calculated for the incident. Use the severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described in the following table.

Incident Severity Values

Attribute	Description
Normal	Indicates there are no known problems related to the associated object. This severity is meant to be informational. Generally, no action is needed for these incidents.
Warning	Indicates there may or may not be a problem related to the associated object.
Minor	Indicates NNMi has detected problems related to the associated object that require further investigation.
Major	Indicates NNMi has detected problems related to the associated object to be resolved before they become critical.
Critical	Indicates NNMi has detected problems related to the associated object that require immediate attention.

See "Monitor Incidents for Problems" for more information about these severity values.

Specify Your Incident Message Format (SNMP Trap Incident)

When configuring an incident, specify the information you want NNMi to include in the incident's Message attribute value. You can use any combination of valid parameter strings and Custom Incident attributes to configure the Message.

Note: The incident Message limit is 1024 characters. If the returned values exceed this limit, NNMi truncates the value starting from the end of the returned text string.

"Valid Parameters for Configuring Incident Messages (SNMP Trap Incident)" (on page 446)

"Include Custom Incident Attributes in Your Message Format (SNMP Trap Incident)" (on page 452)

Valid Parameters for Configuring Incident Messages (SNMP Trap Incident)

When configuring incident messages, you may want to use incident information as part of the message. NNMi provides the following parameter values. Use these parameters as variables when formatting an incident message.

Tip: See the Using the Incident Form for more information about the parameter values.

Note: NNMi stores varbind values as custom incident attributes (CIAs).

See "Specify Your Incident Message Format (SNMP Trap Incident)" (on page 446) for more information about configuring messages.

Parameter strings are available for the following:

- Incident form attributes (Click here for a list of choices.)

Parameter Strings for Attributes from an Incident Form

Parameter String	Description
$category, $cat	Value of the Category attribute in the Incident form.
$count, $cnt	Value representing the number of Custom Incident Attributes that appear in the Incident form.
$family, $fam	Value from the Family attribute in the Incident form.
$firstOccurrenceTime, $fot	Value from the First Occurrence Time attribute in the incident form.
$lastOccurrenceTime, $lot	Value from the Last Occurrence Time attribute in the incident form.
$lifecycleState, $lcs	Value from the Lifecycle State attribute in the Incident form.
$name	Value of the Name attribute from the incident configuration.
$nature, $nat	Value from the Nature attribute in the Incident form.
$origin, $ori	Value from the Origin attribute in the Incident form.
$originOccurrenceTime, $oot	Value from the Origin Occurrence Time attribute in the incident form.
$priority, $pri	Value from the Priority attribute in the Incident form.
$sev, $severity	Value of the Severity attribute of the Incident form.

- Node form attributes (Click here for a list of choices.)

Parameter Strings for Attributes from a Node Form

Parameter String	Description
$managementAddress, $mga	Value from the Management Address attribute of the incident's source Node's form or SNMP Agent form.
$otherSideOfConnectionManagementAddress, $oma	If the incident's Source Node is part of a Layer 2 connection, this attribute is the value of the Management Address of a node on the other side of the Layer 2 connection.
$sourceNodeLongName, $sln	The fully-qualified DNS name as displayed in the Hostname attribute of the incident's source Node's form.
$sourceNodeName, $snn	Value from the Name attribute of the incident's source Node's form.
$sysContact, $sct	Value from the System Contact attribute of the incident's source Node form: General tab.

Parameter String	Description
$sysLocation, $slc	Value from the System Location attribute of the incident's source Node form: General tab.

- Interface form attributes (Click here for a list of choices.)

Parameter Strings for Attributes from an Interface Form

Parameter String	Description
$ifAlias, $ifa	Value from the IfAlias attribute for the interface that is the incident's source object.
$ifConfigDupSetting, $icd	Configured Duplex Setting on the port associated with the interface that is the incident's source object.
$ifDesc, $idc	Value from the ifDesc attribute for the interface that is the incident's source object.
$ifIndex, $idx	Value from the ifIndex attribute for the interface that is the incident's source object.
$ifIpAddr, $iia	IP Address values associated with the interface that is the incident's source object. If multiple IPaddresses are associated with the interface, this parameter returns a comma-separated list.
$ifName, $ifn	Value from the ifName attribute for the interface that is the incident's source object.
$ifPhysAddr, $ipa	Value from the Physical Address attribute for the interface that is the incident's source object.
$ifSpeed, $isp	Value from the ifSpeed attribute for the interface that is the incident's souce object.
$ifType, $itp	Value from the ifType attribute for the interface that is the incident's souce object.

- Layer 2 Connection form attributes (Click here for a list of choices.)

Parameter Strings for Attributes from a Layer 2 Connection Form

Parameter String	Description
$otherSideOfConnectionConfigDupSetting, $ocd	If the incident's source Node is part of a Layer 2 connection, this parameter contains the Configured Duplex Setting on the port associated with the interface on the other side of the connection.
$otherSideOfConnectionIfAlias, $oia	If the incident's Source Node is part of a Layer 2 connection, this parameter is the value of the ifAlias of one of the interfaces on the other side of the Layer 2 connection.
$otherSideOfConnectionIfDesc, $odc	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifDescr attribute value for the interface on the other side of the Layer 2 connection.

Parameter String	Description
$otherSideOfConnectionIfIndex, $odx	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifIndex attribute value for the interface on the other side of the connection.
$otherSideOfConnectionIfName, $ofn	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifName attribute value for the interface on the other side of the connection.

- VLAN form attributes (Click here for a list of choices.)
  Parameter Strings for Attributes from a VLAN Form

Parameter String	Description
$impVlanIds, $ivi	Value from the VLAN Id attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Port tab and open the form for the VLAN of interest. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.
$impVlanNames, $ivn	Value from the VLAN Name attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Ports tab of the Interface form. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.

- Additional information that is not visible in any form (Click here for a list of choices.)
  Parameter Strings for Attributes not visible in any Form

Parameter String	Description
$id	Unique Object Identifier attribute value for the incident (unique across the entire NNMi Database).
$firstOccurrenceTimeMS, $fms	Value from the First Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$lastOccurrenceTimeMS, $lms	Value from the Last Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$oid	Value of the unique object identifier (oid) for the incident configuration that originated from either an SNMP trap, Management Event, or Remote NNM 6.x or 7.x event.
$otherSideOfConnection, $osc	If the incident's Source Node is part of a Layer 2 connection, this attribute is the following combination of values for the node and one of its interfaces on the other side of the Layer 2 connection: The fully-qualified DNS name of the node appended with the interface Name in the following format: <*fully-qualified DNS name*>[*interface_name*]

Parameter String	Description
$originOccurrenceTimeMS, $oms	Value from the Origin Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$sourceNodeUuid, $snu	Universally Unique Object Identifier attribute value of the source node object for the incident (unique across all databases). This identifier distinguishes the source node object instance from all other node objects.
$sourceObjectClass, $soc	Value of the object class for the object you want to include. Use this parameter when you need to request more details of a class of objects through a web service. Examples of object classes include: `com.hp.ov.nms.model.core.Interface` and `com.hp.ov.nms.model.snmp.SnmpAgent`.
$sourceObjectName, $son	Value from the Name attribute of the source object. For example, an interface object is named according to the MIB ifName. Each ifName varies according to the vendor's conventions. Using the name `4/1` as an example, `4` represents `the board number and 1` represents the port number.
$sourceObjectUuid, $sou	Universally Unique Object Identifier attribute value of the source object for the incident (unique across all databases). This identifier distinguishes the source object instance from all other similar object instances..
$uuid	Universally Unique Object Identifier attribute value of the incident (unique across all databases). This identifier distinguishes the incident object instance from all other incident objects.

- Information established in Custom Incident Attributes (Click here for a list of choices.)

Parameter Strings for Attributes Established in Custom Incident Attributes

Parameter String	Description
$<position_number>	Value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1` NNMi stores varbind values as Custom Incident Attributes. If you know the varbind position number, use this parameter.
$<CIA_name>	Value of the name that is used for the custom incident attribute. For example, `$mycompany.mycia`. NNMi provides CIA values for configuring Management Events. See Custom Incident Attributes Provided by NNMi for more information about custom incident attributes.
$<CIA_oid>	Value of the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this parameter when you are not certain of a custom incident attribute (varbind) position number.
$*	Used to indicate you want all of the custom incident attribute values originating as varbinds, to be passed to the action configuration. Each varbind is returned in the

Parameter String	Description
	following format: `$<CIA_name>:<CIA_value>` in which the custom incident attribute name appears followed by the custom incident attribute value.

- Functions to generate values (Click here for a list of choices.)

  The function described in the following table replaces the specified numeric value with the associated text value stored in the CIA.

  Note: The associated MIB must have been loaded using the nnmloadmib.ovpl command.

Functions to Generate Values Within the Incident Message

Function	Description
$oidtext($<position_ number>)	A <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, `$oidtext($2)`. **Note**: The position number you enter must represent a CIA that contains an Object Identifier (OID) value. NNMi returns the textual value of the OID for the CIA specified. Note the following: ■ If the MIB is not loaded, NNMi returns the numeric OID value. ■ If the OID has a MIB instance, the number representing the MIB instance is appended to the textual OID value.
$oidtext($<CIA_ oid>)	The <*CIA_oid*> argument specifies the Object Identifier (OID) for any custom incident attribute that originated as a varbind. For example, `$oidtext($.1.3.6.1.6.3.1.1.5.1.)` Use this argument to the $oidtext() function when you are not certain of a custom incident attribute (varbind) position number. NNMi replaces the numeric value with the textual value of the OID you specify. Note the following: ■ If the MIB is not loaded, NNMi returns the numeric OID value. ■ If the OID has a MIB instance, the number representing the MIB instance is appended to the textual OID value.
$text($<position_ number>)	The <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1`. NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.
$text($<CIA_oid>)	The <CIA_oid> argument specifies the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this argument to the $text function when you are not certain of a custom incident attribute (varbind) position number.

Function	Description
	NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.

Include Custom Incident Attributes in Your Message Format (SNMP Trap Incident)

NNMi includes two categories of CIAs:

- SNMP trap varbinds identified by the Abstract Syntax Notation value (ASN.1). NNMi turns varbinds into CIAs and maintains each varbind's position number. See "Load SNMP Trap Incident Configurations" (on page 429).
- Custom attributes provided by NNMi. See "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312).

You can use CIAs in your message format to extend the amount of information presented. To determine which CIAs are available for any particular incident type, open an Incident view, locate the incident and open the Incident form. Navigate to the **Custom Attributes** tab. A complete list of available CIAs (for that incident type) appears in the table.

To include a CIA in your message format, type the dollar-sign character (`$`) plus any of the following:

- Varbind position number or asterisk (*) to include all varbind values
- Name of the CIA
- Object identifier (oid) of the CIA (useful when the varbind position number is not consistent among vendors)

Note: A single incident cannot include two CIAs with the same name. However, two incidents can contain CIAs having the same names and values.

The following table presents some example formats with the subsequent output.

Example Incident Message Formats

Example Message Format	Output in Incident View
Possible trouble with $3	`Possible trouble with` <varbind 3>
Possible trouble with $11	`Possible trouble with` <varbind 11>
Possible trouble with $77 (where the varbind position 77 does not exist)	`Possible trouble with <Invalid or unknown cia> 77`
Possible trouble with $*	`Possible trouble with` <cia1_name: cia_value>, <cia2_name; cia_value>,< cia*n*_name: cia_value>
Possible trouble with $3x	`Possible trouble with` <varbind 3>`x`
Possible trouble with $1.2.3.4.5	`Possible trouble with` <value of the CIA with oid of 1.2.3.4.5>
Possible trouble with $mycia.mycompany	`Possible trouble with` <value of the CIA with name of mycia.mycompany>

Tip: NNMi provides an error message when a CIA cannot be found. For example, if you enter an unavailable varbind position, name, or object identifier (oid), NNMi returns an "Invalid or unknown cia" error message.

Specify a Description for Your Incident Configuration (SNMP Trap Incident)

NNMi provides the Description attribute to help you further identify the current incident configuration.

Description

Use the description field to provide additional information that you would like to store about the current incident configuration. This description applies only to the configuration entry.

Type a maximum of 2048 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Configure Interface Settings for an SNMP Trap Incident

NNMi enables you to apply a Suppression, Enrichment, Dampen, or Actions incident configuration to a Source Object based on the Source Object's participation in an Interface Group.

Note: Interface Settings override any other Suppression, Enrichment, Dampen, or Actions configuration settings for this incident, including those configured on the Node Settings tab.

Tip: See "Create Interface Groups" (on page 197) for more information about Interface Groups.

For information about each Interface Settings tab:

For information about each SNMP Traps tab:

To apply an incident configuration to a Source Object based on the Source Object's Interface Group:

1. Navigate to the **SNMP Trap Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **SNMP Traps** tab .
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
4. Configure the desired Interface Settings (see table).

5. Configure any Suppression, Dampen, or Enrichment settings for this Interface Group.
6. Click **Save and Close** to save your changes and return to the Incident Configuration form.

Interface Group Attributes

Name	Description
Interface Group	Click the Lookup icon and select Quick Find to select the Interface Group you want to use. See "Use the Quick Find Window" (on page 28) for more information about using Quick Find.
Ordering	Determines the priority order for those interfaces that appear in multiple Interface Groups. The lower the number, the higher the priority. For example, **1** is the highest priority. If an interface is in multiple Interface Groups and more than one of those Interface Groups have been specified in an incident configuration, only the incident configuration with the highest priority will be applied to the interface.
Enable	Use this attribute to temporarily disable an incident's configuration settings. To temporarily disable the Interface Group settings for the selected incident configuration, clear **Enable** ☐. To enable the Interface Group settings for the selected incident configuration, click **Enable** ☑.

Related Topics

"Configure Node Settings for an SNMP Trap Incident" (on page 488)

Configure Incident Suppression Settings for an Interface Group (SNMP Trap Incident)

Note: Interface Settings override any other Suppression settings for this incident, including those from the Node Settings tab.

NNMi enables you to suppress a specified incident configuration based on the Source Object's participation in an Interface Group. When an incident is suppressed:

- It is not stored in the NNMi database
- It does not appear in an incident view in the NNMi console

You can also suppress the incident configuration based on either of the following:

- Source Node's participation in a Node Group. See "Configure Incident Suppression Settings for a Node Group (SNMP Trap Incident)" (on page 489) for more information.
- Incident configuration default settings without specifying a Node or Interface Group. See "Configure Suppression Settings for an SNMP Trap Incident" (on page 529) for more information.

Tip: See "Create Interface Groups" (on page 197) for more information about Interface Groups.

For information about each Interface Settings tab:

To suppress an incident configuration based on an Interface Group:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.

c. Select the **SNMP Traps** tab.

d. Do one of the following:

i. To create an incident configuration, click the New icon, and continue.

ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

iii. To delete an incident configuration, click the Delete icon.

2. Select the **Interface Settings** tab.

3. Do one of the following:

a. To create a new configuration, click the New icon.

b. To edit an existing configuration, select a row, and click the Open icon.

4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for an SNMP Trap Incident" (on page 453) for more information.

5. Select the **Suppression** tab.

6. Configure the desired Suppression behavior (see table).

7. Click **Save and Close** to save your changes and return to the previous form.

Interface Settings Suppression Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings for the specified Interface Group. To temporarily disable the Suppression settings for the selected incident configuration, clear **Enable** ☐. To enable the Suppression settings for the selected incident configuration, click **Enable** ☑.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor. When creating a Payload Filter, note the following: • Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class). • You must use a `ciaName` that already exists in the trap or event you are configuring. • Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together. • View the expression displayed under **Filter String** to see the logic of the expression as it is created. • The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below. The following example filters incidents on voltage state: `AND`

Name	Description

```
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
ciaValue = 5
```

NNMi evaluates the expression above as follows:

```
(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)
```

NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.

- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

information. Click here for more information.

The period asterisk (.*) characters mean *any number of characters of any type at this location*.

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

information. Click here for an example.

The period asterisk (.*) characters mean *any number of characters of any type at this location*.

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

information. Click here for an example.

The period asterisk (.*) characters mean *any number of characters of any type at this location*.

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

information. Click here for an example.

The period asterisk (.*) characters mean *any number of characters of any type at this location*.

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.

Name	Description

Button	Description
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.

Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples.
	
	Example 1
	
	`AND`
	`  ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6`
	`  OR`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3`
	
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:
	
	`AND`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6`
	`  OR`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3`
	
	Example 2
	
	`AND`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`

Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Name Descr

Configure Incident Enrichment Settings for an Interface Group (SNMP Trap Incident)

Note: Interface Settings override any other Enrichment settings for this incident, including those from the Node Settings tab.

NNMi enables you to fine tune and enhance a specified incident configuration based on the Source Object's participation in an Interface Group. The types of items you can fine tune and enhance for a selected incident configuration, include:

- Severity
- Priority
- Category
- Family
- Correlation Nature
- Message
- Assigned To

You can also enrich the incident configuration based on either of the following:

- The incident configuration based on the Source Node's participation in a Node Group. See "Configure Incident Enrichment Settings for a Node Group (SNMP Trap Incident)" (on page 501) for more information.
- Incident configuration default settings without specifying a Node or Interface Group. See "Configure Enrichment Settings for an SNMP Trap Incident" (on page 542) for more information.

Tip: See Create Interface Groups for more information about Interface Groups.

For information about each Interface Settings tab:

For information about each Enrichment tab:

To enrich an incident configuration based on an Interface Group:

1. Navigate to the **SNMP Trap Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **SNMP Traps** tab**.**
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Interface Settings** tab.

3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for an SNMP Trap Incident" (on page 453) for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the New icon, and continue.
   b. To edit an Enrichment configuration, click the Open icon, and continue.
   c. To delete an Enrichment configuration, click the Delete icon.
7. Configure the desired Enrichment behavior (see the "Enrich Configuration Attributes" table)
8. Click **Save and Close** to save your changes and return to the previous form.

Interface Settings Enrich Configuration Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: • Accounting • Application Status • Configuration • Fault • Performance • Security • Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443) for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Possible values include: • Address • Aggregated Port • Board • Connection • Correlation • HSRP • Interface

Name	Description
	• Node • OSPF See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443) for more information.
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below: **Normal** - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there may or may not be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: 5 **None** 4 **Low** 3 **Medium** 2 **High** 1 **Top** **Note**: The icons are displayed only in table views.

Name	Description
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact See Incident Form: General Tab for more information.
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right. You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (SNMP Trap Incident)" (on page 446) "Include Custom Incident Attributes in Your Message Format (SNMP Trap Incident)" (on page 452)
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 2048 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Configure Custom Incident Attributes to Enrich an Incident Configuration (Interface Settings) (SNMP Trap Incidents)

The Custom Incident Attributes (CIAs) tab enables you to specify additional CIAs to be saved with an incoming incident. The persisted data might then be used as an argument to an action for the incident.

When creating a CIA for an incident configuration, you can specify any of the following values:

- Custom Attribute on the source node
- Custom Attribute on the interface (source object)
- Object Attribute (Object Attribute Path)

When specifying an Object Attribute, use the same syntax as when specifying a URL object for an Object-Type Attribute. See "Attributes per Object Type for Full URLs" (on page 883) for more information.

You also specify the Custom Incident Attribute name that will store this information.

For information about each Enrichment tab:

To create a Custom Incident Attribute to enrich an incident configuration:

1. Navigate to the **SNMP Trap Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **SNMP Traps** tab**.**
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for an SNMP Trap Incident" (on page 453) for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
    a. To create an Enrichment configuration, click the New icon, and continue.
    b. To edit an Enrichment configuration, click the Open icon, and continue.
    c. To delete an Enrichment configuration, click the Delete icon.
5. Make sure the Enrichment settings are configure. See "Configure Incident Enrichment Settings for an Interface Group (SNMP Trap Incident)" (on page 462) for more information.
8. Navigate to the **Custom Incident Attributes** tab.
9. Do one of the following:
    a. To create a Custom Incident Attribute, click the New icon, and continue.
    b. To edit a Custom Incident Attribute, click the Open icon, and continue.
    c. To delete a Custom Incident Attribute, click the Delete icon.
10. Configure the Custom Incident Attribute (see the "Custom Incident Attribute" table).
11. Click **Save and Close** to save your changes and return to the previous form.

Custom Incident Attribute

Name	Description
Type	Specifies whether you are using a Custom Attribute on a node, a Custom Attribute on an interface, or an Object Attribute. Possible values are: • Node Custom Attribute • Interface Custom Attribute • Object Attribute Path
Custom Attribute Name or Object Attribute Path	Used to determine the value to be assigned to the Custom Incident Attribute you are configuring. Enter either of the following: • Name of the Custom Attribute on the source node • Name of the Custom Attribute on the interface (source object) • Object Attribute Path When specifying an Object Attribute Path, use the same syntax as when specifying a URL object for an Object-Type Attribute; for example ${deviceProfile.SNMPObjectID}. See "Attributes per Object Type for Full URLs" (on page 883) for more information.
Custom Incident Attribute Name	Name used to identify the Custom Incident Attribute you are configuring. The name limit is 80 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. No spaces are allowed.

Configure a Payload Filter to Enrich an Incident Configuration (Interface Settings) (SNMP Trap Incidents)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be enriched. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **SNMP Trap Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **SNMP Traps** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:

a. To create a new configuration, click the New icon.

b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.

4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for an SNMP Trap Incident" (on page 453) for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:

   a. To create an Enrichment configuration, click the New icon, and continue.

   b. To edit an Enrichment configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

   c. To delete an Enrichment configuration, click the Delete icon.

7. Make sure the Enrichment settings are configured. See "Configure Incident Enrichment Settings for an Interface Group (SNMP Trap Incident)" (on page 462) for more information.
8. Select the **Payload Filter** tab.
9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

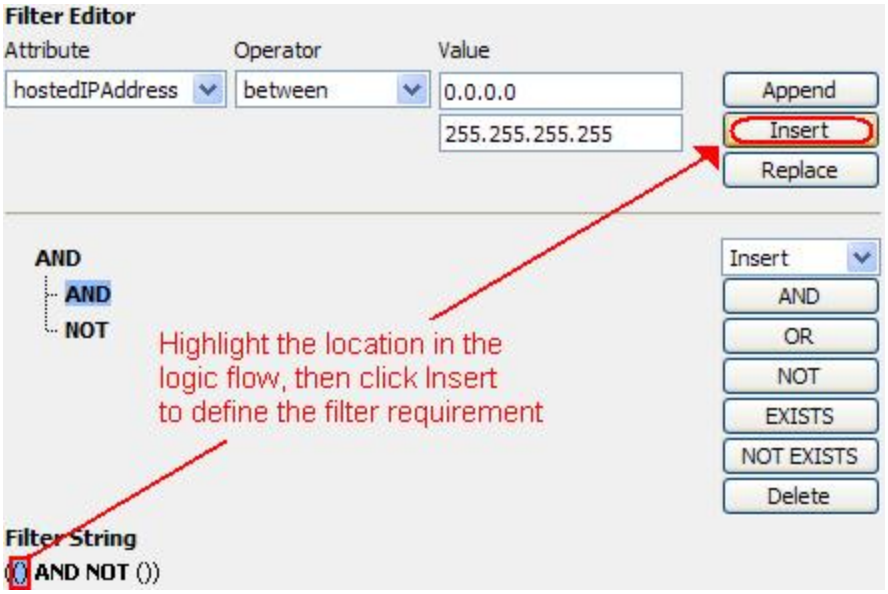
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



10. Click **Save and Close**.
11. Click **Save and Close** to save your changes and return to the previous form.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Operator	Valid operators are described below. ■ **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. ■ **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. ■ **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**.

Attribute	Description

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not have a value.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in

Attribute	Description
	\Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like *Chicago*` finds all traps or events that contain a varbind value that includes the string **Chicago**. ■ **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. ■ **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in` matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. ■ **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Value	The value for which you want NNMi to search. Note the following: ■ The values you enter are case sensitive. ■ NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. ■ The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Incident Dampening Settings for an Interface Group (SNMP Trap Incident)

Note: Interface Settings override any other Dampening settings for this incident, including those from the Node Settings tab.

NNMi enables you to delay the following for an incident configuration based on the Source Object's participation in an Interface Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

You can configure the Dampening settings based on either of the following:

- The Source Node's participation in a Node Group. See "Configure Incident Dampening Settings for a Node Group (SNMP Trap Incident)" (on page 511) for more information.
- Incident configuration default settings without specifying a Node or Interface Group. See "Configure Dampening Settings for an SNMP Trap Incident" (on page 545) for more information.

Tip: See "Create Interface Groups" (on page 197) for more information about Interface Groups.

For information about each Interface Settings tab:

When using the Dampening configuration, note the following:

- By default, the Dampen Interval for SNMP Trap Incidents is six minutes.

  Note: Any Dampen Interval set for less than six minutes might not provide enough time for the Causal Engine to complete its analysis. As a result, incidents associated with intermediate symptoms might appear as root cause incidents.

- After the Dampen Interval has passed, NNMi changes the Lifecycle State to REGISTERED.

See About the Incident Lifecycle for more information about Lifecycle State.

To configure Dampening settings based on an Interface Group:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **SNMP Traps** tab**.**
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete an incident configuration, click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for an SNMP Trap Incident" (on page 453) for more information.
5. Select the **Dampening** tab.
6. Configure the desired Dampening behavior (see table).

7. Click **Save and Close** to save your changes and return to the previous form.

Interface Settings Dampening Configuration Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's dampening settings. To temporarily disable the Dampening Configuration settings for the selected incident configuration, clear **Enable** ☐. To enable the Dampening Configuration settings for the selected incident configuration, click **Enable** ☑.
Hour	Specifies the number of hours to be used for the dampen interval.
Minutes	Specifies the number of minutes to be used for the dampen interval. **Note**: The maximum dampen interval is 60 minutes.
Seconds	Specifies the number of seconds to be used for the dampen interval.

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	
	When creating a Payload Filter, note the following:
	
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	
	The following example filters incidents on voltage state:
	
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`ciaValue = 5`
	
	NNMi evaluates the expression above as follows:
	
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Name	Description

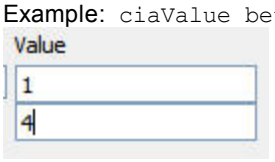
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all values equal to and between the two values specified. Click here for an example.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

Example: `ciaValue between`



matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line. selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line. selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.
Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples. **Example 1** `AND`

Name	Description

Button	Description

```
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Name	Description

Button	Description

	Example 2
	`AND`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Example 2
	`AND`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Example 2
	`AND`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`

Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Name Desc

Configure Incident Actions for an Interface Group (SNMP Trap Incident)

Note: Interface Settings override any other Actions settings for this incident, including those from the Node Settings tab.

For information about each Interface Settings tab:

NNMi enables you to configure incident actions based on a Source Object's participation in an Interface Group.

You can also configure incident actions based on either of the following:

- The Source Node's participation in a Node Group. See "Configure Incident Actions for a Node Group (SNMP Trap Incident)" (on page 521) for more information.
- Incident configuration default settings without specifying a Node or Interface Group. See "Configure Actions for an SNMP Trap Incident" (on page 571) for more information.

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 (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

You can configure actions for incidents generated from SNMP traps, NNM 6.x or 7.x events, and the NNMi management events. Any time an incident configuration changes, the action directory is rescanned and any executable or script files (for example, Jython) are reloaded to the NNMi database. See "Lifecycle Transition Action Form (SNMP Trap Incidents)" (on page 572) for more information about the actions directory.

Tip: Copy any required executable or script files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (SNMP Trap Incidents)" (on page 572) for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. See "Verify that NNMi Services are Running" (on page 47) for more information about log files and where they are located.

To configure an automatic action for an incident:

1. Navigate to the **SNMP Trap Configuration** tab.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration.**
    c. Select the **SNMP Traps** tab.
    d. Do one of the following:
        i. To create a new incident configuration, click the New icon.
        ii. To edit an existing incident configuration, click the Open icon in the row representing the configuration you want to edit.
2. Select the **Interface Settings** tab.

3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
4. Make sure the basic Interface Setting behavior is configured. See "Configure Interface Settings for an SNMP Trap Incident" (on page 453) for more information.
5. Select the **Actions** tab.
6. From the **Lifecycle Actions** table toolbar, do one of the following:
    - To create an Action configuration, click the New icon, and continue.
    - To edit an Action configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
    - To delete an Action configuration, click the Delete icon.
7. In the "Lifecycle Transition Action Form (SNMP Trap Incidents)" (on page 572), provide the required information.
8. Click **Save and Close** to save your changes and return to the **SNMP Trap Configuration** form.

    The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type .

Configure a Payload Filter for an Incident Action (Interface Settings) (SNMP Trap Incidents)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **SNMP Trap Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **SNMP Traps** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:

a. To create a new configuration, click the New icon.

b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.

4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for an SNMP Trap Incident" (on page 453) for more information.
5. Select the **Actions** tab.
6. Do one of the following:

   a. To create an Action configuration, click the New icon, and continue.

   b. To edit an Action configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

   c. To delete an Action configuration, click the Delete icon.

7. Make sure you configure the Action Configuration settings. See "Configure Incident Actions for an Interface Group (SNMP Trap Incident)" (on page 482) for more information.
8. Select the **Payload Filter** tab.
9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

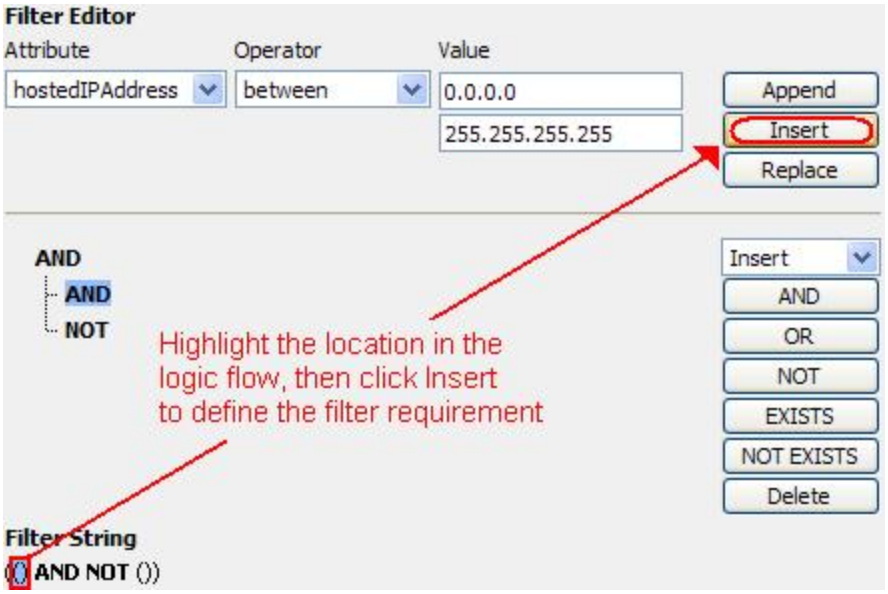
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



10. Click **Save and Close**.
11. Click **Save and Close** to save your changes and return to the previous form.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Operator	Valid operators are described below. ■ **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. ■ **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. ■ **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**.

Attribute	Description

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not have a value.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in

Attribute	Description
	\Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like *Chicago*` finds all traps or events that contain a varbind value that includes the string **Chicago**. ■ **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. ■ **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in` matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. ■ **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Value	The value for which you want NNMi to search. Note the following: ■ The values you enter are case sensitive. ■ NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. ■ The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Node Settings for an SNMP Trap Incident

NNMi enables you to apply a Suppression, Enrichment, Dampen, Actions, or Diagnostics Selections incident configuration to a Source Node based on the Source Node's participation in a Node Group.

Note: Node Settings override any other Suppression, Enrichment, Dampen, Actions, or Diagnostics Selections configuration settings for this incident, except those configured on the Interface Settings tab.

Tip: See "Create Node Groups" (on page 181) for more information about Node Groups.

For information about each Node Settings tab:

For information about each SNMP Traps tab:

To apply an incident configuration to a Source Node based on the Source Node's Node Group:

1. Navigate to the **SNMP Trap Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **SNMP Traps** tab**.**
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.
4. Configure the desired Node Settings (see table).
5. Click **Save and Close** to save your changes and return to the Incident Configuration form.

Node Group Attributes

Name	Description
Node Group	Click the Lookup icon and select Quick Find to select the Node Group you want to use. See "Use the Quick Find Window" (on page 28) for more information about using Quick Find.
Ordering	Determines the priority order for those nodes that appear in multiple Node Groups. The lower the number, the higher the priority. For example, **1** is the highest priority. If a node is in multiple Node Groups and more than one of those Node Groups have been specified in an incident configuration, only the incident configuration with the highest priority will be applied to the node.
Enable	Use this attribute to temporarily disable an incident's suppression settings. To temporarily disable the Node Group settings for the selected incident configuration, clear **Enable** ☐. To enable the Node Group settings for the selected incident configuration, click **Enable** ☑.

Configure Incident Suppression Settings for a Node Group (SNMP Trap Incident)

Note: Node Settings override any other Suppression settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to suppress a specified incident configuration based on the Source Node's participation in a Node Group. When an incident is suppressed:

- It is not stored in the NNMi database
- It does not appear in an incident view in the NNMi console

You can also suppress the incident configuration based on either of the following:

- The Source Object's participation in an Interface Group. See "Configure Incident Suppression Settings for an Interface Group (SNMP Trap Incident)" (on page 454) for more information.
- Incident configuration default settings without specifying a Node or Interface Group. See "Configure Suppression Settings for an SNMP Trap Incident" (on page 529) for more information.

Tip: See "Create Node Groups" (on page 181) for more information about Node Groups.

For information about each Node Settings tab:

To suppress an incident configuration based on a Node Group:

1. Navigate to the **SNMP Trap Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **SNMP Traps** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for an SNMP Trap Incident" (on page 488) for more information.
5. Select the **Suppression** tab.
6. Configure the desired Suppression behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Suppression Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings. To temporarily disable the Suppression settings for the selected incident configuration, clear **Enable**☐. To enable the Suppression settings for the selected incident configuration, click **Enable**☑.

Name	Description

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	
	When creating a Payload Filter, note the following:
	
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	
	The following example filters incidents on voltage state:
	
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`ciaValue = 5`
	
	NNMi evaluates the expression above as follows:
	
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**.

- **<** Finds all values less than the value specified. Click here for an example.

  Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**.

- **<=** Finds all values less than or equal to the value specified. Click here for an example.

  Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

  Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

  `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

  varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

Example:

`ciaValue not in`

matches any incident that contains a varbind with values other than **1** and **2**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

  Example:

  `ciaValue not in`

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

matches any incident that contains a varbind with values other than **1** and **2**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line. selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line. selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.
Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples. **Example 1**

Name	Description

Button	Description

```
AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
```

Name	Description

Button	Description
	OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Name	Description

Configure Incident Enrichment Settings for a Node Group (SNMP Trap Incident)

Note: Node Settings override any other Enrichment configuration for this incident, except those configured on the Interface Settings tab.

NNMi enables you to enhanced a specified incident configuration based on the Source Node's participation in a Node Group. The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To

You can also enrich the incident configuration based on either of the following:

- The Source Object's participation in an Interface Group. See "Configure Incident Enrichment Settings for an Interface Group (SNMP Trap Incident)" (on page 462) for more information.
- Incident configuration default settings without specifying a Node or Interface Group. See "Configure Enrichment Settings for an SNMP Trap Incident" (on page 542) for more information.

For information about each Node Settings tab:

For information about each Enrichment tab:

To configure enrichment settings for a Node Group:

1. Navigate to the **SNMP Trap Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **SNMP Traps** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.

4. Make sure the basic Node Setting behavior is configured. See "Configure Node Settings for an SNMP Trap Incident" (on page 488)for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the New icon, and continue.
   b. To edit an Enrichment configuration, click the Open icon, and continue.
   c. To delete an Enrichment configuration, click the Delete icon.
7. Configure the desired Enrichment behavior (see the "Enrich Configuration Attributes" table)
8. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Enrich Configuration Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: • Accounting • Application Status • Configuration • Fault • Performance • Security • Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443) for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Possible values include: • Address • Aggregated Port • Board • Connection • Correlation • HSRP • Interface • Node • OSPF See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443) for more information.

Name	Description
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below: **Normal** - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there may or may not be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: 5 **None** 4 **Low** 3 **Medium** 2 **High** 1 **Top** **Note**: The icons are displayed only in table views.
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact See Incident Form: General Tab for more information.

Name	Description
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right. You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (SNMP Trap Incident)" (on page 446) "Include Custom Incident Attributes in Your Message Format (SNMP Trap Incident)" (on page 452)
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 2048 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Configure Custom Incident Attributes to Enrich an Incident Configuration (Node Settings) (SNMP Trap Incidents)

The Custom Incident Attributes (CIAs) tab enables you to specify additional CIAs to be saved with an incoming incident. The persisted data might then be used as an argument to an action for the incident.

When creating a CIA for an incident configuration, you can specify any of the following values:

- Custom Attribute on the source node
- Custom Attribute on the interface (source object)
- Object Attribute (Object Attribute Path)

When specifying an Object Attribute, use the same syntax as when specifying a URL object for an Object-Type Attribute. See "Attributes per Object Type for Full URLs" (on page 883) for more information.

You also specify the Custom Incident Attribute name that will store this information.

For information about each Enrichment tab:

To create a Custom Incident Attribute to enrich an incident configuration:

1. Navigate to the **SNMP Trap Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **SNMP Traps** tab.
    d. Do one of the following:

i. To create an incident configuration, click the New icon, and continue.

ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

iii. To delete an incident configuration, click the Delete icon.

2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, and click the Open icon.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for an SNMP Trap Incident" (on page 488)for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the New icon, and continue.
   b. To edit an Enrichment configuration, click the Open icon, and continue.
   c. To delete an Enrichment configuration, click the Delete icon.
7. Make sure the Enrichment settings are configured. See "Configure Incident Enrichment Settings for a Node Group (SNMP Trap Incident)" (on page 501) for more information.
8. Navigate to the **Custom Incident Attributes** tab.
9. Do one of the following:
   a. To create a Custom Incident Attribute, click the New icon, and continue.
   b. To edit a Custom Incident Attribute, click the Open icon, and continue.
   c. To delete a Custom Incident Attribute, click the Delete icon.
10. Configure the Custom Incident Attribute (see the "Custom Incident Attribute" table).
11. Click **Save and Close** to save your changes and return to the previous form.

Custom Incident Attribute

Name	Description
Type	Specifies whether you are using a Custom Attribute on a node, a Custom Attribute on an interface, or an Object Attribute. Possible values are: • Node Custom Attribute • Interface Custom Attribute • Object Attribute Path
Custom Attribute Name or Object Attribute Path	Used to determine the value to be assigned to the Custom Incident Attribute you are configuring. Enter either of the following: • Name of the Custom Attribute on the source node • Name of the Custom Attribute on the interface (source object)

Name	Description
	• Object Attribute Path When specifying an Object Attribute Path, use the same syntax as when specifying a URL object for an Object-Type Attribute; for example ${deviceProfile.SNMPObjectID}. See "Attributes per Object Type for Full URLs" (on page 883) for more information.
Custom Incident Attribute Name	Name used to identify the Custom Incident Attribute you are configuring. The name limit is 80 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. No spaces are allowed.

Configure a Payload Filter to Enrich an Incident Configuration (Node Settings) (SNMP Trap Incidents)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be enriched. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **SNMP Trap Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **SNMP Traps** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.
4. Make sure the basic Node Setting behavior is configured. See "Configure Node Settings for an SNMP Trap Incident" (on page 488)for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
    a. To create an Enrichment configuration, click the New icon, and continue.

b. To edit an Enrichment configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

c. To delete an Enrichment configuration, click the Delete icon.

7. Make sure the Enrichment settings are configured. See "Configure Incident Enrichment Settings for a Node Group (SNMP Trap Incident)" (on page 501) for more information.
8. Select the **Payload Filter** tab.
9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

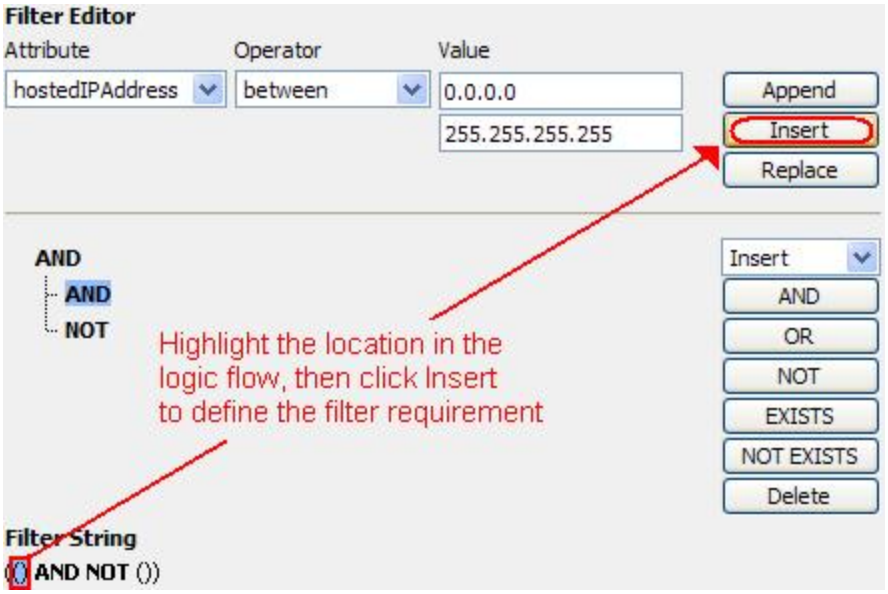
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



10. Click **Save and Close**.
11. Click **Save and Close** to save your changes and return to the previous form.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Operator	Valid operators are described below. ■ **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. ■ **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. ■ **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. ■ **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. ■ **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between` matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. ■ **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`

Attribute	Description

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not have a value.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Examples:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters.

  `ciaValue like *Chicago*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**.

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

Attribute	Description
	 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. ■ **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Value	The value for which you want NNMi to search. Note the following: ■ The values you enter are case sensitive. ■ NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. ■ The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Incident Dampening Settings for a Node Group (SNMP Trap Incident)

Note: Node Settings override any other Dampening settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to delay the following for an incident configuration based on the Source Node's participation in a Node Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

You can also configure Dampening settings based on either of the following:

- The Source Object's participation in an Interface Group. See "Configure Incident Dampening Settings for an Interface Group (SNMP Trap Incident)" (on page 472) for more information.
- Incident configuration default settings without specifying a Node or Interface Group. See "Configure Dampening Settings for an SNMP Trap Incident" (on page 545) for more information.

Tip: See "Create Node Groups" (on page 181) for more information about Node Groups.

For information about each Node Settings tab:

When using the Dampening configuration, note the following:

- By default, the Dampen Interval for SNMP Trap Incidents is six minutes.

  Note: Any Dampen Interval set for less than six minutes might not provide enough time for the Causal Engine to complete its analysis. As a result, incidents associated with intermediate symptoms might appear as root cause incidents.

- After the Dampen Interval has passed, NNMi changes the Lifecycle State to REGISTERED.

See About the Incident Lifecycle for more information about Lifecycle State.

To configure Dampening settings based on a Node Group:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **SNMP Traps** tab.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete an incident configuration, click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, and click the Open icon.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for an SNMP Trap Incident" (on page 488) for more information.
5. Select the **Dampen** tab.
6. Configure the desired Dampen behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Dampen Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's Dampening settings. To temporarily disable the Dampening settings for the selected incident configuration, clear **Enable** ☐. To enable the Dampening settings for the selected incident configuration, click **Enable** ☑.
Hour	Specifies the number of hours to be used for the dampen interval.
Minutes	Specifies the number of minutes to be used for the dampen interval. **Note**: Use a minimum of 6 minutes for the dampen interval. The maximum dampen interval is 60 minutes.
Seconds	Specifies the number of seconds to be used for the dampen interval.

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	
	When creating a Payload Filter, note the following:
	
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	
	The following example filters incidents on voltage state:
	
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`ciaValue = 5`
	
	NNMi evaluates the expression above as follows:
	
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Name	Description

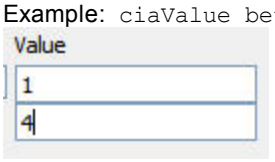
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all values equal to and between the two values specified. Click here for an example.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

Example: `ciaValue between`



matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line. selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line. selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.
Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples. **Example 1**

Name	Description

Button	Description

```
AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
```

Name	Description

Button	Description
	OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Name	Description

Configure Incident Actions for a Node Group (SNMP Trap Incident)

For information about each Node Settings tab:

Note: Node Settings override any other Actions settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to configure incident actions based on a Source Node's participation in a Node Group.

You can also configure incident actions based on either of the following:

- The Source Object's participation in an Interface Group. See "Configure Incident Actions for an Interface Group (SNMP Trap Incident)" (on page 482) for more information.
- Incident configuration default settings without specifying a Node or Interface Group. See "Configure Actions for an SNMP Trap Incident" (on page 571) for more information.

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 (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

You can configure actions for incidents generated from SNMP traps, NNM 6.x or 7.x events, and the NNMi management events. Any time an incident configuration changes, the action directory is rescanned and any executable or script files (for example, Jython) are reloaded to the NNMi database. See "Lifecycle Transition Action Form (SNMP Trap Incidents)" (on page 572) for more information about the actions directory.

Tip: Copy any required executable or script files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (SNMP Trap Incidents)" (on page 572) for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. See "Verify that NNMi Services are Running" (on page 47) for more information about log files and where they are located.

To configure an automatic action for an incident:

1. Navigate to the **SNMP Trap Configuration** tab.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration.**
    c. Select the **SNMP Traps** tab.
    d. Do one of the following:
        i. To create a new incident configuration, click the New icon.
        ii. To edit an existing incident configuration, click the Open icon in the row representing the configuration you want to edit.
2. Select the **Node Settings** tab.

3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
    c. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for an SNMP Trap Incident" (on page 488) for more information.
4. Select the **Actions** tab.
5. From the **Lifecycle Actions** table toolbar, do one of the following:
    - To create an Action configuration, click the New icon, and continue.
    - To edit an Action configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
    - To delete an Action configuration, click the Delete icon.
3. In the "Lifecycle Transition Action Form (SNMP Trap Incidents)" (on page 572), provide the required information.
4. Click **Save and Close** to save your changes and return to the **SNMP Trap Configuration** form.

    The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type .

Configure a Payload Filter for an Incident Action (Node Settings) (SNMP Trap Incidents)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **SNMP Trap Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **SNMP Traps** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:

a. To create a new configuration, click the New icon.

b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.

4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for an SNMP Trap Incident" (on page 488) for more information.
5. Select the **Actions** tab.
6. Do one of the following:

a. To create an Action configuration, click the New icon, and continue.

b. To edit an Action configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

c. To delete an Action configuration, click the Delete icon.

7. Make sure the Action Configuration settings are configured. See "Configure Incident Actions for a Node Group (SNMP Trap Incident)" (on page 521) for more information.
8. Select the **Payload Filter** tab.
9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

a. Plan out the logic needed for your Filter String.

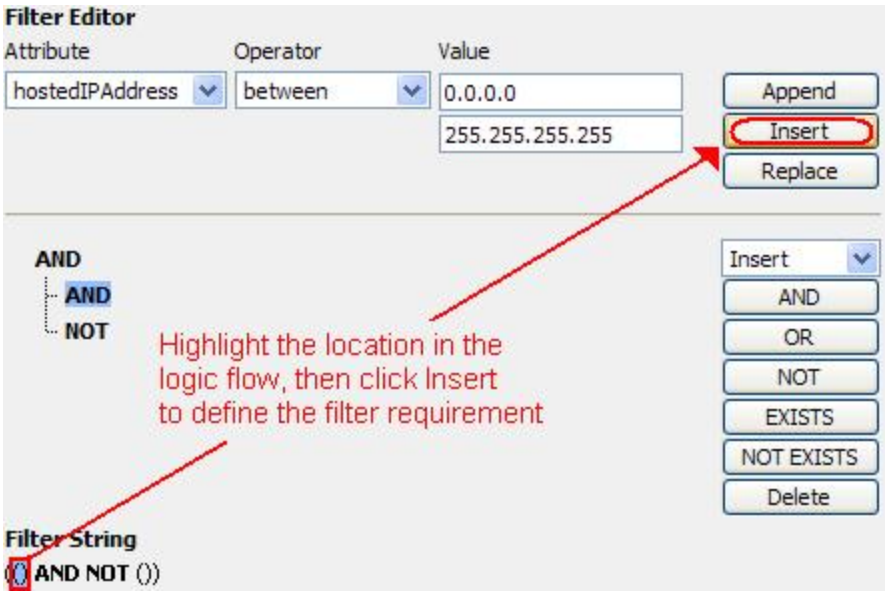
b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

```
(( ) AND NOT ( ))
```

c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



10. Click **Save and Close**.
11. Click **Save and Close** to save your changes and return to the previous form.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Operator	Valid operators are described below. ■ **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. ■ **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. ■ **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. ■ **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. ■ **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between` matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. ■ **in** Finds any match to at least one value in a list of values. Click here for an example. Example:

Attribute	Description

`ciaValue in`

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not have a value.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Examples:

  `ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters.

  `ciaValue like *Chicago*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**.

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

Attribute	Description
	 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. ■ **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Value	The value for which you want NNMi to search. Note the following: ■ The values you enter are case sensitive. ■ NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. ■ The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Diagnostics Selections for a Node Group (SNMP Trap Incident) (NNM iSPI NET)

For information about each Node Settings tab: .

Note: Node Settings override any other Diagnostics Selections settings for this incident, except those configured on the Interface Settings tab.

(*HP Network Node Manager iSPI Network Engineering Toolset Software*) The Diagnostic Selections form enables you to configure NNMi to automatically gather NNM iSPI NET diagnostic information for the

Incident you are configuring. When using this form, you specify the diagnostics you want to run on each applicable node in the specified Node Group.

To configure Diagnostics to run on a Source Node for an incident:

1. Navigate to the **Diagnostics Selection** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration.**
    c. Select the **SNMP Traps** tab.
    d. Do one of the following:
        - To create an Incident configuration, click the New icon, and continue.
        - To edit an Incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
    e. Navigate to **Node Settings** tab, and do one of the following:
        - To create a Node Settings configuration, click the New icon, and continue.
        - To edit a Node Settings configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        - To delete a Node Settings configuration, select the Node setting and click the Delete icon.
    f. Navigate to the **Diagnostic Selection** tab, and do one of the following:
        - To create a Diagnostic Selection setting, click the New icon, and continue.
        - To edit a Diagnostic Selection setting, click the Open icon in the row representing the Diagnostic Selection setting you want to edit, and continue.
        - To delete a Diagnostic Selection setting, select the Diagnostic Selection setting and click the Delete icon.
2. Provide the required information (see table).
3. Click **Save and Close** to save your changes and return to the **Node Settings** form.

After you configure the Diagnostic for the incident and Node Group, the Diagnostic must match the following criteria before the Diagnostic runs:

- The Source Node must be in the specified Node Group.
- The Diagnostic must be valid for the Source Node. (For example, only Nortel switch Diagnostics are run on Nortel switches.)
- The incident's current lifecycle state must match a lifecycle state for which it was configured. (For example, if you configure the Incident to run a specified Diagnostic when the incident is Closed, then if the current Incident's Lifecycle State is Closed, NNMi runs that Diagnostic.)

Note: If a Source Node is in more than one Node Group, the Diagnostic is only run on the node the first time NNMi finds a match for that Source Node based on the configuration Ordering field.

If these criteria are met, NNM iSPI NET runs the Diagnostics and generates Diagnostic reports to help you solve the problem on the Source Node.

After you configure Diagnostics for an incident, you can also run Diagnostics and access the Diagnostics reports on demand, using **Actions → Run Diagnostics (iSPI NET only)** in the Incident form. The same criteria apply (see the criteria above). See Incident Form:Diagnostics Tab for more information.

You can also run and access Diagnostics reports from a Node form. See Node Form: Diagnostics Tab for more information.

Diagnostic Settings Attributes

Attribute	Description
Flow Definition	Select the Diagnostic (Flow Definition) you want to use for the specified Node Group. Click the Lookup icon and choose one of the following options: • Quick View to view summary information for the Flow Definition name displayed. • Quick Find to view the list of possible diagnostic Flow Definitions. NNMi provides diagnostics for the following types of devices: ▪ Cisco switch ▪ Cisco router ▪ Cisco switch/router ▪ Nortel switch See "Diagnostics (Flows) Provided by NNM iSPI NET" (on page 422) for more information about the diagnostics provided and the devices to which they apply.
Lifecycle State	Incident Lifecycle State of the target Incident. If the incident's Lifecycle State matches the value specified here, the Diagnostic runs. The Diagnostic automatically runs on each applicable Source Node in the specified Node Group if the incident has the Lifecycle State currently configured in this attribute of the Diagnostic (Flow Definition - set of automated commands). See About the Incident Lifecycle for more information about Lifecycle State.
Enable	Use this attribute to temporarily disable an incident's Diagnostics settings. To temporarily disable the selected Diagnostics settings, clear **Enable**☐. To enable the selected Diagnostics settings, click **Enable** ☑.

Configure Suppression Settings for an SNMP Trap Incident

For information about each SNMP Traps tab:

NNMi enables you to suppress incidents based on Interface Group, Node Group, or default Suppression settings. NNMi applies your Suppression settings in the following order. Only the first match applies.

1. Interface Group (SNMP Trap Configuration Form: Interface Settings tab)
2. Node Group (SNMP Trap Configuration Form: Node Settings tab)

3. Enrich configuration settings without specifying an Interface Group or Node Group (SNMP Trap Configuration Form: Enrichment tab)

When configuring Interface Settings, Node Settings, or other Suppress Configuration, Enrich Configuration, or Dampening configuration settings for an incident, you can specify a Payload Filter. Payload Filters allows you to use the data that is included with any of the following items before they are stored as incidents in NNMi:

- Traps generated from an SNMP agent
- Management incidents that are generated by NNMi
- Events generated by NNM 6.x or 7.x management stations

Examples of the type of data that can be used as a Payload Filter include SNMP trap varbind names and values as well as Management Event CIA (Custom Incident Attribute) names and values. For example, you might want NNMi to dampen an incident based on a particular status change notification trap and participation within a specified Node Group or Interface Group. To do so, you would first specify participation in the Node Group or Interface Group for the trap you want to dampen. You would also specify a Payload Filter that includes the name of the trap varbind that stores the status information as well as the status change value string of interest.

See "Configure Incident Suppression Settings for an Interface Group (SNMP Trap Incident)" (on page 454) for information about how to suppress an incident for an Interface Group with or without a Payload Filter.

See "Configure Incident Suppression Settings for a Node Group (SNMP Trap Incident)" (on page 489) for more information about how to suppress an incident for a Node Group with or without a Payload Filter.

To configure suppression for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **SNMP Traps** tab.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete an incident configuration, click the Delete icon.
2. Select the **Suppression** tab.
3. Provide the required information (see table)
4. Click **Save and Close** to save your changes and return to the Incident Configuration form.

Suppression Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings. To temporarily disable the Suppress Configuration settings for the selected incident configuration, clear **Enable** ☐. To enable the Suppress Configuration settings for the selected incident configuration, click

Name	Description
	Enable☑.

Name	Description

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	
	When creating a Payload Filter, note the following:
	
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	
	The following example filters incidents on voltage state:
	
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`ciaValue = 5`
	
	NNMi evaluates the expression above as follows:
	
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**.

- **<** Finds all values less than the value specified. Click here for an example.

  Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**.

- **<=** Finds all values less than or equal to the value specified. Click here for an example.

  Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

  Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

  `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

  varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

Example:

`ciaValue not in`

matches any incident that contains a varbind with values other than **1** and **2**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

  Example:

  `ciaValue not in`

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

matches any incident that contains a varbind with values other than **1** and **2**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line. selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line. selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.
Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples. **Example 1** `AND`

Name	Description

Button	Description

```
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Name	Description

Button	Description

	Example 2
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Example 2
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Example 2
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`

Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Configure Enrichment Settings for an SNMP Trap Incident

For information about each SNMP Traps tab:

For information about each Enrichment tab:

NNMi enables you to fine tune and enhance incidents based on Interface Group, Node Group, or default Enrichment settings. NNMi applies your Enrichment settings in the following order. Only the first match applies:

1. Interface Group (SNMP Trap Configuration Form: Interface Settings tab)
2. Node Group (SNMP Trap Configuration Form: Node Settings tab)
3. Enrich configuration settings without specifying an Interface Group or Node Group (SNMP Trap Configuration Form: Enrichment tab)

The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To

Note: Any configuration you specify for Severity, Priority, or Message overrides those values provided in the SNMP Trap Configuration Form: Basics information.

You can also add a Custom Incident Attribute to the incoming incident.

When configuring Interface Settings, Node Settings, or other Suppress Configuration, Enrich Configuration, or Dampening configuration settings for an incident, you can specify a Payload Filter. Payload Filters allows you to use the data that is included with any of the following items before they are stored as incidents in NNMi:

- Traps generated from an SNMP agent
- Management incidents that are generated by NNMi
- Events generated by NNM 6.x or 7.x management stations

Examples of the type of data that can be used as a Payload Filter include SNMP trap varbind names and values as well as Management Event CIA (Custom Incident Attribute) names and values. For example, you might want NNMi to enrich an incident based on a particular status change notification trap and participation within a specified Node Group or Interface Group. To do so, you would first specify participation in the Node Group or Interface Group for the trap you want to enrich. You would also specify a Payload Filter that includes the name of the trap varbind that stores the status information as well as the status change value string of interest.

See "Configure Incident Enrichment Settings for an Interface Group (SNMP Trap Incident)" (on page 462) for information about how to enrich an incident for an Interface Group with or without a Payload Filter.

See "Configure Incident Enrichment Settings for a Node Group (SNMP Trap Incident)" (on page 501) for more information about how to enrich an incident for a Node Group with or without a Payload Filter.

To configure Enrichment settings for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **SNMP Trap Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **SNMP Traps** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Enrichment** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.
4. Provide the required information (see table)
5. Click **Save and Close** to save your changes and return to the Incident Configuration form.

Enrichment Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: • Accounting • Application Status • Configuration • Fault • Performance • Security • Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443) for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Possible values include: • Address • Aggregated Port • Board

Name	Description
	• Connection • Correlation • HSRP • Interface • Node • OSPF See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443) for more information.
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below: **Normal** - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there may or may not be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: 5 **None** 4 **Low** 3 **Medium** 2 **High** 1 **Top** **Note**: The icons are displayed only in table views.

Name	Description
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact See Incident Form: General Tab for more information.
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right. You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (SNMP Trap Incident)" (on page 446) "Include Custom Incident Attributes in Your Message Format (SNMP Trap Incident)" (on page 452)
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 2048 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Configure Dampening Settings for an SNMP Trap Incident

For information about each SNMP Traps tab:

NNMi enables you to delay (dampen) the following for an incident configuration:

- Appearance within Incident views in the NNMi Console
- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software* \ *NNM iSPI NET*)

You can configure Dampening settings based on Interface Group, Node Group, or default Dampening settings. NNMi applies your Dampening settings in the following order. Only the first match applies.

1. Interface Group (SNMP Trap Configuration Form: Interface Settings tab)
2. Node Group (SNMP Trap Configuration Form: Node Settings tab)
3. Dampening configuration settings without specifying an Interface Group or Node Group (SNMP Trap Configuration Form: Dampening tab)

When using Dampening configuration, note the following:

- For all Incident Configurations except Deduplication and Rate Incidents, if the dampened Incident is Closed before the Dampen Interval has passed, NNMi deletes the Incident. If the Incident is the Root Cause Incident, NNMi also deletes any Child Incidents

  Note: To view the number of incidents Closed and deleted while dampened, access the NNMi System Health Report Dampened Metrics (**Help** → **System Information** → **Health** tab).

- For all Incident Configurations except Deduplication and Rate Incidents, if the dampened Incident is Closed before the Dampen Interval has passed, NNMi deletes the Incident. If the Incident is the Root Cause Incident, NNMi also deletes any Child Incidents.
- NNMi always retains the Parent Deduplication or Rate Incident even If its Child Incidents are Closed within the Dampen Interval and subsequently deleted. See "Correlate Duplicate Incidents (Deduplication Configuration)" (on page 346) and "Track Incident Frequency (Rate: Time Period and Count)" (on page 347) for more information about Duplicate and Rate Correlation incidents.
- Any Deduplication and Incidents that have Child Incidents inherit the Dampening settings from their Correlated Children.
- If an incident is a Root Cause Incident and a Child Incident's Dampen Interval is less than the Parent Incident's Dampen Interval, NNMi holds any Child Incidents until the Dampen Interval for the Parent Incident has passed or until the Parent Incident is Closed and subsequently deleted.
- To make sure NNMi handles both Incidents in a Pairwise Configuration the same, configure the same Dampen Interval for each Incident in a Pairwise Incident Configuration.
- After the Dampen Interval has passed, NNMi changes the Lifecycle State to REGISTERED.

  See About the Incident Lifecycle for more information about Lifecycle State.

- You can use a Payload Filter to fine tune the incidents you want to dampen.

When configuring Interface Settings, Node Settings, or other Suppress Configuration, Enrich Configuration, or Dampening configuration settings for an incident, you can specify a Payload Filter. Payload Filters allows you to use the data that is included with any of the following items before they are stored as incidents in NNMi:

- Traps generated from an SNMP agent
- Management incidents that are generated by NNMi
- Events generated by NNM 6.x or 7.x management stations

Examples of the type of data that can be used as a Payload Filter include SNMP trap varbind names and values as well as Management Event CIA (Custom Incident Attribute) names and values. For example, you might want NNMi to dampen an incident based on a particular status change notification trap and participation within a specified Node Group or Interface Group. To do so, you would first specify participation in the Node Group or Interface Group for the trap you want to dampen. You would also specify a Payload Filter that includes the name of the trap varbind that stores the status information as well as the status change value string of interest.

See "Configure Incident Dampening Settings for an Interface Group (SNMP Trap Incident)" (on page 472) for information about how to configure Dampening settings for an Interface Group with or without a Payload Filter.

See "Configure Incident Dampening Settings for a Node Group (SNMP Trap Incident)" (on page 511) for more information about how to configure Dampening for a Node Group with or without a Payload Filter.

To configure Dampening settings for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **SNMP Traps** tab.
   d. Do one of the following:
      i. To create a configuration, click the New icon, and continue.
      ii. To edit configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete a configuration, click the Delete icon.
2. Select the **Dampening** tab.
3. Provide the required information (see table)
4. Click **Save and Close** to save your changes and return to the Incident Configuration form.

Dampening Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's Dampening settings. To temporarily disable the Dampening settings for the selected incident configuration, clear **Enable** ☐. To enable the Dampening settings for the selected incident configuration, click **Enable** ☑.
Hour	Specifies the number of hours to be used for the Dampen Interval.
Minutes	Specifies the number of minutes to be used for the Dampen Interval. Note the following: ■ The maximum Dampen Interval is 60 minutes. ■ Any Dampen Interval set for less than six minutes might not provide enough time for the Causal Engine to complete its analysis. As a result, Incidents associated with intermediate symptoms might appear as Root Cause Incidents.
Seconds	Specifies the number of seconds to be used for the Dampen Interval.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter

Name	Description

	editor.
	
	When creating a Payload Filter, note the following:
	
	■ Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	
	■ You must use a `ciaName` that already exists in the trap or event you are configuring.
	
	■ Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	
	■ View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	
	■ The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	
	The following example filters incidents on voltage state:
	
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`ciaValue = 5`
	
	NNMi evaluates the expression above as follows:
	
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	
	■ The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	
	■ The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Operator	Valid operators are described below. ■ **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **!=** Finds all values not equal to the value specified. Click here for an

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue

example.

Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**.

- **<** Finds all values less than the value specified. Click here for an example.

  Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**.

- **<=** Finds all values less than or equal to the value specified. Click here for an example.

  Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

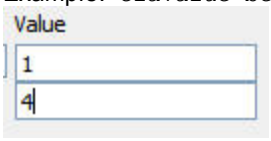
  Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue

`ciaValue in`

Value
4
5

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

  `ciaValue like .*Chicago.*` finds all traps or events that contain a

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue

varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
	`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: ■ The values you enter are case sensitive. ■ NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. ■ The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of

Name	Description

Button	Description
	the expression as it is created. the expression as it is created. the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.

Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples.
	Example 1
	`AND`
	`  ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6`
	`  OR`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:
	`AND`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6`
	`  OR`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3`
	Example 2
	`AND`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`

Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Name Desc

Configure Forward to Global Manager Settings for an SNMP Trap Incident (*NNMi Advanced*)

For information about each SNMP Traps tab:

(*NNMi Advanced - Global Network Management feature*) The NNMi Global Network Management feature enables multiple NNMi management servers to work together while managing different geographic areas of your network. See NNMi's Global Network Management Feature (NNMi Advanced) for more information. The Global Manager combines topology information from multiple Regional Managers, but maintains *a separate set of incidents about those nodes*.

Use the Global Manager Forwarding tab when you want to forward specific SNMP traps from your NNMi management server (a Regional Manager) to all Global Managers in your Global Network Management environment.

Caution: The Global Manager must have an incident configuration for that SNMP trap, otherwise the incoming trap is dropped. See "Export and Import Configuration Settings" (on page 1022) for ideas about sharing incident configurations among NNMi management servers.

When you configure Forward to Global Managers, you can specify an optional Payload Filter for NNMi to use when determining *which occurrences* should be forwarded to Global Managers. Payload Filters enable you to use the data that is included with an occurrence of an incident configuration before it is stored as an incident in the NNMi database.

Examples of the type of data that can be used as a Payload Filter include Custom Incident Attribute names (ciaName) and values (ciaValue). For example, you might want NNMi to forward an incident based on a particular status change notification trap. To do so, you would specify a Payload Filter that includes the name of the Custom Incident Attribute that stores the status information as well as the status change value string of interest.

Tip: see also "Configure Trap Forwarding Destinations" (on page 301).

To configure Forwarding to Global Managers:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **SNMP Traps** tab.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete an incident configuration, click the Delete icon.
2. Select the **Forward to Global Managers** tab.
3. Provide the required information (see table)
4. Click **Save and Close** to save your changes and return to the Incident Configuration form.

Forwarding Configuration Attributes

Name	Description
Enable	Use this attribute to enable or temporarily disable an incident's Forward to Global Managers settings. To temporarily disable the Global Manager Forwarding configuration settings for the selected incident configuration, clear **Enable** ☐. To enable the Global Manager Forwarding settings for the selected incident configuration, click **Enable** ☑.

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that NNMi forwards to other servers. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	
	When creating a Payload Filter, note the following:
	
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	
	The following example filters incidents on voltage state:
	
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`ciaValue = 5`
	
	NNMi evaluates the expression above as follows:
	
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	
	NNMi finds all incidents with a varbind value of `.1.3.6.1.4.1.9.9.13.1.2.1.7` and CIA value of **5**.
	
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Name	Description

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that with a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that with a varbind value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that with a varbind value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all values equal to and between the two values specified. Click here for an example.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

Example: `ciaValue between`

matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident that with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with no varbind values.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this*

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

location.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
	value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
	value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Name	Description

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.

Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples.
	Example 1
	`AND`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6`
	`  OR`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:
	`AND`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6`
	`  OR`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3`
	Example 2
	`AND`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`

Name	Description
	<table><tr><th>Button</th><th>Description</th></tr><tr><td>Delete</td><td>Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr></table>

Name Descr

Configure Deduplication for an SNMP Trap Incident

For information about each SNMP Traps tab:

The deduplication configuration determines what values NNMi should match to detect when an SNMP trap, management event, or remote NNM 6.x/7.x event is a duplicate.

Note the following:

- NNMi applies only one deduplication configuration per incident. If NNMi generates an incident using a specified deduplication configuration, NNMi continues to correlate duplicate incidents using the original configuration. To use a different deduplication configuration for an incident, first delete the current deduplication incident (created using the original deduplication configuration). NNMi generates the next deduplication incident according to the new deduplication configuration settings.
- By default, NNMi updates the **Duplicate Count** every 30 seconds. This interval cannot be changed.
- NNMi continues to update the duplicate count regardless of an incident's lifecycle state. For example, if an incident's **Lifecycle State** is set to **Closed**, the duplicate count continues to be incremented. See About the Incident Lifecycle for more information. This behavior helps you identify situations in which the incident is not yet fixed. Take note if the Duplicate Count is incremented after a lengthy time period has elapsed, which might indicate there is a new problem with the node, interface, or address.
- Each time you stop and restart ovjboss, any incidents that have not yet been correlated or persisted are lost. This means that after a restart of ovjboss, an incoming incident might not be correlated as expected. For example, after a restart of ovjboss, a duplicate incident might not be correlated under its original parent incident. Instead, a new parent incident might be generated. See "Stop or Start an NNMi Process" (on page 45) for more information about starting and stopping the ovjboss process.
- If a Duplicate Correlation Incident is dampened, note the following:
  - Duplicate Correlation Incidents inherit the Dampening settings from its Correlated Children.
  - NNMi always retains the Parent Duplicate Correlation incident, even if its Child Incidents are Closed and subsequently deleted.

    See "Dampening Incident Configurations" (on page 356) for more information about Dampening an Incident Configuration.

To specify or delete a deduplication configuration:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **SNMP Traps** tab.
   d. Do one of the following:
      i. To create a deduplication configuration, click the New icon, and continue.
      ii. To edit a deduplication configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete a deduplication configuration, click the Delete icon.
2. Select the **Deduplication** tab.

3. Provide the required information (see "Deduplication Attributes" table).
4. Click **Save and Close** to save your changes and return to the Incident Configuration form.

Deduplication Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's deduplication configuration. To temporarily disable the deduplication configuration setting, clear **Enable** ☐. To enable the deduplication configuration setting, click **Enable** ☑. **Note**: After a deduplication configuration is enabled, NNMi increments the **Duplicate Count** for an associated incident regardless of the **Lifecycle State** value. For example, if an incident's **Lifecycle State** is set to **Closed**, the duplicate count continues to be incremented. See About the Incident Lifecycle for more information.
Count	Specifies the number of duplicate incidents for the current configuration that NNMi stores at one time. For example, if the Count is 10, after NNMi receives 10 duplicate incidents, NNMi deletes the first (oldest) duplicate incident and keeps the eleventh. (NNMi stores ten maximum.)
Hour Interval	Used with the Minute and Second Intervals to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Hour Interval value is 1, and no Minute or Second Intervals are specified, and the duplicate incident is not generated within one hour, NNMi generates a new duplicate incident the next time it occurs.
Minute Interval	Used with the Hour and Second interval to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Minute Interval is 30 and no Hour or Second Intervals are specified, and the duplicate incident is not generated within 30 minutes, NNMi generates a new duplicate incident the next time it occurs.
Second Interval	Used with the Hour and Minute Intervals to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Second Interval is 120 and no Hour or Minute Intervals are specified, and the duplicate incident is not generated within 120 seconds, NNMi generates a new duplicate incident the next time it occurs.
Correlation Incident Config	Used to access the out-of-the box deduplicaton configuration provided by NNMi. Select the default value **Duplicate Correlation**. **Note**: You can choose to use this configuration as is or edit it. If you want to create a new deduplication configuration, you must create a new incident configuration. After you have created a new incident configuration, it appears in the **Quick Find** list of options. See "Lookup Fields" (on page 27) for more informationn about Quick Find.
Comparison Criteria	Specify the attribute values that must match before the incident is identified as a duplicate. The possible attributes consist of the following choices. • **Name** - The **Name** attribute value from the Incident form: General tab. • **CIA** - Represents any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " (on page 346): ▪ The **Value** attribute from the Incident form: Custom Attributes tab ▪ An SNMP varbind Object ID

Name	Description

- An SNMP varbind position number

  If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " (on page 346)

- **SourceNode** - The **Source Node** attribute value from the Basics attributes listed on the Incident form. The Source Node value is the IP Address or Name of the node for which the incident was generated.

  Note: The Source Node must be stored in the NNMi database.

- **Source Object** - The **Source Object** attribute value from the Basics attributes listed on the Incident form.

  Note: The Source Object must be stored in the NNMi database.

Note: Each attribute value in the option you select must match before the incident is identified as a duplicate. For example, if you select **Name**, only the Incident Name value must match. If you select **Name SourceNode SourceObject CIA**, the Incident Name, Source Node, Source Object, and all Custom Incident Attribute values that you configure as a Parameter Value must match before NNMi identifies the incident as a duplicate.

Selecting an option that includes CIA enables you to further refine the deduplication criteria. For example, you might want to configure deduplication for incidents with CIA values that specify the same State attribute value for a particular network object.

For a description of each Comparison Criteria option, click here.

Comparison Criteria	Description
Name	Value of the **Name** attribute from the Incident form: General tab must match.
Name CIA	Each of the following values must match: • **Name** attribute from the Incident form: General tab • **CIA** - Represents the Value associated with any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " (on page 346): ▪ Name of a Custom Incident Attribute (CIA) provided by NNMi. (See the Incident form: Custom Attributes tab ▪ An SNMP varbind Object ID ▪ An SNMP varbind position number If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " (on page 346)
Name SourceNode	**Note**: Select this option only if the Source Node is stored in the NNMi database. Each of the following values must match: • **Name** attribute from the Incident form: General tab • The **Source Node** attribute value from the Basics attributes listed on the Incident form
Name SourceNode CIA	**Note**: Select this option only if the Source Node is stored in the NNMi database.

Name	Description

Name	Description
Comparison Parameter List	*Optional*. If you selected a Comparison Criteria that includes CIA, you must populate one or more rows in this table. See "Deduplication Comparison Parameters Form " (on page 346).

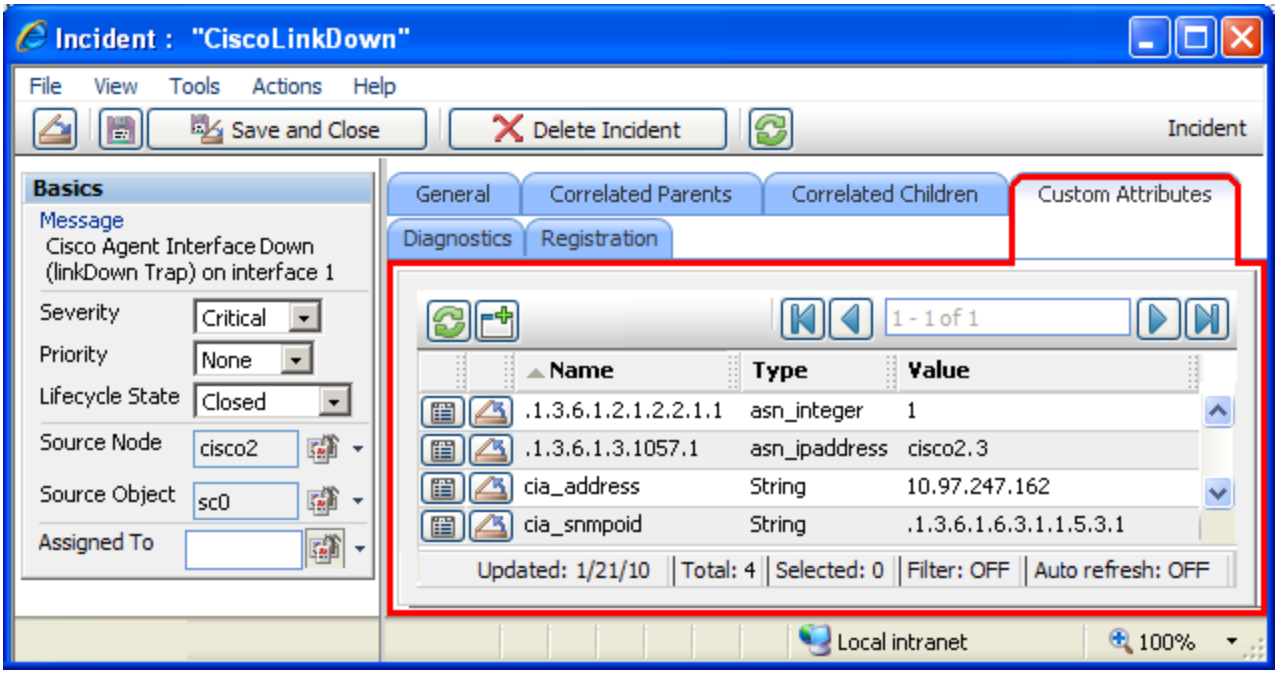
Deduplication Comparison Parameters Form (SNMP Trap Incident)

Comparison Parameter values enable accurate identification of duplicate incidents. Custom Incident Attributes (CIAs) are used as Comparison Parameter values. There are two categories of CIAs:

- SNMP trap varbind values (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312).

The group of available CIAs depends on which incident you are configuring for this Deduplication (for example, CiscoLinkDown). To see which CIAs are available, navigate to an Incident view, select an instance of that incident-type, click the Open icon and navigate to the Custom Attributes tab. The items listed in the table are the CIAs for that particular incident-type. For example, all CiscoLinkDown incidents would have the same group of CIAs shown in the illustration below.

Note: You can also use the CIA (varbind) position number.



To specify a CIA to use in the identification criteria for duplicate incidents:

1. Navigate to the **Deduplication Comparison Params** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Navigate to the **SNMP Traps** tab.
    d. Do one of the following:

- To create a new configuration, click the New icon.
- To edit a configuration, click the Open icon in the row representing the configuration you want to edit.

e. On the form that opens, navigate to the **Deduplication** tab.

f. Locate the **Comparison Parameter List** table.

g. Do one of the following to specify which CIA:

- To add a Custom Incident Attribute parameter specification, click the New icon.
- To edit an existing Custom Incident Attribute parameter specification, click the Open icon in the row representing the configuration you want to edit.

2. In the Parameter Value field, type (or copy and paste) the exact text string from the Incident form, Custom Attribute tab, **Name** attribute value:
   - NNMi-provided CIA value (see "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312)).
   - SNMP trap varbind identified by the Abstract Syntax Notation value (ASN.1).
3. Click **Save and Close** to save your changes and return to the previous configuration form.

Configure Rate (Time Period and Count) for an SNMP Trap Incident

For information about each SNMP Traps tab:

Use Rate configuration to track incident patterns *based on the number of incident reoccurrences within a specified time period*. After the count within the specified time period is reached, NNMi emits a Rate Correlation incident and continues to update the Correlation Notes with the number of occurrences within that rate.

As long as your defined criteria (Count and Hours, Minutes, Seconds) is sustained, the following information is updated in the Correlation Notes of the Rate Correlation incident:

- the actual number of occurrences of incidents for that sustained rate (Count)
- the sustained time interval (Hours, Minutes, Seconds)

For example, you can set a Rate configuration to track when a link is intermittently down at least three times in 30 minutes. NNMi shows the first occurrence of the rate incident in the incident view and uses Correlation Notes to update the number of incidents and time interval to reflect all the incremental incident occurrences and time periods. To continue the example, if the rate of three times in 30 minutes is sustained for 90 minutes, NNMi updates the Correlation Notes to specify that 9 incidents occurred in 90 minutes.

NNMi provides preconfigured Rate correlations. You can add new Rate correlations.

When you open the Incident form of the newest instance:

- On the General tab, two fields notify you that the Rate correlation is working:
  - **Correlation Nature**: Rate
  - **Count**: x
- On the **Correlated Children** tab, each incident is listed in the table.

- If a Rate Correlation Incident is dampened, note the following:
  - Rate Correlation Incidents inherit the Dampening configuration settings from its Correlated Children.
  - NNMi always retains the Parent Rate Correlation Incident, even if its Child Incidents are Closed and subsequently deleted.

    See "Dampening Incident Configurations" (on page 356) for more information about Dampening an Incident Configuration.

To establish a rate correlation within an incident configuration:

1. Navigate to the **Rate** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Open the **Incident Configuration** form.
   d. Navigate to the **SNMP Traps** tab.
   e. Do one of the following:
      - To create a new configuration, click the New icon.
      - To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
   f. On the form that opens, locate the **Rate** tab.
2. Provide the definition for this Rate configuration (see the "Rate Configuration Definition" table).
3. *Optional*. If your Comparison Criteria includes custom incident attributes (CIA) to identify one specific incident, use the Comparison Parameter List table to define each CIA.See "Rate Comparison Parameters Form" (on page 347).
4. Click **Save and Close** to save your changes and return to the Incident Configuration form.

Rate Configuration Definition

Attribute	Description
Enable	Use this attribute to temporarily disable an incident's rate settings. To temporarily disable the Dampen Configuration settings for the selected incident configuration, clear **Enable** ☐. To enable the Dampen Configuration settings for the selected incident configuration, click **Enable** ☑. If enabled, NNMi actively tracks any reoccurrences of the designated incident within the time period you specify, and generates a Rate incident.
Count	Specify the number of reoccurrences required before your Rate Configuration starts working.
Set the Time Period	Specify a time duration within which the reoccurrences are measured. Fill in one or more of the following attribute fields: **Hours**

Attribute	Description
	Minutes **Seconds**
Correlation Incident Config	Click the icon and select Quick Find. Select **Rate Correlation** from the list.
Comparison Criteria	Specify which group of attributes must match before the incident is identified as a duplicate. The possible groups of attributes consist of the following choices. **Name** value of the Incident (from the General tab on the Incident form). **Source Node** value (from the Basics group on the Incident form). Address or name of the node for which the incident was generated. **Source Object** value (from the Basics group on the Incident form). For example, the Source Object for a LinkDown incident is **interface**. **CIA** custom incident attribute values (select from the list displayed on the Custom Attributes tab on the Incident form). If you want to use CIA as part of your comparison criteria, see "Rate Comparison Parameters Form (SNMP Trap Incident)" (on page 569).
Comparison Parameter List	*Optional*. If you selected a Comparison Criteria that includes CIA, you must populate one or more rows in this table. See "Rate Comparison Parameters Form (SNMP Trap Incident)" (on page 569).

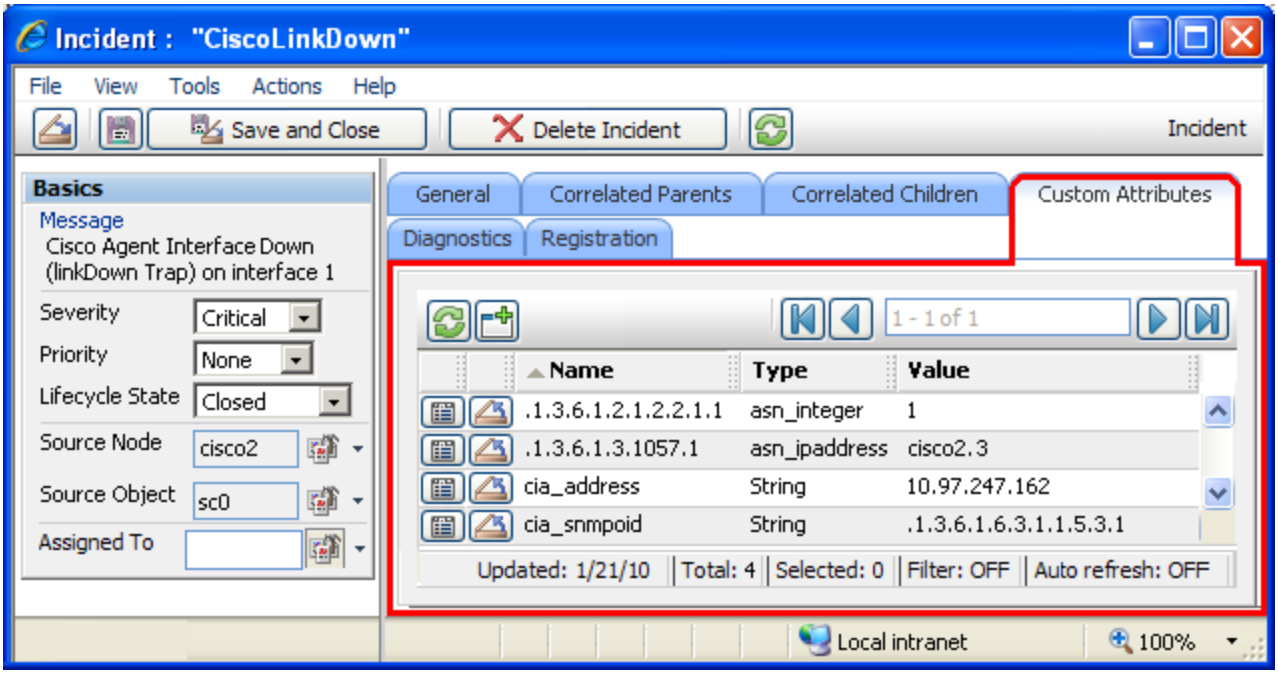
Rate Comparison Parameters Form (SNMP Trap Incident)

Custom Incident Attributes (CIAs) are used as parameter values. Parameter values enable accurate identification of duplicate incidents. There are two categories of CIAs:

- SNMP trap varbind values (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312).

The group of available CIAs depends on which incident you are configuring for this Rate (for example, CiscoLinkDown). To see which CIAs are available, navigate to an Incident view, select an instance of that incident-type, click the Open icon and navigate to the Custom Attributes tab. The items listed in the table are the CIAs for that particular incident-type. For example, all CiscoLinkDown incidents would have the same group of CIAs shown in the illustration below.

Note: You can also use the CIA (varbind) position number.



To specify a CIA to use in the identification criteria for duplicate incidents:

1. Navigate to the **Rate Comparison Params** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Navigate to the **SNMP Traps** tab.
    d. Do one of the following:
        - To create a new configuration, click the New icon.
        - To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
    e. On the form that opens, navigate to the **Rate** tab.
    f. Locate the **Comparison Parameter List** table.
    g. Do one of the following to specify which CIA:
        - To add a Custom Incident Attribute parameter specification, click the New icon.
        - To edit an existing Custom Incident Attribute parameter specification, click the Open icon in the row representing the configuration you want to edit.
2. In the Parameter Value field, type (or copy and paste) the exact text string from the Incident form, Custom Attribute tab, **Name** attribute value:
    - NNMi-provided CIA value (see "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312)).
    - SNMP trap varbind identified by the Abstract Syntax Notation value (ASN.1).
3. Click **Save and Close** to save your changes and return to the previous configuration form.

Configure Actions for an SNMP Trap Incident

For information about each SNMP Traps tab:

For information about each Actions tab:

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 (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

You can also configure incident actions based on either of the following:

- The Source Node's participation in a Node Group. See "Configure Incident Actions for a Node Group (SNMP Trap Incident)" (on page 521) for more information.
- The Source Object's participation in an Interface Group. See "Configure Incident Actions for an Interface Group (SNMP Trap Incident)" (on page 482) for more information.

You can configure actions for incidents generated from SNMP traps, NNM 6.x or 7.x events, and the NNMi management events. Any time an incident configuration changes, the action directory is rescanned and any executable or script files (for example, Jython) are reloaded to the NNMi database. See "Lifecycle Transition Action Form (SNMP Trap Incidents)" (on page 572) for more information about the actions directory.

Tip: Copy any required executable or script files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (SNMP Trap Incidents)" (on page 572) for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. To view the contents of the Actions log, use the **Tools** → **Incident Actions Log** menu option.

See "Verify that NNMi Services are Running" (on page 47) for more information about log files and where they are located.

NNMi sets the default values described in the following table. For information about how to change the default values, see the HP Network Node Manager i Software Deployment Reference, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

Action Server Properties

Property	Description	Default Value
numProcess	Number of actions that can be run at one time.	150
processTimeout	Amount of time in seconds that an action can run before the action server terminates the action.	300
numJythonThreads	Number of threads the action server uses to run Jython scripts	10
userName	User name under which the action server runs.	bin

To configure an automatic action for an incident:

1. Navigate to the **Actions** tab.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration.**
    c. Select the **SNMP Traps** tab.
    d. Do one of the following:
        - To create an incident configuration, click the New icon, and continue.
        - To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        - To delete an incident configuration, select a row, and click the Delete icon.
    e. Select the **Actions** tab.
2. From the **Lifecycle Actions** table toolbar, do one of the following:
    - To create an Action configuration, click the New icon, and continue.
    - To edit an Action configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
    - To delete an Action configuration, click the Delete icon.
3. In the "Lifecycle Transition Action Form (SNMP Trap Incidents)" (on page 572), provide the required information.
4. Click **Save and Close** to save your changes and return to the previous form.

    The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type.

Lifecycle Transition Action Form (SNMP Trap Incidents)

For information about each Action tab:

Use this form to enter the command you want to run when an incident of the type you are configuring is at a particular Lifecycle State. For example, when an incident is generated (**Registered**), you might want to automatically open a trouble ticket or email or page your network operator.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable on the Actions tab or using the **Actions** → **Enable Configuration** option.

To configure an action for an incidents:

1. Navigate to the **Lifecycle Transition Actions** form:
    a. From the workspace navigation pane, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **SNMP Traps** tab.
    d. Select the **Actions** tab.

e. From the **Lifecycle Transition Action** table toolbar, do one of the following:

- To create an Action configuration, click the New icon, and continue.
- To edit an Action configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
- To delete an Action configuration, click the Delete icon.

2. Make your configuration choices (see table).

   Note: NNMi reloads the configuration information anytime the incident configuration is changed.

3. Click **Save and Close** to save your changes and return to the previous form.

Create Action Attributes

Attribute	Description
Lifecycle State	Select a Lifecycle State from the drop-down menu.
Command Type	If you provided a Jython command, select **Jython** from the drop-down list. If you are using an executable or bat file, select **ScriptOrExecutable** from the drop-down list.

Command	Enter one of the following:
	• A Jython method with the required parameters
	• Executable command for the current operating system with the required parameters.
	When entering a **Command** value, note the following:
	• Verify that you do not have two Jython methods with the same name. Otherwise, NNMi is not able to tell which is the correct method to load.
	• You can use the same Jython method for more than one incident configuration.
	• Python files must reside in the following directory:
	Windows:
	`%NnmDataDir%\shared\nnm\actions`
	UNIX:
	`/var/opt/OV/shared/nnm/actions`
	• When using executable files, specify the absolute path name to the executable command or make sure the directory in which the executable file resides is in your PATH environment variable.
	• NNMi provides a set of parameters that can pass attribute values from an incident configuration. See "Valid Parameters for Configuring Incident Actions (Management Events)" (on page 859) for more information.

Configure a Payload Filter for an Action (SNMP Trap Incidents)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

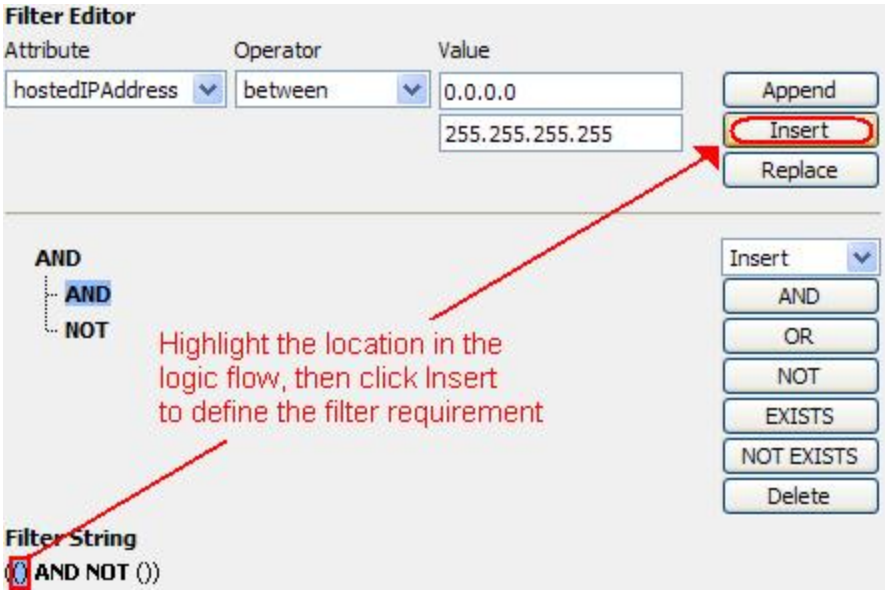
1. Navigate to the **SNMP Trap Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **SNMP Traps** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Actions** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.
4. Select the **Payload Filter** tab.
5. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".
    a. Plan out the logic needed for your Filter String.
    b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

        For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

        ```
        (( ) AND NOT ( ))
        ```

    c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



6. Click **Save and Close**.
7. Click **Save and Close** to save your changes and return to the previous form.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName!=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example.

Attribute	Description

Example: `ciaValue > 4` matches any incident that contains a varbind value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4** .

- **between** Finds all traps or events that include a varbind value equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4** .

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident that contains a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with varbind values.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with no varbind values.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

Attribute	Description

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Examples:

`ciaName like \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters.

`ciaValue like *Chicago*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**.

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2** .

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **`.1.3.6.1.4.1.9.9`**.

  `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Valid Parameters for Configuring Incident Actions (SNMP Trap Incident)

When configuring incident actions, you may want to use incident information as part of the action. NNMi provides the following parameter values. Use these parameters as variables in your Jython or executable files.

Tip: See the Using the Incident Form for more information about the parameter values.

Note: NNMi stores varbind values as custom incident attributes (CIAs).

See "Lifecycle Transition Action Form" (on page 415) for more information about configuring incident actions.

Valid Parameters Visible From an Incident's Form

Parameter Value	Description
$category, $cat	Value of the Category attribute in the Incident form.
$count, $cnt	Value representing the number of Custom Incident Attributes that appear in the Incident form.
$family, $fam	Value from the Family attribute in the Incident form.
$firstOccurrenceTime, $fot	Value from the First Occurrence Time attribute in the incident form.
$lastOccurrenceTime, $lot	Value from the Last Occurrence Time attribute in the incident form.
$lifecycleState, $lcs	Value from the Lifecycle State attribute in the Incident form.
$name	Value of the Name attribute from the incident configuration.
$nature, $nat	Value from the Nature attribute in the Incident form.
$origin, $ori	Value from the Origin attribute in the Incident form.
$originOccurrenceTime, $oot	Value from the Origin Occurrence Time attribute in the incident form.
$priority, $pri	Value from the Priority attribute in the Incident form.
$severity, $sev	Value of the Severity attribute of the Incident form.

Valid Parameters Visible from a Node Form

Parameter Value	Description
$managementAddress, $mga	Value from the Management Address attribute of the incident's source Node's form or SNMP Agent form.
$otherSideOfConnectionManagementAddress, $oma	If the incident's Source Node is part of a Layer 2 connection, this attribute is the value of the Management Address of a node on the other side of the Layer 2 connection.
$sourceNodeLongName, $sln	The fully-qualified DNS name as displayed in the Hostname attribute of the incident's source Node's form.
$sourceNodeName, $snn	Value from the Name attribute of the incident's source Node's form.
$sysContact, $sct	Value from the System Contact attribute of the incident's source Node form: General tab.
$sysLocation, $slc	Value from the System Location attribute of the incident's source Node form: General tab.

Valid Parameters Visible from an Interface Form

Parameter Value	Description
$ifAlias, $ifa	Value from the IfAlias attribute for the interface that is the incident's source object.
$ifConfigDupSetting, $icd	Configured Duplex Setting on the port associated with the interface that is the incident's source object.
$ifDesc, $idc	Value from the ifDesc attribute for the interface that is the incident's source object.
$ifIndex, $idx	Value from the ifIndex attribute for the interface that is the incident's source object.
$ifIpAddr, $iia	IP Address values associated with the interface that is the incident's source object. If multiple IPaddresses are associated with the interface, this parameter returns a comma-separated list.
$ifName, $ifn	Value from the ifName attribute for the interface that is the incident's source object.
$ifPhysAddr, $ipa	Value from the Physical Address attribute for the interface that is the incident's source object.
$ifSpeed, $isp	Value from the ifSpeed attribute for the interface that is the incident's souce object.
$ifType, $itp	Value from the ifType attribute for the interface that is the incident's souce object.

Valid Parameters Visible from a Layer 2 Connection Form

Parameter Value	Description
$otherSideOfConnectionConfigDupSetting, $ocd	If the incident's source Node is part of a Layer 2 connection, this parameter contains the Configured Duplex Setting on the port associated with the interface on the other side of the connection.
$otherSideOfConnectionIfAlias, $oia	If the incident's Source Node is part of a Layer 2 connection, this parameter is the value of the ifAlias of one of the interfaces on the other side of the Layer 2 connection.
$otherSideOfConnectionIfDesc, $odc	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifDescr attribute value for the interface on the other side of the Layer 2 connection.
$otherSideOfConnectionIfIndex, $odx	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifIndex attribute value for the interface on the other side of the connection.
$otherSideOfConnectionIfName, $ofn	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifName attribute value for the interface on the other side of the connection.

Valid Parameters Visible from a VLAN Form

Parameter Value	Description
$impVlanIds, $ivi	Value from the VLAN Id attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Port tab and open the form for the VLAN of interest. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.
$impVlanNames, $ivn	Value from the VLAN Name attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Ports tab of the Interface form. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.

Valid Parameters Not Visible From a Form

Parameter Value	Description
$id	Unique Object Identifier attribute value for the incident (unique across the entire NNMi Database).
$firstOccurrenceTimeMS, $fms	Value from the First Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$lastOccurrenceTimeMS, $lms	Value from the Last Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$oid	Value of the unique object identifier (oid) for the incident configuration that originated from either an SNMP trap, Management Event, or Remote NNM 6.x or 7.x event.
$otherSideOfConnection, $osc	If the incident's Source Node is part of a Layer 2 connection, this attribute is the following combination of values for the node and one of its interfaces on the other side of the Layer 2 connection: The fully-qualified DNS name of the node appended with the interface Name in the following format: <*fully-qualified DNS name*>[*interface_name*]
$originOccurrenceTimeMS, $oms	Value from the Origin Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$sourceNodeUuid, $snu	Universally Unique Object Identifier attribute value of the source node object for the incident (unique across all databases). This identifier distinguishes the source node object instance from all other node objects.
$sourceObjectClass, $soc	Value of the object class for the object you want to include. Use this parameter when you need to request more details of a class of objects through a web service. Examples of object classes include: `com.hp.ov.nms.model.core.Interface` and `com.hp.ov.nms.model.snmp.SnmpAgent`.
$sourceObjectName, $son	Value from the Name attribute of the source object. For example, an interface object is named according to the MIB ifName. Each ifName varies according to the vendor's conventions. Using the name `4/1` as an example, `4` represents `the board number and 1` represents the port

Parameter Value	Description
	number.
$sourceObjectUuid, $sou	Universally Unique Object Identifier attribute value of the source object for the incident (unique across all databases). This identifier distinguishes the source object instance from all other similar object instances..
$uuid	Universally Unique Object Identifier attribute value of the incident (unique across all databases). This identifier distinguishes the incident object instance from all other incident objects.

Valid Parameters Established in Custom Incident Attributes

Parameter Value	Description
$<position_ number>	Value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1` NNMi stores varbind values as Custom Incident Attributes. If you know the varbind position number, use this parameter.
$<CIA_ name>	Value of the name that is used for the custom incident attribute. For example, `$mycompany.mycia`. NNMi provides CIA values for configuring Management Events. See Custom Incident Attributes Provided by NNMi for more information about custom incident attributes.
$<CIA_ oid>	Value of the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this parameter when you are not certain of a custom incident attribute (varbind) position number.
$*	Used to indicate you want all of the custom incident attribute values originating as varbinds, to be passed to the action configuration. Each varbind is returned in the following format: `$<CIA_name>:<CIA_value>` in which the custom incident attribute name appears followed by the custom incident attribute value.

The function described in the following table replaces the specified numeric value with the associated text value stored in the CIA.

Note: The associated MIB must have been loaded using the nnmloadmib.ovpl command.

Functions to Generate Values Within Incident Messages

Function	Description
$text($<position_ number>)	The *<position_number>* argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1`. After the function runs, NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.

Function	Description
$text($<CIA_oid>)	The <CIA_oid> argument specifies the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this argument to the $text function when you are not certain of a custom incident attribute (varbind) position number. After the function runs, NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.

Configure Remote NNM 6.x/7.x Events

NNMi can display incidents from Remote NNM 6.x and 7.x management stations. In the NNMi incident browser, you can manage the lifecycle of incidents generated from previous versions of NNMi.

Tip: Gradually upgrade from NNM 6.x or 7.x to NNMi while using this feature.

To configure NNMi to handle incidents generated from remote NNM 6.x/7.x events, perform the following tasks:

- Configure the NNM 6.x/7.x Management Stations
- Configure what NNMi does with the NNM 6.x/7.x events

Configure Remote NNM 6.x and 7.x Management Stations

There are multiple benefits to configuring NNMi to recognize the NNM 6.x or 7.x management stations in your environment:

- Configure NNMi to receive and display incidents (events) from remote NNM 6.x or 7.x management stations.
- Enable displaying NNM 6.x or 7.x Dynamic Views from forwarded NNM 6.x or 7.x events (see Access NNM 6.x and 7.x Features for more information).
- Filter NNMi view by NNM 6.x or 7.x management station (show only those incidents received from a particular NNM 6.x or 7.x management station).

To display the details of an NNM 6.x or 7.x management station configuration:

1. From the workspace navigation panel, select the **Inventory** workspace.
2. Select the **Management Stations (6.x/7.x)** view.
3. Locate the row representing the management station and click the Open icon.
4. The Management Station form displays.
5. When finished, click the Close icon.

To configure an NNM 6.x or 7.x management station (if your role allows you to do this):

NNMi Advanced. Your NNMi management server must be configured as either an IPv4 or dual stack (IPv4/IPv6) machine to proceed. You must configure an IPv4 address for communication between an NNMi management server and the remote NNM 6.x or 7.x management station. (See the NNMi Advanced Release Notes for details.)

1. Navigate to the **Management Stations (6.x/7.x)** view.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select the **Management Stations (6.x/7.x)** view.
2. Do one of the following:
    - To create an NNM 6.x or 7.x management station configuration, click the New icon, and continue.
    - To edit an NNM 6.x or 7.x management station configuration, select a row, click the Open icon, and continue.
    - To delete an NNM 6.x or 7.x management station configuration, select a row, and click the Delete icon.
3. In the Management Station form, provide the required information:
    - IPv4 address of the remote NNM 6.x or 7.x management station
    - Port number used by the OpenView Application Server (ovas) on the remote NNM 6x or 7x management station
    - Port number used by the web server on the remote NNM 6x or 7x management station
4. Click **Save and Close** to return to the Management Stations (6.x/7.x) view.
5. If this is the first Management Station configuration, you must exit the NNMi console, and start the NNMi console. (You do not need to exit and start the NNMi console when configuring any subsequent NNM 6.x/7.x management stations.)
6. Next, configure which incidents to receive from your NNM 6.x or 7.x management station ("Configure Remote NNM 6.x/7.x Events" (on page 583)).

Remote NNM 6/x/7.x Event Configuration Form

Using NNMi, you can display incidents from Remote NNM 6.x and 7.x management stations . In the NNMi incident browser, you can manage the lifecycle of incidents generated from previous versions of NNMi.

Tip: Gradually upgrade from NNM 6.x or 7.x to NNMi while using this feature.

To configure a Remote NNM 6.x/7.x event:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Remote NNM 6.x/7.x Events** tab.
    d. Do one of the following:
        - To create a Remote NNM 6.x/7.x Event configuration, click the New icon.
        - To edit a Remote NNM 6.x/7.x Event configuration, click the Open icon in the row representing the configuration you want to edit.

    Note: In the Remote NNM 6.x/7.x Event Configuration form, verify that **Enable** is selected.

2. Make your configuration choices (see table).
3. Click **Save and Close** to save your changes and return to the **Incident Configuration** form.

Tasks for Remote NNM 6.x/7.x Event Configuration

Task	How
"Specify the Incident Configuration Name (Remote 6.x/7.x Event)" (on page 588)	Use the **Basics** group of the Remote NNM 6.x/7.x Event Configuration form. Specify a name that helps you to identify the configuration for subsequent use.
Specify whether you want to enable this configuration.	In the **Basics** group of the Remote NNM 6.x/7.x Event Configuration form, make sure **Enable** is checked for each configuration you want to use.
Display the NNMi Remote Incident as a Root Cause Incident	Use the **Basics** group of the Remote NNM 6.x/7.x Event Configuration form.
"Specify Category and Family Attribute Values for Organizing Your Incidents (Remote NNM 6.x/7x Events)" (on page 589)	Use the **Basics** group of the Remote NNM 6.x/7.x Event Configuration form. You can organize your incidents using Category and Family.
"Specify the Incident Severity (Remote NNM 6.x/7.x Events)" (on page 591)	Use the **Basics** group of the Remote NNM 6.x/7.x Event Configuration form. Possible Severity values include: **Normal, Warning, Minor, Major,** and **Critical**.
"Specify Your Incident Message Format (Remote NNM 6.x/7.x Events)" (on page 592)	Use the **Basics** group of the Remote NNM 6.x/7.x Event Configuration form. The message format determines the message to be displayed for the incident.
"Specify a Description for Your Incident Configuration (Remote NNM 6.x/7.x Events)" (on page 598)	Use the **Basics** group of the Remote NNM 6.x/7.x Event Configuration form. Provide a meaningful description.
Specify an Author for Your Remote NNM 6.x/7.x Event Configuration	Use the **Basics** pane of the **Remote NNM 6/x.7.x Event Configuration** form to indicate who created or last modified the event. See Author form for important information. **Caution**: If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. Click the Lookup icon and select Quick Find to access the list of existing Author values or click New to create one.

After you complete the Basic Configuration for the remote NNM 6.x or 7.x event, you can also choose to configure the information described in the following table.

Additional Configurations

Task	How
"Correlate Duplicate Incidents (Deduplication Configuration)" (on page 346)	Select the **Deduplication** tab to specify duplicate incidents that you want to be suppressed.

Task	How
"Track Incident Frequency (Rate: Time Period and Count)" (on page 347)	Select the **Rate** tab to specify a rate for duplicate incidents. After the rate limit is reached, NNMi generates an Incident to notify you of the problem.
"Configure an Action for an Incident" (on page 414)	Select the **Actions** tab to specify actions that should occur automatically when an incident changes its Lifecycle State.
"Configure Diagnostics for an Incident (NNM iSPI NET)" (on page 421)	Select the **Node Settings** tab to specify diagnostic actions that should occur automatically when an incident reaches a selected Lifecycle State for a node that belongs to a particular Node Group.

Configure Basic Settings for a Remote NNM 6.x/7.x Event Incident

The Basics settings for a Remote NNM 6.x/7.x event incident specifies general information for an incident configuration, including the name, severity, and message.

Note: In the **Basics** group of the **Remote NNM 6/x.7.x Event Configuration** form, verify that **Enable** ☑ is selected for each configuration you want to use.

For information about each Remote NNM 6/x.7.x Events tab:

To configure Basic settings for a Remote NNM 6.x/7.x Event incident:

Navigate to the **Remote NNM 6/x.7.x Event Configuration** form:

1. From the workspace navigation panel, select the **Configuration** workspace.
2. Select **Incident Configuration**.
3. Select the **Remote NNM 6/x.7.x Events** tab.
4. Do one of the following:
   a. To create an incident configuration, click the New icon, and continue.
   b. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   c. To delete an incident configuration, click the Delete icon.
5. Configure the required Basic settings (see table).
6. Click **Save and Close** to save your changes and return to the Incident Configuration form.

Basics Attributes for SNMP Trap Configuration

Task	How
"Specify the Incident Configuration Name (Remote 6.x/7.x Event)" (on page 588)	Use the **Basics** pane of the **Remote NNM 6/x.7.x Event Configuration** form. Specify a name that helps you to identify the configuration for subsequent use.
Specify whether you want to enable this configuration.	In the **Basics** group of the **Remote NNM 6/x.7.x Event Configuration** form, verify that **Enable** ☑ is selected for each configuration you want to use.
"Display an SNMP Trap or an NNM 6.x/7.x Event as a Root Cause Incident" (on page 443)	Use the **Basics** pane of the **Remote NNM 6/x.7.x Event Configuration** form.

Task	How
"Specify Category and Family Attribute Values for Organizing Your Incidents (Remote NNM 6.x/7x Events)" (on page 589)	Use the **Basics** pane of the **Remote NNM 6/x.7.x Event Configuration** form. You can organize your incidents using Category and Family.
"Specify the Incident Severity (Remote NNM 6.x/7.x Events)" (on page 591)	Use the **Basics** pane of the **Remote NNM 6/x.7.x Event Configuration** form. Possible Severity values include: **Normal, Warning, Minor, Major,** and **Critical**.
"Specify Your Incident Message Format (Remote NNM 6.x/7.x Events)" (on page 592)	Use the **Basics** pane of the **Remote NNM 6/x.7.x Event Configuration** form. The message format determines the message to be displayed for the incident.
"Specify a Description for Your Incident Configuration (Remote NNM 6.x/7.x Events)" (on page 598)	Use the **Basics** pane of the **Remote NNM 6/x.7.x Event Configuration** form. Provide a meaningful description.
Specify an Author for Your Incident Configuration (Remote NNM 6.x/7.x Events)	Use the **Basics** pane of the **Remote NNM 6/x.7.x Event Configuration** form to indicate who created or last modified the event. See Author form for important information. **Caution**: If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. Click the Lookup icon and select Quick Find to access the list of existing Author values or click New to create one.

After you complete the Basic Configuration for the remote NNM 6/x.7.x event, you can also choose to configure the information described in the following table.

Additional Incident Configurations

Task	How
"Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident" (on page 599)	Select the **Interface Settings** tab to specify an Interface Group to which you want your incident configuration to apply.
"Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" (on page 634)	Select the **Node Settings** tab to specify a Node Group to which you want your incident configuration to apply.
"Configure Suppression Settings for a Remote NNM 6.x/7.x Event Incident" (on page 675)	Select the **Suppression** tab to specify the criteria for discarding incidents that match the selected incident configuration.
"Configure Enrichment Settings for a Remote NNM 6.x/7.x Event Incident" (on page 684)	Select the **Enrichment** tab to specify enhancements for the selected incident configuration.
"Configure Dampening Settings for a Remote NNM 6.x/7.x Event Incident" (on page 687)	Select the **Dampen** tab to specify the time interval that must be met before the incident appears in an Incident view.

Task	How
"Configure Deduplication for a Remote NNM 6.x/7.x Event Incident" (on page 707)	Select the **Deduplication** tab to specify duplicate incidents that you want to be suppressed.
"Configure Rate (Time Period and Count) for a Remote NNM 6.x/7.x Event Incident" (on page 712)	Select the **Rate** tab to specify a rate for duplicate incidents. After the rate limit is reached, NNMi generates an Incident to notify you of the problem.
"Configure Actions for a Remote NNM 6.x/7.x Event Incident" (on page 716)	Select the **Actions** tab to specify actions that should occur automatically when an incident changes its Lifecycle State.

Specify the Incident Configuration Name (Remote 6.x/7.x Event)

When providing the Name for an incident configuration, use the following guidelines:

Name

The name is used to identify the incident configuration and must be unique. Use a name that will help you to remember the purpose or kind of management event, remote NNM 6.x or 7.x event or SNMP trap, for which you are configuring an incident. Name is also used to identify your Pairwise configurations.

Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. No spaces are allowed.

Display an SNMP Trap or an NNM 6.x/7.x Event as a Root Cause Incident

SNMP trap and NNM 6.x/7.x events normally appear as symptoms rather than as root cause incidents. However, there might be times when you want an SNMP or NNM 6.x/7.x event to appear as a root cause incident. For example, you might want an HSRP state change (cHsrpStateChange, 1.3.6.1.4.1.9.9.106.2.0.1) trap to be listed as a root cause. This trap might occur when the hot standby has gone down indicating the system is at risk if there is a failover.

Note: To reduce "noise" associated with duplicate incidents, NNMi changes the incident Correlation Nature to **Symptom** for any user-defined Root Cause incidents that exceed the rate or deduplication threshold. Any Stream Correlation incident associated with the user-defined Root Cause incident is change to a Root Cause incident. See Stream Correlation Incidents View for more information about Stream Correlations.

To display an SNMP trap or NNM 6.x/7.x Event as a root cause incident:

Select **Root Cause** ☑ in the **SNMP Trap** or **Remote NNM 6.x/7.x Event Configuration** form.

The next time the incident occurs, the Correlation Nature attribute value changes to **Root Cause**.

To no longer display an SNMP trap or NNM 6.x/7.x Event s a root cause incident:

Clear **Root Cause** ☐ in the **SNMP Trap** or **Remote NNM 6.x/7.x Event Configuration** form.

The next time the incident occurs, the Correlation Nature attribute value changes to **Root Cause**.

Specify Category and Family Attribute Values for Organizing Your Incidents (Remote NNM 6.x/7x Events)

When configuring incidents, NNMi provides the Category and Family attributes to help you organize your incidents.

Preconfigured Categories
The Category attribute helps you organize your incidents. Select the category that you want to be associated with this type of incident when it appears in an incident view. Each of the possible Category values is described in the following table.

Incident Categories Provided by NNMi

Category	Description
Accounting	Used to indicate problems related to usage statistics and allocation of costs associated with the billing of time and services provided by devices. This category is not used by NNMi with default configurations, but it is available for incidents you define.
Application Status	Indicates there is a problem with the health of the NNMi software. Examples of these kinds of events include license expiration (see "Extend a Licensed Capacity" (on page 1020)) or that a certain NNMi process or service lost connection to the Process Status Manager (see "Stop or Start an NNMi Process" (on page 45) and "Stop or Start NNMi Services" (on page 49)).
Configuration	Indicates there is a problem with the configuration of a managed device. For example, there is a physical address mismatch.
Fault	Indicates a problem with the network, for example Node Down.
Performance	Indicates a threshold has been exceeded. For example, a utility has exceeded 90 percent.
Security	Indicates there is a problem related to authentication. For example, an SNMP authentication failure.
Status	Indicates some kind of status message. Examples of these kinds of incidents include "SNMP Link Up" or an "HSRP Group status Normal" message.

Note: You can add your own Category entries to NNMi. See "Create an Incident Category (Remote NNM 6.x/7.x Event)" (on page 589) for more information.

Note: You can add your own Family entries to NNMi. See "Create an Incident Family (Remote NNM 6/x.7.x Event)" (on page 590) for more information.

Create an Incident Category (Remote NNM 6.x/7.x Event)

The Category attribute helps you organize your incidents. Create any Category that makes sense to you and your team. For a list of the Category codes provided by NNMi, "Specify Category and Family Attribute Values for Organizing Your Incidents (Remote NNM 6.x/7x Events)" (on page 589).

To create a new incident Category:

1. Navigate to the **Incident Category** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.

c. Select the **Remote NNM 6.x/7.x Events** tab.

d. Do one of the following:

- To create an incident configuration, click the New icon, and continue.
- To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

e. In the configuration form, locate the **Category** attribute.

f. Click the Lookup icon, and select New.

2. Provide the required information (see table).
3. Click **Save and Close** to save your changes and return to the previous form.

Category Code Attributes

Name	Description
Label	Incident category name. For example, Hardware Faults, or Cisco Error. Maximum size is 255 characters. Alpha-numeric, spaces, and underline characters are allowed.

Unique Key	**Caution**: After you click **Save and Close**, this value cannot be changed.
	Used as a unique identifier when exporting and importing configuration definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the label value as part of the unique key as shown in the following examples:
	`com.<your_company_name>.nnm.trapConf.category.<category_label>`
	`com.<your_company_name>.nnm.eventConf.category.<category_label>`
	`com.<your_company_name>.nnm.inciConf.category.<category_label>`
	The maximum length is 80 characters. Alpha-numeric characters and periods are allowed. Spaces are not allowed.

Create an Incident Family (Remote NNM 6/x.7.x Event)

The Family attribute helps you organize your incidents. Create any Family that makes sense to you and your team. For a list of the Family codes provided by NNMi, "Specify Category and Family Attribute Values for Organizing Your Incidents (Remote NNM 6.x/7x Events)" (on page 589).

To create a new incident Family:

1. Navigate to the **Incident Family** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **Remote NNM 6.x/7.x Events** tab.

a. Do one of the following:

- To create an incident configuration, click the New icon, and continue.
- To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

b. In the configuration form, locate the **Family** attribute.

c. Click the Lookup icon, and select New.

2. Provide the required information (see table).
3. Click **Save and Close** to save your changes and return to the previous form.

Family Attributes

Name	Description
Label	Family name. For example, Hardware Faults, or Cisco Error. Maximum size is 255 characters. Any character type is valid.

Unique Key	**Caution**: After you click **Save and Close**, this value cannot be changed.
	Used as a unique identifier when exporting and importing configuration definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the label value as part of the unique key as shown in the following examples:
	`com.<your_company_name>.nnm.trapConf.family.<family_label>`
	`com.<your_company_name>.nnm.eventConf.family.<family_label>`
	`com.<your_company_name>.nnm.inciConf.family.<family_label>`
	The maximum length is 80 alpha-numeric characters, periods allowed, no spaces allowed.

Specify the Incident Severity (Remote NNM 6.x/7.x Events)

The incident severity represents the seriousness calculated for the incident. Use the severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described in the following table.

Incident Severity Values

Attribute	Description
Normal	Indicates there are no known problems related to the associated object. This severity is meant to be informational. Generally, no action is needed for these incidents.
Warning	Indicates there may or may not be a problem related to the associated object.
Minor	Indicates NNMi has detected problems related to the associated object that require further investigation.
Major	Indicates NNMi has detected problems related to the associated object to be resolved before they become critical.
Critical	Indicates NNMi has detected problems related to the associated object that require immediate attention.

See "Monitor Incidents for Problems" for more information about these severity values.

Specify Your Incident Message Format (Remote NNM 6.x/7.x Events)

When configuring an incident, specify the information you want NNMi to include in the incident's Message attribute value. You can use any combination of valid parameter strings and Custom Incident attributes to configure the Message.

Note: The incident Message limit is 1024 characters. If the returned values exceed this limit, NNMi truncates the value starting from the end of the returned text string.

"Valid Parameters for Configuring Incident Messages (Remote NNM 6.x/7.x Events)" (on page 592)

"Include Custom Incident Attributes in Your Message Format (Remote NNM 6.x/7.x Events)" (on page 597)

Valid Parameters for Configuring Incident Messages (Remote NNM 6.x/7.x Events)

When configuring incident messages, you may want to use incident information as part of the message. NNMi provides the following parameter values. Use these parameters as variables when formatting an incident message.

Tip: See the Using the Incident Form for more information about the parameter values.

Note: NNMi stores varbind values as custom incident attributes (CIAs).

See "Specify Your Incident Message Format (Remote NNM 6.x/7.x Events)" (on page 592) for more information about configuring messages.

Parameter strings are available for the following:

- Incident form attributes (Click here for a list of choices.)

Parameter Strings for Attributes from an Incident Form

Parameter String	Description
$category, $cat	Value of the Category attribute in the Incident form.
$count, $cnt	Value representing the number of Custom Incident Attributes that appear in the Incident form.
$family, $fam	Value from the Family attribute in the Incident form.
$firstOccurrenceTime, $fot	Value from the First Occurrence Time attribute in the incident form.
$lastOccurrenceTime, $lot	Value from the Last Occurrence Time attribute in the incident form.
$lifecycleState, $lcs	Value from the Lifecycle State attribute in the Incident form.
$name	Value of the Name attribute from the incident configuration.
$nature, $nat	Value from the Nature attribute in the Incident form.
$origin, $ori	Value from the Origin attribute in the Incident form.
$originOccurrenceTime, $oot	Value from the Origin Occurrence Time attribute in the incident form.
$priority, $pri	Value from the Priority attribute in the Incident form.
$sev, $severity	Value of the Severity attribute of the Incident form.

- Node form attributes (Click here for a list of choices.)

Parameter Strings for Attributes from a Node Form

Parameter String	Description
$managementAddress, $mga	Value from the Management Address attribute of the incident's source Node's form or SNMP Agent form.
$otherSideOfConnectionManagementAddress, $oma	If the incident's Source Node is part of a Layer 2 connection, this attribute is the value of the Management Address of a node on the other side of the Layer 2 connection.
$sourceNodeLongName, $sln	The fully-qualified DNS name as displayed in the Hostname attribute of the incident's source Node's form.
$sourceNodeName, $snn	Value from the Name attribute of the incident's source Node's form.
$sysContact, $sct	Value from the System Contact attribute of the incident's source Node form: General tab.
$sysLocation, $slc	Value from the System Location attribute of the incident's source Node form: General tab.

- Interface form attributes (Click here for a list of choices.)

Parameter Strings for Attributes from an Interface Form

Parameter String	Description
$ifAlias, $ifa	Value from the IfAlias attribute for the interface that is the incident's source object.
$ifConfigDupSetting, $icd	Configured Duplex Setting on the port associated with the interface that is the incident's source object.
$ifDesc, $idc	Value from the ifDesc attribute for the interface that is the incident's source object.
$ifIndex, $idx	Value from the ifIndex attribute for the interface that is the incident's source object.
$ifIpAddr, $iia	IP Address values associated with the interface that is the incident's source object. If multiple IPaddresses are associated with the interface, this parameter returns a comma-separated list.
$ifName, $ifn	Value from the ifName attribute for the interface that is the incident's source object.
$ifPhysAddr, $ipa	Value from the Physical Address attribute for the interface that is the incident's source object.
$ifSpeed, $isp	Value from the ifSpeed attribute for the interface that is the incident's souce object.
$ifType, $itp	Value from the ifType attribute for the interface that is the incident's souce object.

- Layer 2 Connection form attributes (Click here for a list of choices.)

Parameter Strings for Attributes from a Layer 2 Connection Form

Parameter String	Description
$otherSideOfConnectionConfigDupSetting, $ocd	If the incident's source Node is part of a Layer 2 connection, this parameter contains the Configured Duplex Setting on the port associated with the interface on the other side of the connection.
$otherSideOfConnectionIfAlias, $oia	If the incident's Source Node is part of a Layer 2 connection, this parameter is the value of the ifAlias of one of the interfaces on the other side of the Layer 2 connection.
$otherSideOfConnectionIfDesc, $odc	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifDescr attribute value for the interface on the other side of the Layer 2 connection.
$otherSideOfConnectionIfIndex, $odx	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifIndex attribute value for the interface on the other side of the connection.
$otherSideOfConnectionIfName, $ofn	If the incident's Source Node is part of a Layer 2

Parameter String	Description
	connection, this parameter contains the ifName attribute value for the interface on the other side of the connection.

- VLAN form attributes (Click here for a list of choices.)
  Parameter Strings for Attributes from a VLAN Form

Parameter String	Description
$impVlanIds, $ivi	Value from the VLAN Id attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Port tab and open the form for the VLAN of interest. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.
$impVlanNames, $ivn	Value from the VLAN Name attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Ports tab of the Interface form. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.

- Additional information that is not visible in any form (Click here for a list of choices.)
  Parameter Strings for Attributes not visible in any Form

Parameter String	Description
$id	Unique Object Identifier attribute value for the incident (unique across the entire NNMi Database).
$firstOccurrenceTimeMS, $fms	Value from the First Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$lastOccurrenceTimeMS, $lms	Value from the Last Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$oid	Value of the unique object identifier (oid) for the incident configuration that originated from either an SNMP trap, Management Event, or Remote NNM 6.x or 7.x event.
$otherSideOfConnection, $osc	If the incident's Source Node is part of a Layer 2 connection, this attribute is the following combination of values for the node and one of its interfaces on the other side of the Layer 2 connection: The fully-qualified DNS name of the node appended with the interface Name in the following format: <*fully-qualified DNS name*>[*interface_name*]
$originOccurrenceTimeMS, $oms	Value from the Origin Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$sourceNodeUuid, $snu	Universally Unique Object Identifier attribute value of the source node object for the incident (unique across all databases). This identifier distinguishes the source node object instance from all other node objects.

Parameter String	Description
$sourceObjectClass, $soc	Value of the object class for the object you want to include. Use this parameter when you need to request more details of a class of objects through a web service. Examples of object classes include: `com.hp.ov.nms.model.core.Interface` and `com.hp.ov.nms.model.snmp.SnmpAgent`.
$sourceObjectName, $son	Value from the Name attribute of the source object. For example, an interface object is named according to the MIB ifName. Each ifName varies according to the vendor's conventions. Using the name `4/1` as an example, `4` represents `the board number and 1` represents the port number.
$sourceObjectUuid, $sou	Universally Unique Object Identifier attribute value of the source object for the incident (unique across all databases). This identifier distinguishes the source object instance from all other similar object instances..
$uuid	Universally Unique Object Identifier attribute value of the incident (unique across all databases). This identifier distinguishes the incident object instance from all other incident objects.

- Information established in Custom Incident Attributes (Click here for a list of choices.)

Parameter Strings for Attributes Established in Custom Incident Attributes

Parameter String	Description
$<position_number>	Value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1` NNMi stores varbind values as Custom Incident Attributes. If you know the varbind position number, use this parameter.
$<CIA_name>	Value of the name that is used for the custom incident attribute. For example, `$mycompany.mycia`. NNMi provides CIA values for configuring Management Events. See Custom Incident Attributes Provided by NNMi for more information about custom incident attributes.
$<CIA_oid>	Value of the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this parameter when you are not certain of a custom incident attribute (varbind) position number.
$*	Used to indicate you want all of the custom incident attribute values originating as varbinds, to be passed to the action configuration. Each varbind is returned in the following format: `$<`*`CIA_name`*`>:<`*`CIA_value`*`>` in which the custom incident attribute name appears followed by the custom incident attribute value.

- Functions to generate values (Click here for a list of choices.)

  The function described in the following table replaces the specified numeric value with the associated text value stored in the CIA.

  Note: The associated MIB must have been loaded using the nnmloadmib.ovpl command.

Functions to Generate Values Within the Incident Message

Function	Description
$oidtext($<position_number>)	A <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, `$oidtext($2)`. **Note**: The position number you enter must represent a CIA that contains an Object Identifier (OID) value. NNMi returns the textual value of the OID for the CIA specified. Note the following: ■ If the MIB is not loaded, NNMi returns the numeric OID value. ■ If the OID has a MIB instance, the number representing the MIB instance is appended to the textual OID value.
$oidtext($<CIA_oid>)	The <*CIA_oid*> argument specifies the Object Identifier (OID) for any custom incident attribute that originated as a varbind. For example, `$oidtext($.1.3.6.1.6.3.1.1.5.1.)` Use this argument to the $oidtext() function when you are not certain of a custom incident attribute (varbind) position number. NNMi replaces the numeric value with the textual value of the OID you specify. Note the following: ■ If the MIB is not loaded, NNMi returns the numeric OID value. ■ If the OID has a MIB instance, the number representing the MIB instance is appended to the textual OID value.
$text($<position_number>)	The <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1`. NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.
$text($<CIA_oid>)	The <CIA_oid> argument specifies the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this argument to the $text function when you are not certain of a custom incident attribute (varbind) position number. NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.

Include Custom Incident Attributes in Your Message Format (Remote NNM 6.x/7.x Events)

NNMi includes two categories of CIAs:

- SNMP trap varbinds identified by the Abstract Syntax Notation value (ASN.1). NNMi turns varbinds into CIAs and maintains each varbind's position number. See "Load SNMP Trap Incident Configurations"

(on page 429).

- Custom attributes provided by NNMi. See "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312).

You can use CIAs in your message format to extend the amount of information presented. To determine which CIAs are available for any particular incident type, open an Incident view, locate the incident and open the Incident form. Navigate to the **Custom Attributes** tab. A complete list of available CIAs (for that incident type) appears in the table.

To include a CIA in your message format, type the dollar-sign character (`$`) plus any of the following:

- Varbind position number or asterisk (*) to include all varbind values
- Name of the CIA
- Object identifier (oid) of the CIA (useful when the varbind position number is not consistent among vendors)

Note: A single incident cannot include two CIAs with the same name. However, two incidents can contain CIAs having the same names and values.

The following table presents some example formats with the subsequent output.

Example Incident Message Formats

Example Message Format	Output in Incident View
Possible trouble with $3	`Possible trouble with` <varbind 3>
Possible trouble with $11	`Possible trouble with` <varbind 11>
Possible trouble with $77 (where the varbind position 77 does not exist)	`Possible trouble with <Invalid or unknown cia> 77`
Possible trouble with $*	`Possible trouble with` <cia1_name: cia_value>, <cia2_name; cia_value>,< cia*n*_name: cia_value>
Possible trouble with $3x	`Possible trouble with` <varbind 3>`x`
Possible trouble with $1.2.3.4.5	`Possible trouble with` <value of the CIA with oid of 1.2.3.4.5>
Possible trouble with $mycia.mycompany	`Possible trouble with` <value of the CIA with name of mycia.mycompany>

Tip: NNMi provides an error message when a CIA cannot be found. For example, if you enter an unavailable varbind position, name, or object identifier (oid), NNMi returns an "Invalid or unknown cia" error message.

Specify a Description for Your Incident Configuration (Remote NNM 6.x/7.x Events)

NNMi provides the Description attribute to help you further identify the current incident configuration.

Description

Use the description field to provide additional information that you would like to store about the current incident configuration. This description applies only to the configuration entry.

Type a maximum of 2048 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident

Note: Interface Settings override any other Suppression, Enrichment, Dampen, or Actions settings for this incident, including those configured on the Node Settings tab.

NNMi enables you to apply an incident configuration to a Source Object based on the Source Object's participation in an Interface Group.

Tip: See "Create Interface Groups" (on page 197) for more information about Interface Groups.

For information about each Interface Settings tab:

For information about each Remote NNM 6.x/7.x Events tab:

To apply an incident configuration to a Source Object based on the Source Object's Interface Group:

1. Navigate to the **Remote NNM 6/x.7.x Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Remote NNM 6.x/7.x Events** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
4. Configure the desired Interface Settings (see table).
5. Configure any Suppression, Dampen, or Enrichment settings for this Interface Group.
6. Click **Save and Close** to save your changes and return to the Incident Configuration form.

Interface Group Attributes

Name	Description
Interface Group	Click the Lookup icon and select Quick Find to select the Interface Group you want to use. See "Use the Quick Find Window" (on page 28) for more information about using Quick Find.
Ordering	Determines the priority order for those interfaces that appear in multiple Interface Groups. The lower the number, the higher the priority. For example, **1** is the highest priority. If an interface is in multiple Interface Groups and more than one of those Interface Groups have been specified in an incident configuration, only the incident configuration with the highest priority will be applied to the interface.

Name	Description
Enable	Use this attribute to temporarily disable an incident's configuration settings. To temporarily disable the Interface Group settings for the selected incident configuration, clear **Enable** ☐. To enable the Interface Group settings for the selected incident configuration, click **Enable** ☑.

Related Topics

"Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" (on page 634)

Configure Incident Suppression Settings for an Interface Group (Remote NNM 6.x/7.x Events)

Note: Interface Settings override any other Suppression settings for this incident, including those from the Node Settings tab.

NNMi enables you to suppress a specified incident configuration based on the Source Object's participation in an Interface Group.

Note: You can also suppress the incident configuration based on the Source Node's participation in a Node Group. See "Configure Incident Suppression Settings for a Node Group (Remote NNM 6.x/7.x Events)" (on page 636) for more information.

Tip: See "Create Interface Groups" (on page 197) for more information about Interface Groups.

For information about each Interface Settings tab:

To suppress an incident configuration based on an Interface Group:

1. Navigate to the **Remote NNM 6/x.7.x Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Remote NNM 6.x/7.x Events** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.
4. Make sure the basic Interface Setting behavior is configured. See "Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident" (on page 599) for more information.

5. Select the **Suppression** tab.
6. Configure the desired Suppression behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

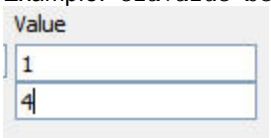
Interface Settings Suppression Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings for the specified Interface Group. To temporarily disable the Suppression settings for the selected incident configuration, clear **Enable** ☐. To enable the Suppression settings for the selected incident configuration, click **Enable** ☑.

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	
	When creating a Payload Filter, note the following:
	
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	
	The following example filters incidents on voltage state:
	
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`ciaValue = 5`
	
	NNMi evaluates the expression above as follows:
	
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Name	Description

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all values equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

  `ciaValue like .*Chicago.*` finds all traps or events that contain a

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like .*Chicago.*` finds all traps or events that do not

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	contain a varbind value that includes the string **Chicago**. contain a varbind value that includes the string **Chicago**. contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.
Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples. **Example 1**

Name	Description

Button	Description

```
AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
```

Name	Description

Button	Description
	ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Name Desc

Configure Incident Enrichment Settings for an Interface Group (Remote NNM 6.x/7.x Events)

Note: Interface Settings override any other Enrichment settings for this incident, including those from the Node Settings tab.

NNMi enables you to fine tune and enhance a specified incident configuration based on the Source Object's participation in an Interface Group. The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To

Note: You can also enhance the incident configuration based on the Source Node's participation in a Node Group. See "Configure Incident Enrichment Settings for a Node Group (Remote NNM 6.x/7.x Events)" (on page 647) for more information.

Tip: See Create Interface Groups for more information about Interface Groups.

For information about each Interface Settings tab:

For information about each Enrichment tab:

To enrich an incident configuration based on an Interface Group:

1. Navigate to the **Remote NNM 6/x.7.x Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Remote NNM 6.x/7.x Events** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.
4. Make sure you configured the basic Interface Setting behavior. See "Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident" (on page 599) for more information.

5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the New icon, and continue.
   b. To edit an Enrichment configuration, click the Open icon, and continue.
   c. To delete an Enrichment configuration, click the Delete icon.
7. Configure the desired Enrichment behavior (see the "Enrich Configuration Attributes" table)
8. Click **Save and Close** to save your changes and return to the previous form.

Interface Settings Enrichment Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: • Accounting • Application Status • Configuration • Fault • Performance • Security • Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443) for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Possible values include: • Address • Aggregated Port • Board • Connection • Correlation • HSRP • Interface • Node • OSPF See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443) for more information.
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below:

Name	Description
	Normal - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there may or may not be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: **None** **Low** **Medium** **High** **Top** **Note**: The icons are displayed only in table views.
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact See Incident Form: General Tab for more information.

Name	Description
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right. You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (Remote NNM 6.x/7.x Events)" (on page 592) "Include Custom Incident Attributes in Your Message Format (Remote NNM 6.x/7.x Events)" (on page 597)
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 2048 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Configure Custom Incident Attributes to Enrich an Incident Configuration (Interface Settings) (Remote NNM 6.x/7.x Events)

The Custom Incident Attributes (CIAs) tab enables you to specify additional CIAs to be saved with an incoming incident. The persisted data might then be used as an argument to an action for the incident.

When creating a CIA for an incident configuration, you can specify any of the following values:

- Custom Attribute on the source node
- Custom Attribute on the interface (source object)
- Object Attribute (Object Attribute Path)

When specifying an Object Attribute, use the same syntax as when specifying a URL object for an Object-Type Attribute. See "Attributes per Object Type for Full URLs" (on page 883) for more information.

You also specify the Custom Incident Attribute name that will store this information.

For information about each Enrichment tab:

To create a Custom Incident Attribute to enrich an incident configuration:

1. Navigate to the **Remote NNM 6/x.7.x Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Remote NNM 6.x/7.x Events** tab.
    d. Do one of the following:

i. To create an incident configuration, click the New icon, and continue.

ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

iii. To delete an incident configuration, click the Delete icon.

2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, and click the Open icon.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident" (on page 599) for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the New icon, and continue.
   b. To edit an Enrichment configuration, click the Open icon, and continue.
   c. To delete an Enrichment configuration, click the Delete icon.
7. Make sure the Enrichment settings are configured. See "Configure Incident Enrichment Settings for an Interface Group (Remote NNM 6.x/7.x Events)" (on page 608) for more information.
8. Navigate to the **Custom Incident Attributes** tab.
9. Do one of the following:
   a. To create a Custom Incident Attribute, click the New icon, and continue.
   b. To edit a Custom Incident Attribute, click the Open icon, and continue.
   c. To delete a Custom Incident Attribute, click the Delete icon.
10. Configure the Custom Incident Attribute (see the "Custom Incident Attribute" table).
11. Click **Save and Close** to save your changes and return to the previous form.

Custom Incident Attribute

Name	Description
Custom Incident Attribute Name	Name used to identify the Custom Incident Attribute you are configuring. The name limit is 80 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. No spaces are allowed.
Type	Specifies whether you are using a Custom Attribute on a node, a Custom Attribute on an interface, or an Object Attribute. Possible values are: • Node Custom Attribute • Interface Custom Attribute • Object Attribute Path

Name	Description
Custom Attribute Name or Object Attribute Path	Used to determine the value to be assigned to the Custom Incident Attribute you are configuring. Enter either of the following: • Name of the Custom Attribute on the source node • Name of the Custom Attribute on the interface (source object) • Object Attribute Path When specifying an Object Attribute Path, use the same syntax as when specifying a URL object for an Object-Type Attribute; for example ${deviceProfile.SNMPObjectID}. See "Attributes per Object Type for Full URLs" (on page 883) for more information.

Configure a Payload Filter to Enrich an Incident Configuration (Interface Settings) (Remote NNM 6.x/7.x Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be enriched. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Remote NNM 6/x.7.x Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Remote NNM 6.x/7.x Events** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.
4. Make sure you configured the basic Interface Setting behavior. See "Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident" (on page 599) for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:

a. To create an Enrichment configuration, click the New icon, and continue.

b. To edit an Enrichment configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

c. To delete an Enrichment configuration, click the Delete icon.

7. Make sure the Enrichment settings are configured. See "Configure Incident Enrichment Settings for an Interface Group (Remote NNM 6.x/7.x Events)" (on page 608) for more information.
8. Select the **Payload Filter** tab.
9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

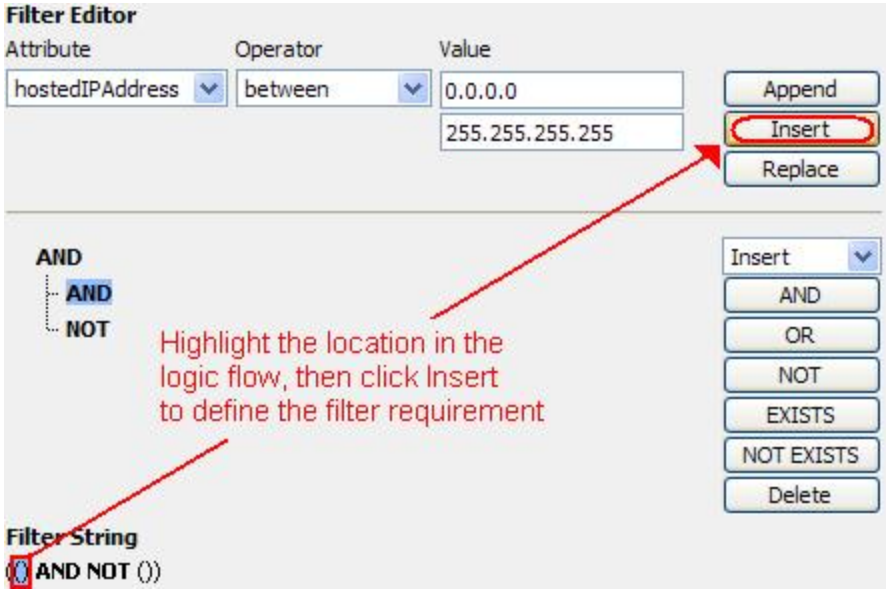
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



10. Click **Save and Close**.
11. Click **Save and Close** to save your changes and return to the previous form.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Operator	Valid operators are described below. ■ **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. ■ **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. ■ **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. ■ **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. ■ **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between` matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. ■ **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`

Attribute	Description

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not have a value.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Examples:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters.

  `ciaValue like *Chicago*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**.

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

Attribute	Description

matches any incident that contains a varbind with values other than **1** and **2**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**.

  `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Value	The value for which you want NNMi to search. Note the following: ■ The values you enter are case sensitive. ■ NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. ■ The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Incident Dampening Settings for an Interface Group (Remote NNM 6.x/7.x Events)

Note: Interface Settings override any other Dampening settings for this incident, including those from the Node Settings tab.

NNMi enables you to delay the following for an incident configuration based on the Source Object's participation in an Interface Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

Note: You can also suppress the incident configuration based on the Source Node's participation in a Node Group. See "Configure Incident Dampening Settings for a Node Group (Remote NNM 6.x/7.x Events)" (on page 657) for more information.

Tip: See "Create Interface Groups" (on page 197) for more information about Interface Groups.

For information about each Interface Settings tab:

When using the Dampening configuration, note the following:

- By default, the Dampen Interval for Remote NNM 6.x/7.x Events is six minutes.

  Note: Any Dampen Interval set for less than six minutes might not provide enough time for the Causal Engine to complete its analysis. As a result, incidents associated with intermediate symptoms might appear as root cause incidents.

- NNMi initially assigns incidents with Dampening settings configured a Lifecycle State of DAMPENED.
- After the Dampen Interval has passed, NNMi changes the Lifecycle State to REGISTERED.

See About the Incident Lifecycle for more information about Lifecycle State.

To configure Dampening for an incident based on an Interface Group:

1. Navigate to the **Remote NNM 6/x.7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **Remote NNM 6.x/7.x Events** tab.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete an incident configuration, click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, and click the Open icon.
4. Make sure you configured the basic Interface Setting behavior. See "Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident" (on page 599) for more information.
5. Select the **Dampening** tab.
6. Configure the desired Dampening behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Interface Settings Dampening Configuration Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's dampening settings. To temporarily disable the Dampening Configuration settings for the selected incident configuration, clear **Enable** ☐. To enable the Dampening Configuration settings for the selected incident configuration, click **Enable** ☑.
Hour	Specifies the number of hours to be used for the dampen interval.
Minutes	Specifies the number of minutes to be used for the dampen interval. **Note**: The maximum dampen interval is 60 minutes.
Seconds	Specifies the number of seconds to be used for the dampen interval.

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	
	When creating a Payload Filter, note the following:
	
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	
	The following example filters incidents on voltage state:
	
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`ciaValue = 5`
	
	NNMi evaluates the expression above as follows:
	
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Name	Description

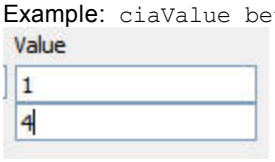
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all values equal to and between the two values specified. Click here for an example.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

Example: `ciaValue between`



matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line. selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line. selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.
Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples. **Example 1** `AND`

Name	Description

Button	Description

```
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Name	Description

Button	Description

	Example 2
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Example 2
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Example 2
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`

Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Name Desc

Configure Incident Actions for an Interface Group (Remote NNM 6.x/7.x Event)

Note: Interface Settings override any other Actions settings for this incident, including those from the Node Settings tab.

For information about each Interface Settings tab:

NNMi enables you to configure incident actions based on a Source Object's participation in an Interface Group.

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 (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

You can configure actions for incidents generated from SNMP traps, NNM 6.x or 7.x events, and the NNMi management events. Any time an incident configuration changes, the action directory is rescanned and any Jython files are reloaded to the NNMi database. See "Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" (on page 717) for more information about the actions directory.

Tip: Copy any required Jython files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" (on page 717) for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. See "Verify that NNMi Services are Running" (on page 47) for more information about log files and where they are located.

To configure an automatic action for an incident:

1. Navigate to the **Remote NNM 6/x.7.x Event Configuration** tab.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration.**
    c. Select the **Remote NNM 6.x/7.x Events** tab.
    d. Do one of the following:
        i. To create a new incident configuration, click the New icon, and continue.
        ii. To edit an existing incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
2. Select the **Interface Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
4. Make sure you configured the basic Interface Setting behavior. See "Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident" (on page 599) for more information.
5. Select the **Actions** tab.

6. From the **Lifecycle Actions** table toolbar, do one of the following:
   - To create an Action configuration, click the New icon, and continue.
   - To edit an Action configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an Action configuration, click the Delete icon.
7. In the "Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" (on page 717), provide the required information.
8. Click **Save and Close** to save your changes and return to the previous form.

   The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type.

Configure a Payload Filter for an Incident Action (Interface Settings) (Remote NNM 6.x/7.x Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Remote NNM 6/x.7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **Remote NNM 6.x/7.x Events** tab.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete an incident configuration, click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit a configuration, click the Open icon in the row representing the configuration you want to edit.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident" (on page 599) for more information.
5. Select the **Actions** tab.
6. Do one of the following:

a. To create an Action configuration, click the New icon, and continue.

b. To edit an Action configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

c. To delete an Action configuration, click the Delete icon.

7. Make sure the Action Configuration settings are configured. See "Configure Incident Actions for an Interface Group (Remote NNM 6.x/7.x Event)" (on page 628) for more information.
8. Select the **Payload Filter** tab.
9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

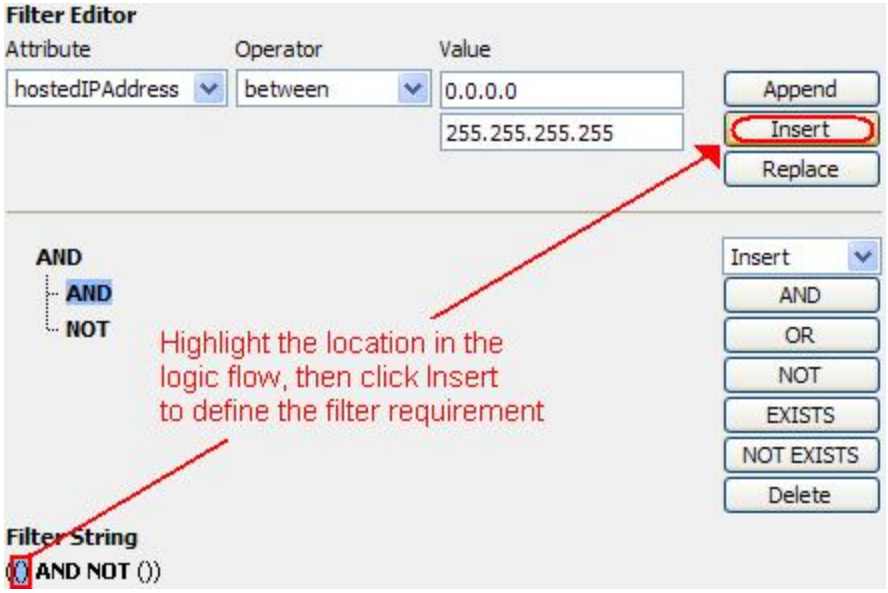
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



10. Click **Save and Close**.
11. Click **Save and Close** to save your changes and return to the previous form.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Operator	Valid operators are described below. ■ **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. ■ **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. ■ **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. ■ **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. ■ **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between` matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. ■ **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`

Attribute	Description

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not have a value.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Examples:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters.

  `ciaValue like *Chicago*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**.

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

Attribute	Description

matches any incident that contains a varbind with values other than **1** and **2**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**.

  `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Value	The value for which you want NNMi to search. Note the following: ■ The values you enter are case sensitive. ■ NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. ■ The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Node Settings for a Remote NNM 6.x/7.x Event Incident

Note: Node Settings override any other Suppression, Enrichment, Dampen, Actions, or Diagnostics Selections configuration settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to apply an incident configuration to a Source Node based on the Source Node's participation in a Node Group. If the Source Node is not a member of the Node Group specified, the incident is neither displayed nor stored in the NNMi database.

Tip: See "Create Node Groups" (on page 181) for more information about Node Groups.

For information about each Node Settings tab:

For information about each Remote NNM 6.x/7.x Events tab:

To apply an incident configuration to a Source Node based on the Source Node's Node Group:

1. Navigate to the **Remote NNM 6/x.7.x Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Remote NNM 6.x/7.x Events** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.
4. Configure the desired Node Settings (see table).
5. Click **Save and Close** to save your changes and return to the previous form.

Node Group Attributes

Name	Description
Node Group	Click the Lookup icon and select Quick Find to select the Node Group you want to use. See "Use the Quick Find Window" (on page 28) for more information about using Quick Find.
Ordering	Determines the priority order for those nodes that appear in multiple Node Groups. The lower the number, the higher the priority. For example, **1** is the highest priority. If a node is in multiple Node Groups and more than one of those Node Groups have been specified in an incident configuration, only the incident configuration with the highest priority will be applied to the node.
Enable	Use this attribute to temporarily disable an incident's suppression settings. To temporarily disable the Node Group settings for the selected incident configuration, clear **Enable**. To enable the Node Group settings for the selected incident configuration, click **Enable**.

Configure Incident Suppression Settings for a Node Group (Remote NNM 6.x/7.x Events)

Note: Node Settings override any other Suppression settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to suppress a specified incident configuration based on the Source Node's participation in a Node Group.

Note: You can also suppress the incident configuration based on the Source Object's participation in an Interface Group. See "Configure Incident Suppression Settings for an Interface Group (SNMP Trap Incident)" (on page 454) for more information.

Tip: See "Create Node Groups" (on page 181) for more information about Node Groups.

For information about each Node Settings tab:

To suppress an incident configuration based on a Node Group:

1. Navigate to the **Incident Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Remote NNM 6/x.7.x Event Configuration**.
    c. Select the **Remote NNM 6.x/7.x Events**tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.
4. Make sure you configured the basic Node Setting behavior. See "Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" (on page 634) for more information.
5. Select the **Suppression** tab.
6. Configure the desired Suppression behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Suppression Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings. To temporarily disable the Suppression settings for the selected incident configuration, clear **Enable**☐. To enable the Suppression settings for the selected incident configuration, click **Enable**☑.

Name	Description

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	
	When creating a Payload Filter, note the following:
	
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	
	The following example filters incidents on voltage state:
	
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`ciaValue = 5`
	
	NNMi evaluates the expression above as follows:
	
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**.

- **<** Finds all values less than the value specified. Click here for an example.

  Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**.

- **<=** Finds all values less than or equal to the value specified. Click here for an example.

  Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

  Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

  `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

  varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

Example:

`ciaValue not in`

matches any incident that contains a varbind with values other than **1** and **2**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

  Example:

  `ciaValue not in`

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

matches any incident that contains a varbind with values other than **1** and **2**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line. selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line. selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.
Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples. **Example 1**

Name	Description

Button	Description

```
AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
```

Name	Description

Button	Description
	OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Name	Description

Configure Incident Enrichment Settings for a Node Group (Remote NNM 6.x/7.x Events)

Note: Node Settings override any other Enrichment settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to enhanced a specified incident configuration based on the Source Node's participation in a Node Group. The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To

To configure enrichment settings for a Node Group:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Remote NNM 6.x/7.x Events tab**.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.
4. Make sure you configured the basic Node Setting behavior. See "Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" (on page 634)for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
    a. To create an Enrichment configuration, click the New icon, and continue.
    b. To edit an Enrichment configuration, click the Open icon, and continue.
    c. To delete an Enrichment configuration, click the Delete icon.

7. Configure the desired Enrichment behavior (see the "Enrich Configuration Attributes" table)
8. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Enrich Configuration Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: • Accounting • Application Status • Configuration • Fault • Performance • Security • Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443) for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Possible values include: • Address • Aggregated Port • Board • Connection • Correlation • HSRP • Interface • Node • OSPF See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443) for more information.
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below: **Normal** - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there may or may not be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object that require further investigation.

Name	Description
	Major - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: **None** **Low** **Medium** **High** **Top** **Note**: The icons are displayed only in table views.
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact See Incident Form: General Tab for more information.

Name	Description
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right. You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (SNMP Trap Incident)" (on page 446) "Include Custom Incident Attributes in Your Message Format (SNMP Trap Incident)" (on page 452)
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 2048 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Configure Custom Incident Attributes to Enrich an Incident Configuration (Node Settings) (Remote NNM 6.x/7.x Events)

The Custom Incident Attributes (CIAs) tab enables you to specify additional CIAs to be saved with an incoming incident. The persisted data might then be used as an argument to an action for the incident.

When creating a CIA for an incident configuration, you can specify any of the following values:

- Custom Attribute on the source node
- Custom Attribute on the interface (source object)
- Object Attribute (Object Attribute Path)

When specifying an Object Attribute, use the same syntax as when specifying a URL object for an Object-Type Attribute. See "Attributes per Object Type for Full URLs" (on page 883) for more information.

You also specify the Custom Incident Attribute name that will store this information.

For information about each Enrichment tab:

To create a Custom Incident Attribute to enrich an incident configuration:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select **Remote NNM 6.x/7.x Events** tab.
    d. Do one of the following:

i. To create an incident configuration, click the New icon, and continue.

ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

iii. To delete an incident configuration, click the Delete icon.

2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, and click the Open icon.
4. Make sure you configured the basic Node Setting behavior. See "Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" (on page 634) for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the New icon, and continue.
   b. To edit an Enrichment configuration, click the Open icon, and continue.
   c. To delete an Enrichment configuration, click the Delete icon.
7. Make sure the Enrichment settings are configure. See "Configure Incident Enrichment Settings for a Node Group (Remote NNM 6.x/7.x Events)" (on page 647) for more information.
8. Navigate to the **Custom Incident Attributes** tab.
9. Do one of the following:
   a. To create a Custom Incident Attribute, click the New icon, and continue.
   b. To edit a Custom Incident Attribute, click the Open icon, and continue.
   c. To delete a Custom Incident Attribute, click the Delete icon.
10. Configure the Custom Incident Attribute (see the "Custom Incident Attribute" table).
11. Click **Save and Close** to save your changes and return to the previous form.

Custom Incident Attribute

Name	Description
Custom Incident Attribute Name	Name used to identify the Custom Incident Attribute you are configuring. The name limit is 80 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. No spaces are allowed.
Type	Specifies whether you are using a Custom Attribute on a node, a Custom Attribute on an interface, or an Object Attribute. Possible values are: • Node Custom Attribute • Interface Custom Attribute • Object Attribute Path

Name	Description
Custom Attribute Name or Object Attribute Path	Used to determine the value to be assigned to the Custom Incident Attribute you are configuring. Enter either of the following: • Name of the Custom Attribute on the source node • Name of the Custom Attribute on the interface (source object) • Object Attribute Path When specifying an Object Attribute Path, use the same syntax as when specifying a URL object for an Object-Type Attribute; for example ${deviceProfile.SNMPObjectID}. See "Attributes per Object Type for Full URLs" (on page 883) for more information.

Configure a Payload Filter to Enrich an Incident Configuration (Node Settings) (Remote NNM 6.x/7.x Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be enriched. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Remote NNM 6.x/7.x Events** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.
4. Make sure you configured the basic Node Setting behavior. See "Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" (on page 634) for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:

a. To create an Enrichment configuration, click the New icon, and continue.

b. To edit an Enrichment configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

c. To delete an Enrichment configuration, click the Delete icon.

7. Make sure the Enrichment settings are configure. See "Configure Incident Enrichment Settings for a Node Group (Remote NNM 6.x/7.x Events)" (on page 647) for more information.
8. Select the **Payload Filter** tab.
9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

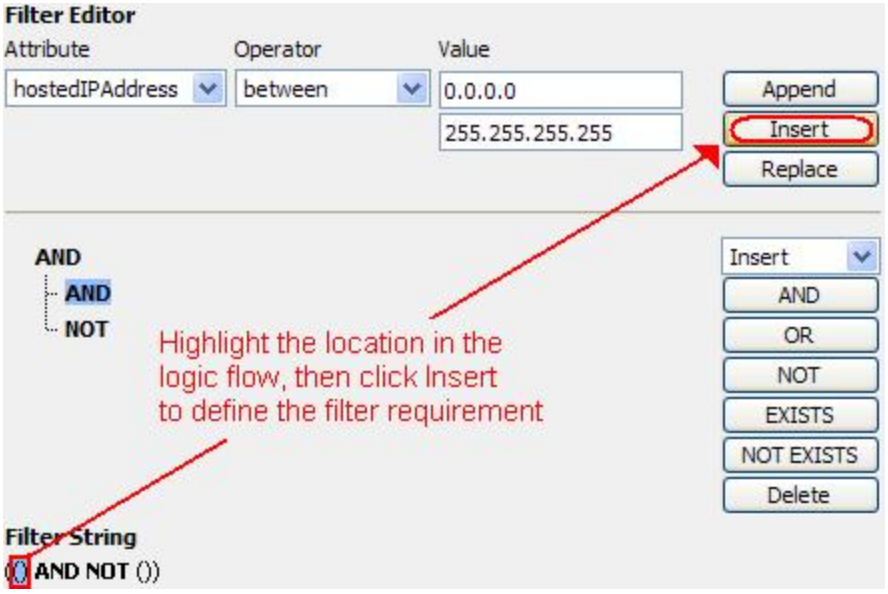
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



10. Click **Save and Close**.
11. Click **Save and Close** to save your changes and return to the previous form.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Operator	Valid operators are described below. ■ **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. ■ **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. ■ **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. ■ **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. ■ **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between` matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. ■ **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`

Attribute	Description

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not have a value.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Examples:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters.

  `ciaValue like *Chicago*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**.

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

Attribute	Description
	 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. ■ **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Value	The value for which you want NNMi to search. Note the following: ■ The values you enter are case sensitive. ■ NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. ■ The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Incident Dampening Settings for a Node Group (Remote NNM 6.x/7.x Events)

Note: Node Settings override any other Dampening settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to delay the following for an incident configuration based on the Source Node's participation in a Node Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

Note: You can also suppress the incident configuration based on the Source Object's participation in an Interface Group. See "Configure Incident Dampening Settings for an Interface Group (Remote NNM 6.x/7.x Events)" (on page 618) for more information.

Tip: See "Create Node Groups" (on page 181) for more information about Node Groups.

For information about each Node Settings tab:

When using the Dampening configuration, note the following:

- By default, the dampen interval for Remote NNM 6.x/7.x Events is six minutes.

  Note: Any dampen interval set for less than six minutes might not provide enough time for the Causal Engine to complete its analysis. As a result, incidents associated with intermediate symptoms might appear as root cause incidents.
- NNMi initially assigns incidents with Dampening settings configured a Lifecycle State of DAMPENED.
- After the dampen interval has passed, NNMi changes the Lifecycle State to REGISTERED.

See About the Incident Lifecycle for more information about Lifecycle State.

To configure Dampening for an incident based on a Node Group:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **Remote NNM 6.x/7.x Events** tab**.**
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete an incident configuration, click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, and click the Open icon.
4. Make sure you configured the basic Node Setting behavior. See "Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" (on page 634)for more information.
5. Select the **Dampen** tab.
6. Configure the desired Dampen behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Dampen Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's Dampening settings. To temporarily disable the Dampening settings for the selected incident configuration, clear **Enable** ☐. To enable the Dampening settings for the selected incident configuration, click **Enable** ☑.
Hour	Specifies the number of hours to be used for the dampen interval.
Minutes	Specifies the number of minutes to be used for the dampen interval. **Note**: Use a minimum of 6 minutes for the dampen interval. The maximum dampen interval is 60 minutes.
Seconds	Specifies the number of seconds to be used for the dampen interval.

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	
	When creating a Payload Filter, note the following:
	
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	
	The following example filters incidents on voltage state:
	
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`ciaValue = 5`
	
	NNMi evaluates the expression above as follows:
	
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Name	Description

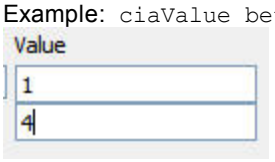
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all values equal to and between the two values specified. Click here for an example.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

Example: `ciaValue between`



matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line. selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line. selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.
Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples. **Example 1**

Name	Description

Button	Description

```
AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
```

Name	Description

Button	Description
	OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Name	Description

Configure Incident Actions for a Node Group (Remote NNM 6.x/7.x Events)

Note: Interface Settings override any other Actions settings for this incident, including those from the Node Settings tab.

For information about each Node Settings tab:

NNMi enables you to configure incident actions based on a Source Node's participation in a Node Group.

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 (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

You can configure actions for incidents generated from SNMP traps, NNM 6.x or 7.x events, and the NNMi management events. Any time an incident configuration changes, the action directory is rescanned and any Jython files are reloaded to the NNMi database. See "Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" (on page 717) for more information about the actions directory.

Tip: Copy any required Jython files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" (on page 717) for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. See "Verify that NNMi Services are Running" (on page 47) for more information about log files and where they are located.

To configure an automatic action for an incident:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** tab.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration.**
    c. Select the **Remote NNM 6.x/7.x Events** tab.
    d. Do one of the following:
        i. To create a new incident configuration, click the New icon.
        ii. To edit an existing incident configuration, click the Open icon in the row representing the configuration you want to edit.
2. Select the **Node Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon
4. Make sure you configured the basic Node Setting behavior. See "Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" (on page 634) for more information.
5. Select the **Actions** tab.
6. From the **Lifecycle Actions** table toolbar, do one of the following:

- To create an Action configuration, click the New icon, and continue.
- To edit an Action configuration, click the Open icon, and continue.
- To delete an Action configuration, click the Delete icon.

7. In the "Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" (on page 717), provide the required information.
8. Click **Save and Close** to save your changes and return to the previous form.

   The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type.

Configure a Payload Filter for an Incident Action (Node Settings) (Remote NNM 6.x/7.x Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **Remote NNM 6.x/7.x Events** tab.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete an incident configuration, click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   - To create a new configuration, click the New icon.
   - To edit a configuration, click the Open icon in the row representing the configuration you want to edit.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" (on page 634) for more information.
5. Select the **Actions** tab.
6. Do one of the following:
   - To create an Action configuration, click the New icon, and continue.
   - To edit an Action configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an Action configuration, click the Delete icon.

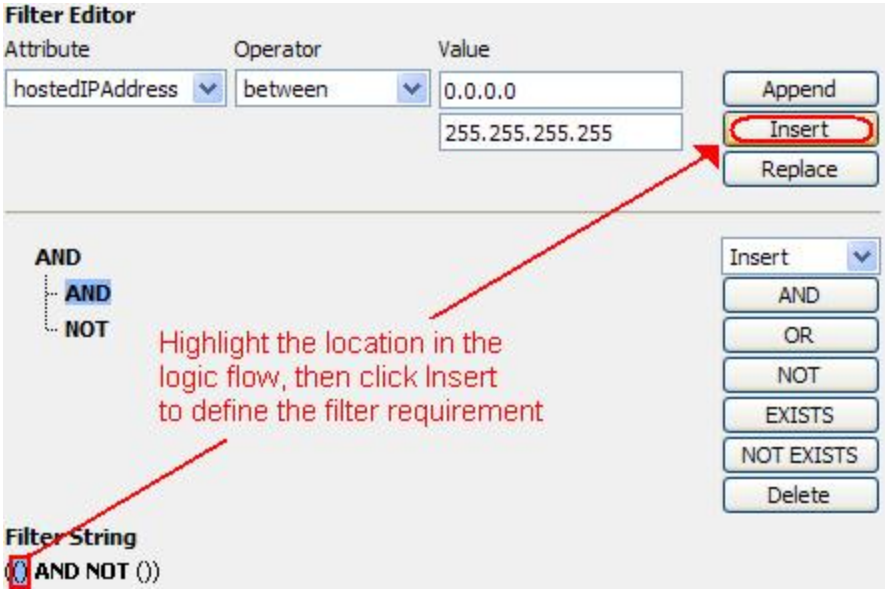
7. Make sure you configured the Action Configuration settings. See "Configure Incident Actions for a Node Group (Remote NNM 6.x/7.x Events)" (on page 667) for more information.
8. Select the **Payload Filter** tab.
9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".
    a. Plan out the logic needed for your Filter String.
    b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

       For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

       ```
       (( ) AND NOT ( ))
       ```

    c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

       For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



10. Click **Save and Close**.
11. Click **Save and Close** to save your changes and return to the previous form.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ▪ ciaName ▪ ciaValue
Operator	Valid operators are described below. ▪ **=** Finds all values equal to the value specified. Click here for an example.

Attribute	Description

Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**.

- **!=** Finds all values not equal to the value specified. Click here for an example.

  Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**.

- **<** Finds all values less than the value specified. Click here for an example.

  Example: `ciaValue < 6` matches any incident with a varbind value less than **6**.

- **<=** Finds all values less than or equal to the value specified. Click here for an example.

  Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

  Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

Attribute	Description

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not have a value.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Examples:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters.

  `ciaValue like *Chicago*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**.

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this*

Attribute	Description
	location. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Value	The value for which you want NNMi to search. Note the following: ■ The values you enter are case sensitive. ■ NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. ■ The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Diagnostics Selections for a Node Group (Remote NNM 6.x/7.x Events)

Note: Node Settings override any other Diagnostics Selections settings for this incident, except those configured on the Interface Settings tab.

For information about each Node Settings tab: .

(*HP Network Node Manager iSPI Network Engineering Toolset Software*) The Diagnostic Selections form enables you to configure NNMi to automatically gather NNM iSPI NET diagnostic information for the

Incident you are configuring. When using this form, you specify the diagnostics you want to run on each applicable node in the specified Node Group.

To configure Diagnostics to run on a Source Node for an incident:

1. Navigate to the **Diagnostics Selection** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration.**
    c. Select the **Remote NNM 6.x/7.x Events** tab.
    d. Do one of the following:
        - To create an Incident configuration, click the New icon, and continue.
        - To edit an Incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
    e. Navigate to **Node Settings** tab, and do one of the following:
        - To create a Node Settings configuration, click the New icon, and continue.
        - To edit a Node Settings configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        - To delete a Node Settings configuration, select the Node setting and click the Delete icon.
    f. Navigate to the **Diagnostic Selection** tab, and do one of the following:
        - To create a Diagnostic Selection setting, click the New icon, and continue.
        - To edit a Diagnostic Selection setting, click the Open icon in the row representing the Diagnostic Selection setting you want to edit, and continue.
        - To delete a Diagnostic Selection setting, select the Diagnostic Selection setting and click the Delete icon.
2. Provide the required information (see table).
3. Click **Save and Close** to save your changes and return to the previous form.

After you configure the Diagnostic for the incident and Node Group, the Diagnostic must match the following criteria before the Diagnostic runs:

- The Source Node must be in the specified Node Group.
- The Diagnostic must be valid for the Source Node. (For example, only Nortel switch Diagnostics are run on Nortel switches.)
- The incident's current lifecycle state must match a lifecycle state for which it was configured. (For example, if you configure the Incident to run a specified Diagnostic when the incident is Closed, then if the current Incident's Lifecycle State is Closed, NNMi runs that Diagnostic.)

Note: If a Source Node is in more than one Node Group, the Diagnostic is only run on the node the first time NNMi finds a match for that Source Node based on the configuration Ordering field.

If these criteria are met, NNM iSPI NET runs the Diagnostics and generates Diagnostic reports to help you solve the problem on the Source Node.

After you configure Diagnostics for an incident, you can also run Diagnostics and access the Diagnostics reports on demand, using **Actions → Run Diagnostics (iSPI Net only)** in the Incident form. The same criteria apply (see the criteria above). See Incident Form:Diagnostics Tab for more information.

You can also run and access Diagnostics reports from a Node form. See Node Form: Diagnostics Tab for more information.

Diagnostic Settings Attributes

Attribute	Description
Flow Definition	Select the Diagnostic (Flow Definition) you want to use for the specified Node Group. Click the Lookup icon and choose one of the following options: • Quick View to view summary information for the Flow Definition name displayed. • Quick Find to view the list of possible diagnostic Flow Definitions. NNMi provides diagnostics for the following types of devices: ▪ Cisco switch ▪ Cisco router ▪ Cisco switch/router ▪ Nortel switch See "Diagnostics (Flows) Provided by NNM iSPI NET" (on page 422) for more information about the diagnostics provided and the devices to which they apply.
Lifecycle State	Incident Lifecycle State of the target Incident. If the incident's Lifecycle State matches the value specified here, the Diagnostic runs. The Diagnostic automatically runs on each applicable Source Node in the specified Node Group if the incident has the Lifecycle State currently configured in this attribute of the Diagnostic (Flow Definition - set of automated commands). See About the Incident Lifecycle for more information about Lifecycle State.
Enable	Use this attribute to temporarily disable an incident's Diagnostics settings. To temporarily disable the selected Diagnostics settings, clear **Enable** ☐. To enable the selected Diagnostics settings, click **Enable** ☑.

Configure Suppression Settings for a Remote NNM 6.x/7.x Event Incident

For information about each Remote NNM 6.x/7.x Events tab:

NNMi enables you to suppress incidents based on Interface Group, Node Group, or default Suppression settings. NNMi applies your Suppression settings in the following order. Only the first match applies.

1. Interface Group (Remote 6.x/7.x Event Configuration Form: Interface Settings tab)
2. Node Group (Remote 6.x/7.x Event Configuration Form: Node Settings tab)

3. Enrich configuration settings without specifying an Interface Group or Node Group (Remote 6.x/7.x Event Configuration Form: Enrichment tab)

A Payload Filter allows you to use the data that is included with any of the following items before they are stored as incidents in NNMi:

- Traps generated from an SNMP agent
- Management incidents that are generated by NNMi
- Events generated by NNM 6.x or 7.x management stations

Examples of the type of data that can be used as a Payload Filter include SNMP trap varbind names and values as well as CIA (Custom Incident Attribute) names and values. For example, you might want NNMi to suppress a particular status change notification trap for a specified Node Group or Interface Group. To do so, you could include the name of the trap varbind that stores this information as well as the particular status change value string the traps that you want to suppress should contain.

See "Configure Incident Suppression Settings for an Interface Group (Remote NNM 6.x/7.x Events)" (on page 600) for information about how to suppress an incident for an Interface Group with or without a Payload Filter.

See "Configure Incident Suppression Settings for a Node Group (Remote NNM 6.x/7.x Events)" (on page 636) for more information about how to suppress an incident for a Node Group with or without a Payload Filter.

To configure suppression for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **Remote NNM 6.x/7.x Events** tab.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete an incident configuration, click the Delete icon.
2. Select the **Suppression** tab.
3. Provide the required information (see table)
4. Click **Save and Close** to save your configuration return to the previous form.

Suppression Configuration Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings. To temporarily disable the Suppression settings for the selected incident configuration, clear **Enable** ☐. To enable the Suppression settings for the selected incident configuration, click **Enable** ☑.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used

Name	Description

to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

When creating a Payload Filter, note the following:

- Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
- You must use a `ciaName` that already exists in the trap or event you are configuring.
- Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.

  The following example filters incidents on voltage state:

  ```
  AND
      ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
      ciaValue = 5
  ```

  NNMi evaluates the expression above as follows:

  ```
  (ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)
  ```

  NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

1.3.6.1.4.1.9.9.13.1.2.1.7.

- **<** Finds all values less than the value specified. Click here for an example.

  Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**.

- **<=** Finds all values less than or equal to the value specified. Click here for an example.

  Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

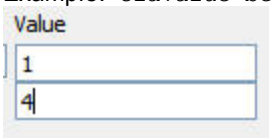
  Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

  `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

`ciaValue not in`

matches any incident that contains a varbind with values other than **1** and **2**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.

Name	Description

Button	Description
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.

Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples.
	
	Example 1
	
	`AND`
	`  ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6`
	`  OR`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3`
	
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:
	
	`AND`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6`
	`  OR`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3`
	
	Example 2
	
	`AND`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`

Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Configure Enrichment Settings for a Remote NNM 6.x/7.x Event Incident

For information about each Remote NNM 6.x/7.x Events tab:

NNMi enables you to fine tune and enhance incidents based on Interface Group, Node Group, or default Enrichment settings. NNMi applies your Enrichment settings in the following order. Only the first match applies.

1. Interface Group (Remote 6.x/7.x Event Configuration Form: Interface Settings tab)
2. Node Group (Remote 6.x/7.x Event Configuration Form: Node Settings tab)
3. Enrichment configuration settings without specifying an Interface Group or Node Group (Remote 6.x/7.x Event Configuration Form: Enrichment tab.)

The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To

Note: Any configuration you specify for Severity, Priority, or Message overrides those values provided in the Remote NNM 6.x/7.x Event Configuration Form: Basics information.

A Payload Filter allows you to use the data that is included with any of the following items before they are stored as incidents in NNMi:

- Traps generated from an SNMP agent
- Management incidents that are generated by NNMi
- Events generated by NNM 6.x or 7.x management stations

Examples of the type of data that can be used as a Payload Filter include SNMP trap varbind names and values as well as CIA (Custom Incident Attribute) names and values. For example, you might want NNMi to suppress a particular status change notification trap for a specified Node Group or Interface Group. To do so, you could include the name of the trap varbind that stores this information as well as the particular status change value string the traps that you want to suppress should contain.

See "Configure Incident Enrichment Settings for an Interface Group (Remote NNM 6.x/7.x Events)" (on page 608) for information about how to enrich an incident for an Interface Group with or without a Payload Filter.

See "Configure Incident Enrichment Settings for a Node Group (Remote NNM 6.x/7.x Events)" (on page 647) for more information about how to enrich an incident for a Node Group with or without a Payload Filter.

To configure Enrich Settings for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Remote NNM 6.x/7.x Events** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Enrichment** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.
4. Provide the required information (see table)
5. Click **Save and Close** to save your changes and return to the previous form.

Enrichment Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: • Accounting • Application Status • Configuration • Fault • Performance • Security • Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443) for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Possible values include: • Address • Aggregated Port • Board • Connection • Correlation • HSRP

Name	Description
	• Interface • Node • OSPF See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443) for more information.
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below: **Normal** - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there may or may not be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: 5 **None** 4 **Low** 3 **Medium** 2 **High** 1 **Top** **Note**: The icons are displayed only in table views.

Name	Description
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact See Incident Form: General Tab for more information.
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right. You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (SNMP Trap Incident)" (on page 446) "Include Custom Incident Attributes in Your Message Format (SNMP Trap Incident)" (on page 452)
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 2048 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Configure Dampening Settings for a Remote NNM 6.x/7.x Event Incident

For information about each Remote NNM 6.x/7.x Events tab:

NNMi enables you to delay the following for an incident configuration based on the Source Object's participation in an Interface Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

You can dampen incidents based on Interface Group, Node Group, or default Dampening settings. NNMi applies your Dampening settings in the following order. Only the first match applies.

1. Interface Group (Remote 6.x/7.x Event Configuration Form: Interface Settings tab)
2. Node Group (Remote 6.x/7.x Event Configuration Form: Node Settings tab)
3. Dampening settings without specifying an Interface Group or Node Group (Remote 6.x/7.x Event Configuration Form: Dampening tab)

When using the Dampening configuration, note the following:

- Duplicate and Rate Correlation incidents inherit the Dampening settings from its Correlated Children. If the Correlated Children are Closed while Dampened, and therefore deleted, NNMi retains the parent Duplicate or Rate Correlation incident. See "Correlate Duplicate Incidents (Deduplication Configuration)" (on page 346) and "Track Incident Frequency (Rate: Time Period and Count)" (on page 347) for more information about Duplicate and Rate Correlation incidents.

  Note: To view the number of incidents Closed and deleted while dampened, access the NNMi System Health Report Dampened Metrics (**Help** → **System Information** → **Health** tab).

- After the Dampen Interval has passed, NNMi changes the Lifecycle State to REGISTERED.

  See About the Incident Lifecycle for more information about Lifecycle State.

See "Configure Incident Dampening Settings for an Interface Group (Remote NNM 6.x/7.x Events)" (on page 618) for information about how to configure Dampening for an Interface Group with or without a Payload Filter.

See "Configure Incident Dampening Settings for a Node Group (Remote NNM 6.x/7.x Events)" (on page 657) for more information about how to configure Dampening for a Node Group with or without a Payload Filter.

To configure Dampening settings for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **Remote NNM 6.x/7.x Events** tab.
   d. Do one of the following:
      i. To create a configuration, click the New icon, and continue.
      ii. To edit configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete a configuration, click the Delete icon.
2. Select the **Dampening** tab.
3. Provide the required information (see table)
4. Click **Save and Close** to save your changes and return to the previous form.

Dampening Attributes

Name	Description
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

- Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex.Pattern class).

Name	Description
Hour	Specifies the number of hours to be used for the Dampen Interval.
Minutes	Specifies the number of minutes to be used for the Dampen Interval. Note the following: ■ The maximum Dampen Interval is 60 minutes. ■ Any Dampen Interval set for less than six minutes might not provide enough time for the Causal Engine to complete its analysis. As a result, Incidents associated with intermediate symptoms might appear as Root Cause Incidents.
Seconds	Specifies the number of seconds to be used for the Dampen Interval.

Name	Description

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	
	When creating a Payload Filter, note the following:
	
	■ Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	
	■ You must use a `ciaName` that already exists in the trap or event you are configuring.
	
	■ Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	
	■ View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	
	■ The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	
	The following example filters incidents on voltage state:
	
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`ciaValue = 5`
	
	NNMi evaluates the expression above as follows:
	
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	
	■ The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	
	■ The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Operator	Valid operators are described below. ■ **=** Finds all values equal to the value specified. Click here for an example.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue

Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**.

- **!=** Finds all values not equal to the value specified. Click here for an example.

  Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**.

- **<** Finds all values less than the value specified. Click here for an example.

  Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**.

- **<=** Finds all values less than or equal to the value specified. Click here for an example.

  Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

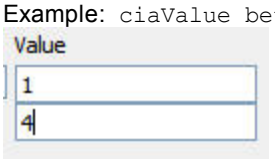
  Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue

■ **in** Finds any match to at least one value in a list of values. Click here for an example.

Example:

```
ciaValue in
```

Value
4
5

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

■ **is not null** Finds all non-blank values. Click here for an example.

Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

■ **is null** Finds all blank values. Click here for an example.

Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

■ **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for more information.

The period asterisk (.*) characters mean *any number of characters of any type at this location*.

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue

`ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
	string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: ■ The values you enter are case sensitive. ■ NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. ■ The `between, in` and `not in` operators require that each value be entered on a separate line.

Name	Description

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.

Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples.
	Example 1
	`AND`
	`ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6`
	`OR`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6`
	`OR`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3`
	Example 2
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`

Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Name	Desc

Configure Forward to Global Managers Settings for a Remote 6.x/7.x Event Incident (*NNMi Advanced*)

For information about each Remote 6.x/7.x Events tab:

(*NNMi Advanced - Global Network Management feature*) The NNMi Global Network Management feature enables multiple NNMi management servers to work together while managing different geographic areas of your network. See NNMi's Feature (NNMi Advanced) for more information. The Global Manager combines topology information from multiple Regional Managers, but maintains *a separate set of incidents about those nodes*.

Use the Global Manager Forwarding tab when you want to forward specific NNM 6.x/7.x Events from your NNMi management server (a Regional Manager) to all Global Managers in your Global Network Management environment.

Caution: The Global Manager must have an incident configuration for that NNM 6.x/7.x Event, otherwise the in-coming NNM 6.x/7.x Event is dropped. See "Export and Import Configuration Settings" (on page 1022) for ideas about sharing incident configurations among NNMi management servers.

When you configure Forward to Global Managers, you can specify an optional Payload Filter for NNMi to use when determining *which occurrences* should be forwarded to Global Managers. Payload Filters enables you to use the data that is included with an occurrence of an incident configuration before it is stored as an incident in the NNMi database.

Examples of the type of data that can be used as a Payload Filter include Custom Incident Attribute names (ciaName) and values (ciaValue). For example, you might want NNMi to forward an incident based on a particular status change notification trap. To do so, you would specify a Payload Filter that includes the name of the Custom Incident Attribute that stores the status information as well as the status change value string of interest.

To configure forwarding to Global Managers:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **Remote NNM 6.x/7.x Events** tab.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete an incident configuration, click the Delete icon.
2. Select the **Forward to Global Managers** tab.
3. Provide the required information (see table)
4. Click **Save and Close** to save your changes and return to the previous form.

Forwarding Configuration Attributes

Name	Description
Enable	Use this attribute to enable or temporarily disable an incident's GNM settings.

Name	Description
	To temporarily disable the GNM Forwarding Configuration settings for the selected incident configuration, clear **Enable** ☐. To enable the GNM Forwarding settings for the selected incident configuration, click **Enable** ☑.

Name	Description

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that NNMi forwards to other servers. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	
	When creating a Payload Filter, note the following:
	
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	
	The following example filters incidents on voltage state:
	
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`ciaValue = 5`
	
	NNMi evaluates the expression above as follows:
	
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	
	NNMi finds all incidents with a varbind value of `.1.3.6.1.4.1.9.9.13.1.2.1.7` and CIA value of **5**.
	
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

- **!=** Finds all values not equal to the value specified. Click here for an example.

  Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**.

- **<** Finds all values less than the value specified. Click here for an example.

  Example: `ciaValue < 6` matches any incident that contains a varbind value less than **6**.

- **<=** Finds all values less than or equal to the value specified. Click here for an example.

  Example: `ciaValue <= 6` matches any incident that contains a varbind value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

  Example: `ciaValue > 4` matches any incident that contains a varbind value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

matches any incident that contains a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with no varbind values.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

  `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

for an example.

Example:

```
ciaValue not in
```

matches any incident that contains a varbind with values other than **1** and **2**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

  for an example.

  Example:

  ```
  ciaValue not in
  ```

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

matches any incident that contains a varbind with values other than **1** and **2**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

matches any incident that contains a varbind with values other than **1** and **2**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

- **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

- **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

- **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Name	Description

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.

Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples.
	Example 1
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6`
	`OR`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6`
	`OR`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3`
	Example 2
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`

Delete	Deletes the selected expression.

Name	Description
	<table><tr><th>Button</th><th>Description</th></tr><tr><td></td><td>**Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr></table>

Name | Description

Configure Deduplication for a Remote NNM 6.x/7.x Event Incident

For information about each Remote 6.x/7.x Events tab:

The deduplication configuration determines what values NNMi should match to detect when an SNMP trap, management event, or remote NNM 6.x/7.x event is a duplicate.

Note the following:

- NNMi applies only one deduplication configuration per incident . If NNMi generates an incident using a specified deduplication configuration, NNMi continues to correlate duplicate incidents using the original configuration. To use a different deduplication configuration for an incident, first delete the current deduplication incident (created using the original deduplication configuration). NNMi generates the next deduplication incident according to the new deduplication configuration settings.
- By default, NNMi updates the **Duplicate Count** every 30 seconds. This interval cannot be changed.
- NNMi continues to update the duplicate count regardless of an incident's lifecycle state. For example, if an incident's **Lifecycle State** is set to **Closed**, the duplicate count continues to be incremented. See About the Incident Lifecycle for more information. This behavior helps you identify situations in which the incident is not yet fixed. Take note if the Duplicate Count is incremented after a lengthy time period has elapsed, which might indicate there is a new problem with the node, interface, or address.
- Each time you stop and restart ovjboss, any incidents that have not yet been correlated or persisted are lost. This means that after a restart of ovjboss, an incoming incident might not be correlated as expected. For example, after a restart of ovjboss, a duplicate incident might not be correlated under its original parent incident. Instead, a new parent incident might be generated. See "Stop or Start an NNMi Process" (on page 45)for more information about starting and stopping the ovjboss process.
- If a Duplicate Correlation Incident is dampened, note the following:
  - Duplicate Correlation Incidents inherit the Dampening settings from its Correlated Children.
  - NNMi always retains the Parent Duplicate Correlation incident, even if its Child Incidents are Closed and subsequently deleted.

    See "Dampening Incident Configurations" (on page 356) for more information about Dampening an Incident Configuration.

To specify or delete a deduplication configuration:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **Remote NNM 6.x/7.x Events** tab.
   d. Do one of the following:
      i. To create a deduplication configuration, click the New icon, and continue.
      ii. To edit a deduplication configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete a deduplication configuration, click the Delete icon.
2. Select the **Deduplication** tab.

3. Provide the required information (see "Deduplication Attributes" table).
4. Click **Save and Close** to save your changes and return to the previous form.

Deduplication Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's deduplication configuration. To temporarily disable the deduplication configuration setting, clear **Enable** ☐. To enable the deduplication configuration setting, click **Enable** ☑. **Note**: After a deduplication configuration is enabled, NNMi increments the **Duplicate Count** for an associated incident regardless of the **Lifecycle State** value. For example, if an incident's **Lifecycle State** is set to **Closed**, the duplicate count continues to be incremented. See About the Incident Lifecycle for more information.
Count	Specifies the number of duplicate incidents for the current configuration that NNMi stores at one time. For example, if the Count is 10, after NNMi receives 10 duplicate incidents, NNMi deletes the first (oldest) duplicate incident and keeps the eleventh. (NNMi stores ten maximum.)
Hour Interval	Used with the Minute and Second Intervals to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Hour Interval value is 1, and no Minute or Second Intervals are specified, and the duplicate incident is not generated within one hour, NNMi generates a new duplicate incident the next time it occurs.
Minute Interval	Used with the Hour and Second interval to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Minute Interval is 30 and no Hour or Second Intervals are specified, and the duplicate incident is not generated within 30 minutes, NNMi generates a new duplicate incident the next time it occurs.
Second Interval	Used with the Hour and Minute Intervals to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Second Interval is 120 and no Hour or Minute Intervals are specified, and the duplicate incident is not generated within 120 seconds, NNMi generates a new duplicate incident the next time it occurs.
Correlation Incident Config	Used to access the out-of-the box deduplicaton configuration provided by NNMi. Select the default value **Duplicate Correlation**. **Note**: You can choose to use this configuration as is or edit it. If you want to create a new deduplication configuration, you must create a new incident configuration. After you have created a new incident configuration, it appears in the **Quick Find** list of options. See "Lookup Fields" (on page 27) for more informationn about Quick Find.
Comparison Criteria	Specify the attribute values that must match before the incident is identified as a duplicate. The possible attributes consist of the following choices. • **Name** - The **Name** attribute value from the Incident form: General tab. • **CIA** - Represents any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " (on page 346): ▪ The **Value** attribute from the Incident form: Custom Attributes tab

Name	Description
	▪ An SNMP varbind Object ID ▪ An SNMP varbind position number If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " (on page 346)
	• **SourceNode** - The **Source Node** attribute value from the Basics attributes listed on the Incident form. The Source Node value is the IP Address or Name of the node for which the incident was generated. **Note**: The Source Node must be stored in the NNMi database. • **Source Object** - The **Source Object** attribute value from the Basics attributes listed on the Incident form. **Note**: The Source Object must be stored in the NNMi database. **Note**: Each attribute value in the option you select must match before the incident is identified as a duplicate. For example, if you select **Name**, only the Incident Name value must match. If you select **Name SourceNode SourceObject CIA**, the Incident Name, Source Node, Source Object, and all Custom Incident Attribute values that you configure as a Parameter Value must match before NNMi identifies the incident as a duplicate. Selecting an option that includes CIA enables you to further refine the deduplication criteria. For example, you might want to configure deduplication for incidents with CIA values that specify the same State attribute value for a particular network object. For a description of each Comparison Criteria option, click here.

Comparison Criteria	Description
Name	Value of the **Name** attribute from the Incident form: General tab must match.
Name CIA	Each of the following values must match: • **Name** attribute from the Incident form: General tab • **CIA** - Represents the Value associated with any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " (on page 346): ▪ Name of a Custom Incident Attribute (CIA) provided by NNMi. (See the Incident form: Custom Attributes tab ▪ An SNMP varbind Object ID ▪ An SNMP varbind position number If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " (on page 346)
Name SourceNode	**Note**: Select this option only if the Source Node is stored in the NNMi database. Each of the following values must match: • **Name** attribute from the Incident form: General tab • The **Source Node** attribute value from the Basics attributes listed on the Incident form
Name	**Note**: Select this option only if the Source Node is stored in the NNMi

Name attribute from the Incident form: General tab

Name	Description

Name	Description
Comparison Parameter List	*Optional*. If you selected a Comparison Criteria that includes CIA, you must populate one or more rows in this table. See "Deduplication Comparison Parameters Form " (on page 346).

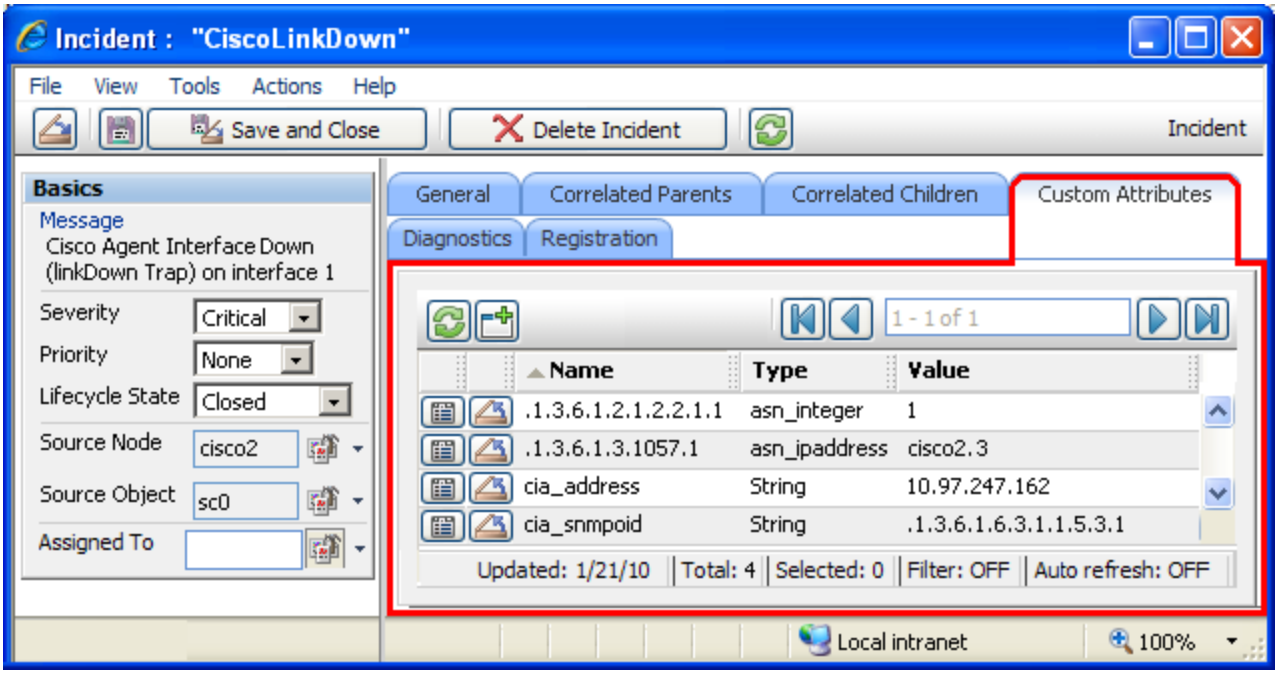
Deduplication Comparison Parameters Form (Remote NNM 6.x/7.x Events)

Comparison Parameter values enable accurate identification of duplicate incidents. Custom Incident Attributes (CIAs) are used as Comparison Parameter values. There are two categories of CIAs:

- SNMP trap varbind values (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312).

The group of available CIAs depends on which incident you are configuring for this Deduplication (for example, CiscoLinkDown). To see which CIAs are available, navigate to an Incident view, select an instance of that incident-type, click the Open icon and navigate to the Custom Attributes tab. The items listed in the table are the CIAs for that particular incident-type. For example, all CiscoLinkDown incidents would have the same group of CIAs shown in the illustration below.

Note: You can also use the CIA (varbind) position number.



To specify a CIA to use in the identification criteria for duplicate incidents:

1. Navigate to the **Deduplication Comparison Params** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Remote NNMi 6.x/7.x Events** tab.
    d. Do one of the following:

- To create a new configuration, click the New icon.
- To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.

e. On the form that opens, navigate to the **Deduplication** tab.

f. Locate the **Comparison Parameter List** table.

g. Do one of the following to specify which CIA:

- To add a Custom Incident Attribute parameter specification, click the New icon.
- To edit an existing Custom Incident Attribute parameter specification, click the Open icon in the row representing the configuration you want to edit.

2. In the Parameter Value field, type (or copy and paste) the exact text string from the Incident form, Custom Attribute tab, **Name** attribute value:
   - NNMi-provided CIA value (see "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312)).
   - SNMP trap varbind identified by the Abstract Syntax Notation value (ASN.1).
3. Click **Save and Close** to save your changes and return to the previous configuration form.

Configure Rate (Time Period and Count) for a Remote NNM 6.x/7.x Event Incident

For information about each Remote NNM 6.x/7.x Events tab:

Use Rate configuration to track incident patterns *based on the number of incident reoccurrences within a specified time period*. After the count within the specified time period is reached, NNMi emits a Rate Correlation incident and continues to update the Correlation Notes with the number of occurrences within that rate.

As long as your defined criteria (Count and Hours, Minutes, Seconds) is sustained, the following information is updated in the Correlation Notes of the Rate Correlation incident:

- the actual number of occurrences of incidents for that sustained rate (Count)
- the sustained time interval (Hours, Minutes, Seconds)

For example, you can set a Rate configuration to track when a link is intermittently down at least three times in 30 minutes. NNMi shows the first occurrence of the rate incident in the incident view and uses Correlation Notes to update the number of incidents and time interval to reflect all the incremental incident occurrences and time periods. To continue the example, if the rate of three times in 30 minutes is sustained for 90 minutes, NNMi updates the Correlation Notes to specify that 9 incidents occurred in 90 minutes.

NNMi provides preconfigured Rate correlations. You can add new Rate correlations.

When you open the Incident form of the newest instance:

- On the General tab, two fields notify you that the Rate correlation is working:
  - **Correlation Nature**: Rate
  - **Count**: x
- On the **Correlated Children** tab, each incident is listed in the table.

- If a Rate Correlation Incident is dampened, note the following:
  - Rate Correlation Incidents inherit the Dampening configuration settings from its Correlated Children.
  - NNMi always retains the Parent Rate Correlation Incident, even if its Child Incidents are Closed and subsequently deleted.

    See "Dampening Incident Configurations" (on page 356) for more information about Dampening an Incident Configuration.

To establish a rate correlation within an incident configuration:

1. Navigate to the **Rate** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Open the **Incident Configuration** form.
   d. Select the **Remote NNM 6.x/7.x Events** tab.
   e. Do one of the following:
      - To create a new configuration, click the New icon.
      - To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
   f. On the form that opens, locate the **Rate** tab.
2. Provide the definition for this Rate configuration (see the "Rate Configuration Definition" table).
3. *Optional*. If your Comparison Criteria includes custom incident attributes (CIA) to identify one specific incident, use the Comparison Parameter List table to define each CIA.See "Rate Comparison Parameters Form" (on page 347).
4. Click **Save and Close** to save your changes and return to the previous form.

Rate Configuration Definition

Attribute	Description
Enable	Use this attribute to temporarily disable an incident's rate settings. To temporarily disable the Dampen Configuration settings for the selected incident configuration, clear **Enable** ☐. To enable the Dampen Configuration settings for the selected incident configuration, click **Enable** ☑. If enabled, NNMi actively tracks any reoccurrences of the designated incident within the time period you specify, and generates a Rate incident.
Count	Specify the number of reoccurrences required before your Rate Configuration starts working.
Set the Time Period	Specify a time duration within which the reoccurrences are measured. Fill in one or more of the following attribute fields: **Hours** **Minutes**

Attribute	Description
	Seconds
Correlation Incident Config	Click the icon and select Quick Find. Select **Rate Correlation** from the list.
Comparison Criteria	Specify which group of attributes must match before the incident is identified as a duplicate. The possible groups of attributes consist of the following choices. **Name** value of the Incident (from the General tab on the Incident form). **Source Node** value (from the Basics group on the Incident form). Address or name of the node for which the incident was generated. **Source Object** value (from the Basics group on the Incident form). For example, the Source Object for a LinkDown incident is **interface**. **CIA** custom incident attribute values (select from the list displayed on the Custom Attributes tab on the Incident form). If you want to use CIA as part of your comparison criteria, see "Rate Comparison Parameters Form (Remote NNM 6.x/7.x Events)" (on page 714).
Comparison Parameter List	*Optional*. If you selected a Comparison Criteria that includes CIA, you must populate one or more rows in this table. See "Rate Comparison Parameters Form (Remote NNM 6.x/7.x Events)" (on page 714).

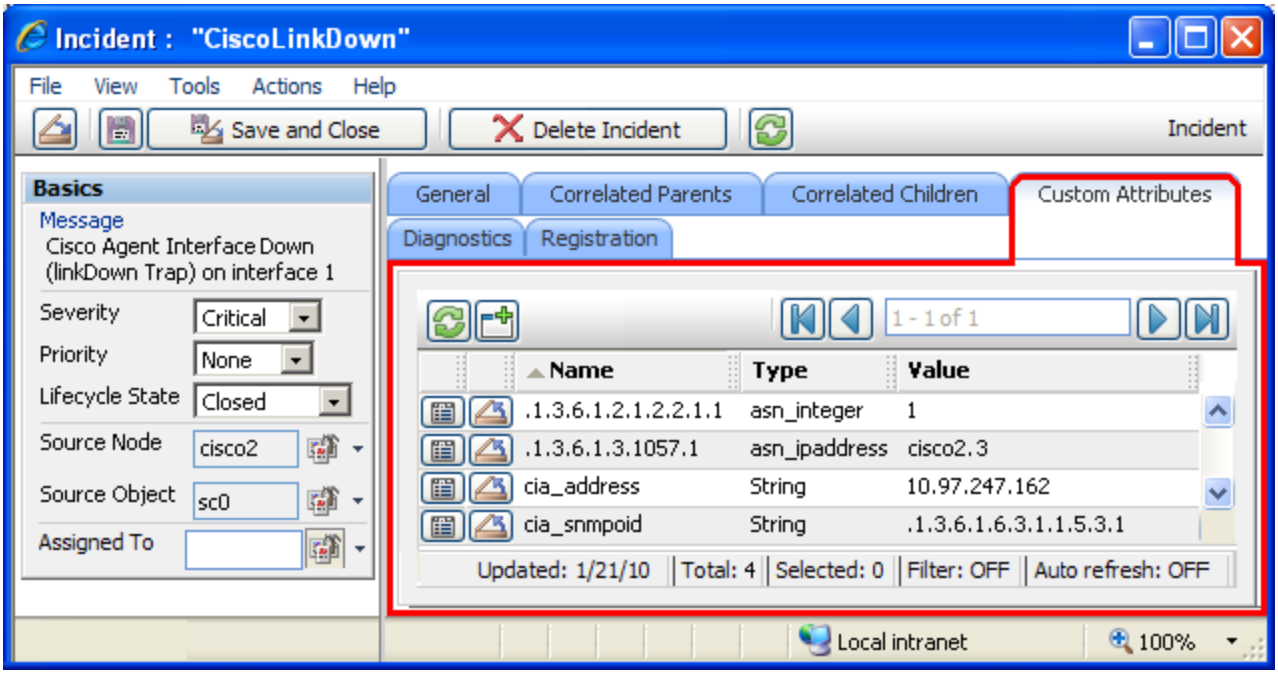
Rate Comparison Parameters Form (Remote NNM 6.x/7.x Events)

Custom Incident Attributes (CIAs) are used as parameter values. Parameter values enable accurate identification of duplicate incidents. There are two categories of CIAs:

- SNMP trap varbind values (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312).

The group of available CIAs depends on which incident you are configuring for this Rate (for example, CiscoLinkDown). To see which CIAs are available, navigate to an Incident view, select an instance of that incident-type, click the Open icon and navigate to the Custom Attributes tab. The items listed in the table are the CIAs for that particular incident-type. For example, all CiscoLinkDown incidents would have the same group of CIAs shown in the illustration below.

Note: You can also use the CIA (varbind) position number.



To specify a CIA to use in the identification criteria for duplicate incidents:

1. Navigate to the **Rate Comparison Params** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Remote NNMi 6.x/7.x Events** tab.
    d. Do one of the following:
        - To create a new configuration, click the New icon.
        - To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
    e. On the form that opens, navigate to the **Rate** tab.
    f. Locate the **Comparison Parameter List** table.
    g. Do one of the following to specify which CIA:
        - To add a Custom Incident Attribute parameter specification, click the New icon.
        - To edit an existing Custom Incident Attribute parameter specification, click the Open icon in the row representing the configuration you want to edit.
2. In the Parameter Value field, type (or copy and paste) the exact text string from the Incident form, Custom Attribute tab, **Name** attribute value:
    - NNMi-provided CIA value (see "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312)).
    - SNMP trap varbind identified by the Abstract Syntax Notation value (ASN.1).
3. Click **Save and Close** to save your changes and return to the previous configuration form.

Configure Actions for a Remote NNM 6.x/7.x Event Incident

For information about each Remote NNM 6.x/7.x Events tab:

For information about each Action tab:

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 (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

You can configure actions for incidents generated from SNMP traps, NNM 6.x or 7.x events, and the NNMi management events. Any time an incident configuration changes, the action directory is rescanned and any Jython files are reloaded to the NNMi database. See "Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" (on page 717) for more information about the actions directory.

Tip: Copy any required Jython files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" (on page 717) for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. To view the contents of the Actions log, use the **Tools** → **Incident Actions Log** menu option.

See "Verify that NNMi Services are Running" (on page 47) for more information about log files and where they are located.

NNMi sets the default values described in the following table. For information about how to change the default values, see the HP Network Node Manager i Software Deployment Reference, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

Action Server Properties

Property	Description	Default Value
numProcess	Number of actions that can be run at one time.	150
processTimeout	Amount of time in seconds that an action can run before the action server terminates the action.	300
numJythonThreads	Number of threads the action server uses to run Jython scripts	10
userName	User name under which the action server runs.	bin

To configure an automatic action for an incident:

1. Navigate to the **Actions** tab.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration.**
    c. Select the **Remote NNM 6.x/7.x Events** tab.

d. Do one of the following:

- To create an incident configuration, click the New icon, and continue.
- To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue
- To delete an incident configuration, select a row, and click the Delete icon.

e. Select the **Actions** tab.

2. From the **Lifecycle Actions** table toolbar, do one of the following:

- To create an Action configuration, click the New icon, and continue.
- To edit an Action configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
- To delete an Action configuration, click the Delete icon.

3. In the "Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" (on page 717), provide the required information.

4. Click **Save and Close** to save your changes and return to the previous form.

   The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type .

Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)

For information about each Action tab:

Use this form to enter the command you want to run when an incident of the type you are configuring is at a particular Lifecycle State. For example, when an incident is generated (**Registered**), you might want to automatically open a trouble ticket or email or page your network operator.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable on the Actions tab or using the **Actions** → **Enable Configuration** option.

To configure an action for an incidents:

1. Navigate to the **Lifecycle Transition Actions** form:

   a. From the workspace navigation pane, select the **Configuration** workspace.

   b. Select **Incident Configuration**.

   c. Select the **Remote NNM 6.x/7.x Events** tab.

   d. Select the **Actions** tab.

   e. From the **Lifecycle Transition Action** table toolbar, do one of the following:

   - To create an Action configuration, click the New icon, and continue.
   - To edit an Action configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an Action configuration, click the Delete icon.

2. Make your configuration choices (see table).

Note: NNMi reloads the configuration information anytime the incident configuration is changed.

3. Click **Save and Close** to save your changes and return to the previous form.

Create Action Attributes

Attribute	Description
Lifecycle State	Select a Lifecycle State from the drop-down menu.
Command Type	If you provided a Jython command, select **Jython** from the drop-down list. If you are using an executable or bat file, select **ScriptOrExecutable** from the drop-down list.

Command	Enter one of the following:
	• A Jython method with the required parameters
	• Executable command for the current operating system with the required parameters.
	When entering a **Command** value, note the following:
	• Left or right bracket ([ ]) and backtick (` Unicode character: 0060 hex = 96 dec) characters are not allowed in the **Command** attribute. If you need these characters in your shell script, place them in a shell script file and reference that file from the **Command** attribute.
	• **Windows only**: Shell commands are not allowed in the **Command** attribute. If you need to use shell commands, place them in a shell script file and reference that file from the **Command** attribute.
	• Verify that you do not have two Jython methods with the same name. Otherwise, NNMi is not able to tell which is the correct method to load.
	• You can use the same Jython method for more than one incident configuration.
	• Python files or other executable scripts need to reside in the following directory:
	Note: Use absolute paths to executables instead of relying on the PATH variable as it might not be set correctly.
	Windows:
	`%NnmDataDir%\shared\nnm\actions`
	UNIX:
	`/var/opt/OV/shared/nnm/actions`
	• NNMi provides a set of parameters that can pass attribute values from an incident configuration. See "Valid Parameters for Configuring Incident Actions (Management Events)" (on page 859) for more information.

Configure a Payload Filter for an Action (Remote NNM 6.x/7.x Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline

as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

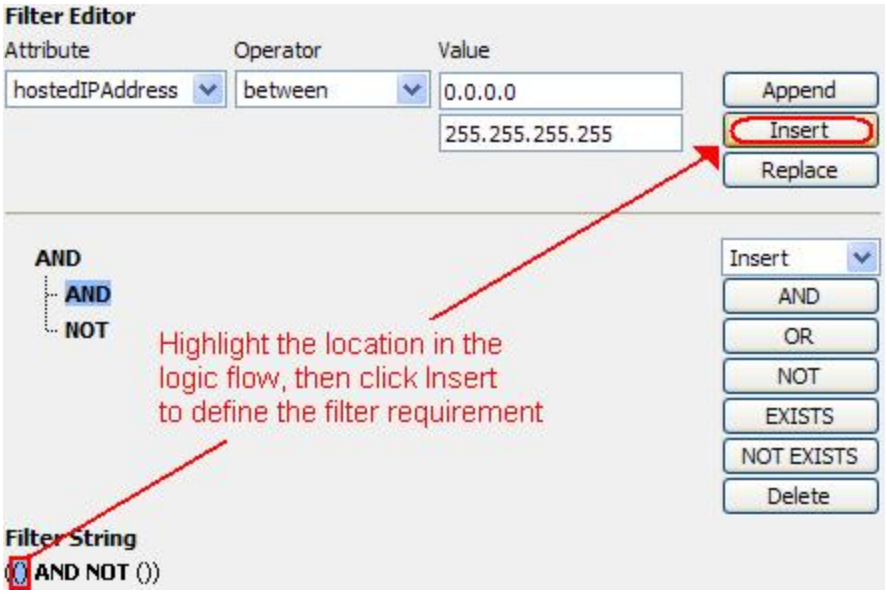
1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Remote NNM 6.x/7.x Events** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Actions** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.
4. Select the **Payload Filter** tab.
5. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".
    a. Plan out the logic needed for your Filter String.
    b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

        For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

        ```
        (( ) AND NOT ( ))
        ```

    c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



6. Click **Save and Close**.
7. Click **Save and Close** to save your changes and return to the previous form.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind value less than **6** . • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind value less than or equal to **6** . • **>** Finds all values greater than the value specified. Click here for an example.

Attribute	Description

Example: `ciaValue > 4` matches any incident that contains a varbind value greater than **4** .

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4** .

- **between** Finds all traps or events that include a varbind value equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4** .

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident that contains a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with no varbind values.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in

Attribute	Description
	\Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like *Chicago*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in` matches any incident that contains a varbind with values other than **1** and **2** . **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **`.1.3.6.1.4.1.9.9`**. `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Valid Parameters for Configuring Incident Actions (Remote NNM 6.x/7.x Events)

When configuring incident actions, you may want to use incident information as part of the action. NNMi provides the following parameter values. Use these parameters as variables in your Jython methods or executable files.

Tip: See the Using the Incident Form for more information about the parameter values.

Note: NNMi stores varbind values as custom incident attributes (CIAs).

See "Lifecycle Transition Action Form" (on page 415) for more information about configuring incident actions.

Valid Parameters Visible From an Incident's Form

Parameter Value	Description
$category, $cat	Value of the Category attribute in the Incident form.
$count, $cnt	Value representing the number of Custom Incident Attributes that appear in the Incident form.
$family, $fam	Value from the Family attribute in the Incident form.
$firstOccurrenceTime, $fot	Value from the First Occurrence Time attribute in the incident form.
$lastOccurrenceTime, $lot	Value from the Last Occurrence Time attribute in the incident form.
$lifecycleState, $lcs	Value from the Lifecycle State attribute in the Incident form.
$name	Value of the Name attribute from the incident configuration.
$nature, $nat	Value from the Nature attribute in the Incident form.
$origin, $ori	Value from the Origin attribute in the Incident form.
$originOccurrenceTime, $oot	Value from the Origin Occurrence Time attribute in the incident form.
$priority, $pri	Value from the Priority attribute in the Incident form.
$severity, $sev	Value of the Severity attribute of the Incident form.

Valid Parameters Visible from a Node Form

Parameter Value	Description
$managementAddress, $mga	Value from the Management Address attribute of the incident's source Node's form or SNMP Agent form.
$otherSideOfConnectionManagementAddress, $oma	If the incident's Source Node is part of a Layer 2 connection, this attribute is the value of the Management Address of a node on the other side of the Layer 2 connection.
$sourceNodeLongName, $sln	The fully-qualified DNS name as displayed in the Hostname attribute of the incident's source Node's form.
$sourceNodeName, $snn	Value from the Name attribute of the incident's source Node's form.
$sysContact, $sct	Value from the System Contact attribute of the incident's source Node form: General tab.
$sysLocation, $slc	Value from the System Location attribute of the incident's source Node form: General tab.

Valid Parameters Visible from an Interface Form

Parameter Value	Description
$ifAlias, $ifa	Value from the IfAlias attribute for the interface that is the incident's source object.
$ifConfigDupSetting, $icd	Configured Duplex Setting on the port associated with the interface that is the incident's source object.
$ifDesc, $idc	Value from the ifDesc attribute for the interface that is the incident's source object.
$ifIndex, $idx	Value from the ifIndex attribute for the interface that is the incident's source object.
$ifIpAddr, $iia	IP Address values associated with the interface that is the incident's source object. If multiple IPaddresses are associated with the interface, this parameter returns a comma-separated list.
$ifName, $ifn	Value from the ifName attribute for the interface that is the incident's source object.
$ifPhysAddr, $ipa	Value from the Physical Address attribute for the interface that is the incident's source object.
$ifSpeed, $isp	Value from the ifSpeed attribute for the interface that is the incident's souce object.
$ifType, $itp	Value from the ifType attribute for the interface that is the incident's souce object.

Valid Parameters Visible from a Layer 2 Connection Form

Parameter Value	Description
$otherSideOfConnectionConfigDupSetting, $ocd	If the incident's source Node is part of a Layer 2 connection, this parameter contains the Configured Duplex Setting on the port associated with the interface on the other side of the connection.
$otherSideOfConnectionIfAlias, $oia	If the incident's Source Node is part of a Layer 2 connection, this parameter is the value of the ifAlias of one of the interfaces on the other side of the Layer 2 connection.
$otherSideOfConnectionIfDesc, $odc	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifDescr attribute value for the interface on the other side of the Layer 2 connection.
$otherSideOfConnectionIfIndex, $odx	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifIndex attribute value for the interface on the other side of the connection.
$otherSideOfConnectionIfName, $ofn	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifName attribute value for the interface on the other side of the connection.

Valid Parameters Visible from a VLAN Form

Parameter Value	Description
$impVlanIds, $ivi	Value from the VLAN Id attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Port tab and open the form for the VLAN of interest. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.
$impVlanNames, $ivn	Value from the VLAN Name attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Ports tab of the Interface form. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.

Valid Parameters Not Visible From a Form

Parameter Value	Description
$id	Unique Object Identifier attribute value for the incident (unique across the entire NNMi Database).
$firstOccurrenceTimeMS, $fms	Value from the First Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$lastOccurrenceTimeMS, $lms	Value from the Last Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$oid	Value of the unique object identifier (oid) for the incident configuration that originated from either an SNMP trap, Management Event, or Remote NNM 6.x or 7.x event.
$otherSideOfConnection, $osc	If the incident's Source Node is part of a Layer 2 connection, this attribute is the following combination of values for the node and one of its interfaces on the other side of the Layer 2 connection: The fully-qualified DNS name of the node appended with the interface Name in the following format: <*fully-qualified DNS name*>[*interface_name*]
$originOccurrenceTimeMS, $oms	Value from the Origin Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$sourceNodeUuid, $snu	Universally Unique Object Identifier attribute value of the source node object for the incident (unique across all databases). This identifier distinguishes the source node object instance from all other node objects.
$sourceObjectClass, $soc	Value of the object class for the object you want to include. Use this parameter when you need to request more details of a class of objects through a web service. Examples of object classes include: `com.hp.ov.nms.model.core.Interface` and `com.hp.ov.nms.model.snmp.SnmpAgent`.
$sourceObjectName, $son	Value from the Name attribute of the source object. For example, an interface object is named according to the MIB ifName. Each ifName varies according to the vendor's conventions. Using the name `4/1` as an

Parameter Value	Description
	example, `4` represents `the board number and 1` represents the port number.
$sourceObjectUuid, $sou	Universally Unique Object Identifier attribute value of the source object for the incident (unique across all databases). This identifier distinguishes the source object instance from all other similar object instances..
$uuid	Universally Unique Object Identifier attribute value of the incident (unique across all databases). This identifier distinguishes the incident object instance from all other incident objects.

Valid Parameters Established in Custom Incident Attributes

Parameter Value	Description
$<position_ number>	Value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1` NNMi stores varbind values as Custom Incident Attributes. If you know the varbind position number, use this parameter.
$<CIA_ name>	Value of the name that is used for the custom incident attribute. For example, `$mycompany.mycia`. NNMi provides CIA values for configuring Management Events. See Custom Incident Attributes Provided by NNMi for more information about custom incident attributes.
$<CIA_ oid>	Value of the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this parameter when you are not certain of a custom incident attribute (varbind) position number.
$*	Used to indicate you want all of the custom incident attribute values originating as varbinds, to be passed to the action configuration. Each varbind is returned in the following format: `$<CIA_name>:<CIA_value>` in which the custom incident attribute name appears followed by the custom incident attribute value.

The function described in the following table replaces the specified numeric value with the associated text value stored in the CIA.

Note: The associated MIB must have been loaded using the nnmloadmib.ovpl command.

Functions to Generate Values Within Incident Messages

Function	Description
$text($<position_ number>)	The *<position_number>* argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1`. After the function runs, NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.

Function	Description
$text($<CIA_ oid>)	The <CIA_oid> argument specifies the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this argument to the $text function when you are not certain of a custom incident attribute (varbind) position number. After the function runs, NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.

Configure Management Events

Management events are those events that are generated from the NNMi Causal Engine. You can configure how you want these events to be displayed in the incident views provided by NNMi. The types of things you configure include its name, category, and the format of its message.

To configure a management event:

1. Navigate to the **Incident Configuration** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
2. Select the **Management Events** tab.
3. Do one of the following:
    a. To create a management event configuration, click the New icon, and continue.
    b. To edit a management event configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
    c. To delete a management event configuration, select a row, and click the Delete icon.
4. In the Management Event Configuration form, provide the required information.
5. Click **Save and Close** to save your changes and return to the **Incident Configuration** form.

The next time that a management event of this type arrives into the database, NNMi creates an associated incident to display in the appropriate console incident views.

Management Event Form

To configure incidents originating from management events:

1. Navigate to the **Incident Configuration** form:
    a. From the workspace navigation pane, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
2. Select the **Management Events** tab.
3. Make your configuration choices (see table).

Note: If you want to add or edit a management event configuration, verify that **Enabled** ☑ is selected.

a. To add a management event configuration, click the New icon, and continue.

b. To edit a management event configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

c. To delete a management event configuration, click the Delete icon.

4. Click **Save and Close** to save your changes and return to the **Incident Configuration** form.

Tasks for Management Event Configuration

Task	How
"Specify the Incident Configuration Name (Management Events)" (on page 732)	Use the **Basics** group of the **Management Event Configuration** form. Specify a name that helps you to identify the configuration for subsequent use.
Specify whether you want to enable this configuration.	In the **Basics** group of the **Management Event Configuration** form, verify that **Enable** ☑ is selected for each configuration you want to use.
"Specify Category and Family Attribute Values for Organizing Your Incidents (Management Events)" (on page 733)	Use the **Basics** group of the **Management Event Configuration** form. You can organize your incidents using Category and Family.
"Specify the Incident Severity (Management Events)" (on page 735)	Use the **Basics** group of the **Management Event Configuration** form. Possible Severity values include: **Normal, Warning, Minor, Major,** and **Critical**.
"Specify Your Incident Message Format (Management Events)" (on page 736)	Use the **Basics** group of the **Management Event Configuration** form. The message format determines the message to be displayed for the incident.
"Specify a Description for Your Incident Configuration (Management Events)" (on page 742)	Use the **Basics** group of the **Management Event Configuration** form. Provide a meaningful description.
Specify an Author for Your Incident Configuration (Management Events)	Use the **Basics** pane of the **Management Event Configuration** form to indicate who created or last modified the event. See Author form for important information. **Caution**: If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. Click the Lookup icon and select Quick Find to access the list of existing Author values or click New to create one.

After you complete the Basic Configuration for the management event, you can also choose to configure the information described in the following table.

Additional Configurations

Task	How
"Correlate Duplicate Incidents (Deduplication Configuration)" (on page 346)	Select the **Deduplication** tab to specify duplicate incidents that you want to be suppressed.
"Track Incident Frequency (Rate: Time Period and Count)" (on page 347)	Select the **Rate** tab to specify a rate for duplicate incidents. After the rate limit is reached, NNMi generates an Incident to notify you of the problem.
"Configure an Action for an Incident" (on page 414)	Select the **Actions** tab to specify actions that should occur automatically when an incident changes its Lifecycle State.
"Configure Diagnostics for an Incident (NNM iSPI NET)" (on page 421)	Select the **Configuration Per Node Group** tab to specify diagnostic actions that should occur automatically when an incident reaches a selected Lifecycle State for a node that belongs to a particular Node Group.

Configure Basic Settings for a Management Event Incident

The Basics settings for a Remote NNM 6.x/7.x event incident specifies general information for an incident configuration, including the name, severity, and message.

Note: In the **Basics** group of the **Management Event Configuration** form, verify that **Enable** is selected for each configuration you want to use.

For information about each Management Events tab:

To configure Basic settings for a Management Event incident:

Navigate to the **Management Event Configuration** form:

1. From the workspace navigation panel, select the **Configuration** workspace.
2. Select **Incident Configuration**.
3. Select the **Management Events** tab .
4. Do one of the following:
   a. To create an incident configuration, click the New icon, and continue.
   b. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   c. To delete an incident configuration, click the Delete icon.
5. Configure the required Basic settings (see the Basic Attributes table).
6. Click **Save and Close** to save your changes and return to the Incident Configuration form. NNMi uses the SNMP Object ID to enable forwarding of Management Events as SNMP traps. NNMi automatically assigns a unique SNMP Object ID to all Management Events provided by NNMi.

Basic Attributes for SNMP Trap Configuration

Task	How
"Specify the Incident Configuration Name	Use the **Basics** pane of the **Management Event Configuration** form. Specify a name that helps you to identify the configuration for subsequent use.

Task	How
(Management Events)" (on page 732)	
SNMP Object ID	The SNMP Object ID assigned by NNMi. Note the following about the SNMP Object ID that appears in the Basics settings of the Management Event Configuration form: • The Management Event SNMP Object ID is used when sending Management Events to another application. For example, you might want to send NNMi Management Event to an event consolidator such as HP Operations Manager. The SNMP Object ID is used to uniquely identify the management event in the application receiving the event. • NNMi assigns a unique SNMP Object ID to each Management Event configuration it provides. If you choose to create a new Management Event configuration, NNMi assigns the following "generic" SNMP Object ID to these user-created configurations: `.1.3.6.1.4.1.11.2.17.19.2.0.9999` • For user-defined Management Events, a combination of the SNMP Object ID and the user-defined event name must be used to uniquely identify the Management Event in an application receiving the event. • For more information, see the *HP Network Node Manager i Software Deployment Reference*, which is available at: http://h20230.www2.hp.com/selfsolve/manuals. • If you choose to create a new Management Event configuration, NNMi automatically assigns the same "generic" SNMP Object ID to all new Management Event configurations.
Specify whether you want to enable this configuration.	In the **Basics** group of the **Management Event Configuration** form, verify that **Enable** ☑ is selected for each configuration you want to use.
"Specify Category and Family Attribute Values for Organizing Your Incidents (Management Events)" (on page 733)	Use the **Basics** pane of the **Management Event Configuration** form. You can organize your incidents using Category and Family.
"Specify the Incident Severity (Management Events)" (on page 735)	Use the **Basics** pane of the **Management Event Configuration** form. Possible Severity values include: **Normal, Warning, Minor, Major,** and **Critical**.
"Specify Your Incident Message Format (Management Events)" (on page 736)	Use the **Basics** pane of the **Management Event Configuration**form. The message format determines the message to be displayed for the incident.
"Specify a Description for Your Incident Configuration (Management Events)" (on page 742)	Use the **Basics** pane of the **Management Event Configuration** form. Provide a meaningful description.

Task	How
Specify an Author for Your Incident Configuration (Management Events)	Use the **Basics** pane of the **Management Event Configuration** form to indicate who created or last modified the event. See Author form for important information. **Caution**: If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. Click the Lookup icon and select Quick Find to access the list of existing Author values or click New to create one.

After you complete the Basic Configuration for the remote NNM 6/x.7.x event, you can also choose to configure the information described in the following table.

Additional Incident Configurations

Task	How
"Configure Interface Settings for a Management Event Incident" (on page 743)	Select the **Interface Settings** tab to specify an Interface Group to which you want your incident configuration to apply.
"Configure Node Settings for a Management Event Incident" (on page 778)	Select the **Node Settings** tab to specify a Node Group to which you want your incident configuration to apply.
"Configure Suppression Settings for a Management Event Incident" (on page 815)	Select the **Suppression** tab to specify the criteria for discarding incidents that match the selected incident configuration.
"Configure Enrichment Settings for a Management Event Incident" (on page 826)	Select the **Enrichment** tab to specify enhancements for the selected incident configuration.
"Configure Dampening Settings for a Management Event Incident" (on page 829)	Select the **Dampen** tab to specify the time interval that must be met before the incident appears in an Incident view.
"Configure Deduplication for a Management Event Incident" (on page 841)	Select the **Deduplication** tab to specify duplicate incidents that you want to be suppressed.
"Configure Rate (Time Period and Count) for a Management Event Incident" (on page 846)	Select the **Rate** tab to specify a rate for duplicate incidents. After the rate limit is reached, NNMi generates an Incident to notify you of the problem.
"Configure Actions for a Management Event Incident" (on page 850)	Select the **Actions** tab to specify actions that should occur automatically when an incident changes its Lifecycle State.

Specify the Incident Configuration Name (Management Events)

When providing the Name for an incident configuration, use the following guidelines:

Name

The name is used to identify the incident configuration and must be unique. Use a name that will help you to remember the purpose or kind of management event, remote NNM 6.x or 7.x event or SNMP trap, for which you are configuring an incident. Name is also used to identify your Pairwise configurations.

Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. No spaces are allowed.

Specify Category and Family Attribute Values for Organizing Your Incidents (Management Events)

When configuring incidents, NNMi provides the Category and Family attributes to help you organize your incidents.

Preconfigured Categories
The Category attribute helps you organize your incidents. Select the category that you want to be associated with this type of incident when it appears in an incident view. Each of the possible Category values is described in the following table.

Incident Categories Provided by NNMi

Category	Description
Accounting	Used to indicate problems related to usage statistics and allocation of costs associated with the billing of time and services provided by devices. This category is not used by NNMi with default configurations, but it is available for incidents you define.
Application Status	Indicates there is a problem with the health of the NNMi software. Examples of these kinds of events include license expiration (see "Extend a Licensed Capacity" (on page 1020)) or that a certain NNMi process or service lost connection to the Process Status Manager (see "Stop or Start an NNMi Process" (on page 45) and "Stop or Start NNMi Services" (on page 49)).
Configuration	Indicates there is a problem with the configuration of a managed device. For example, there is a physical address mismatch.
Fault	Indicates a problem with the network, for example Node Down.
Performance	Indicates a threshold has been exceeded. For example, a utility has exceeded 90 percent.
Security	Indicates there is a problem related to authentication. For example, an SNMP authentication failure.
Status	Indicates some kind of status message. Examples of these kinds of incidents include "SNMP Link Up" or an "HSRP Group status Normal" message.

Note: You can add your own Category entries to NNMi. See "Create an Incident Category (Management Events)" (on page 733) for more information.

Note: You can add your own Family entries to NNMi. See "Create an Incident Family (Management Events)" (on page 734) for more information.

Create an Incident Category (Management Events)

The Category attribute helps you organize your incidents. Create any Category that makes sense to you and your team. For a list of the Category codes provided by NNMi, "Specify Category and Family Attribute

Values for Organizing Your Incidents (Management Events)" (on page 733).

To create a new incident Category:

1. Navigate to the **Management Events Category** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **Management Events** tab.
   d. Do one of the following:
      - To create an incident configuration, click the New icon, and continue.
      - To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      - To delete an incident configuration, click the Delete icon.
   e. In the configuration form, locate the **Category** attribute.
   f. Click the Lookup icon, and select New.
2. Provide the required information (see table).
3. Click **Save and Close** to save your changes and return to the previous form.

Category Code Attributes

Name	Description
Label	Incident category name. For example, Hardware Faults, or Cisco Error. Maximum size is 255 characters. Alpha-numeric, spaces, and underline characters are allowed.

Unique Key	**Caution**: After you click **Save and Close**, this value cannot be changed.
	Used as a unique identifier when exporting and importing configuration definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the label value as part of the unique key as shown in the following examples:
	`com.<your_company_name>.nnm.trapConf.category.<category_label>`
	`com.<your_company_name>.nnm.eventConf.category.<category_label>`
	`com.<your_company_name>.nnm.inciConf.category.<category_label>`
	The maximum length is 80 characters. Alpha-numeric characters and periods are allowed. Spaces are not allowed.

Create an Incident Family (Management Events)

The Family attribute helps you organize your incidents. Create any Family that makes sense to you and your team. For a list of the Family codes provided by NNMi, "Specify Category and Family Attribute Values for Organizing Your Incidents (Management Events)" (on page 733).

To create a new incident Family:

1. Navigate to the **Incident Family** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **Management Events** tab.
   d. Do one of the following:
      - To create an incident configuration, click the New icon, and continue.
      - To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      - To delete an incident configuration, click the Delete icon.
   e. In the configuration form, locate the **Family** attribute.
   f. Click the Lookup icon, and select New.
2. Provide the required information (see table).
3. Click **Save and Close** to save your changes and return to the previous form.

Family Attributes

Name	Description
Label	Family name. For example, Hardware Faults, or Cisco Error. Maximum size is 255 characters. Any character type is valid.

Unique Key	**Caution**: After you click **Save and Close**, this value cannot be changed.
	Used as a unique identifier when exporting and importing configuration definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the label value as part of the unique key as shown in the following examples:
	`com.<your_company_name>.nnm.trapConf.family.<family_label>`
	`com.<your_company_name>.nnm.eventConf.family.<family_label>`
	`com.<your_company_name>.nnm.inciConf.family.<family_label>`
	The maximum length is 80 alpha-numeric characters, periods allowed, no spaces allowed.

Specify the Incident Severity (Management Events)

The incident severity represents the seriousness calculated for the incident. Use the severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described in the following table.

Incident Severity Values

Attribute	Description
Normal	Indicates there are no known problems related to the associated object. This severity is meant to be informational. Generally, no action is needed for these incidents.
Warning	Indicates there may or may not be a problem related to the associated object.

Attribute	Description
Minor	Indicates NNMi has detected problems related to the associated object that require further investigation.
Major	Indicates NNMi has detected problems related to the associated object to be resolved before they become critical.
Critical	Indicates NNMi has detected problems related to the associated object that require immediate attention.

See "Monitor Incidents for Problems" for more information about these severity values.

Specify Your Incident Message Format (Management Events)

When configuring an incident, specify the information you want NNMi to include in the incident's Message attribute value. You can use any combination of valid parameter strings and Custom Incident attributes to configure the Message.

Note: The incident Message limit is 1024 characters. If the returned values exceed this limit, NNMi truncates the value starting from the end of the returned text string.

"Valid Parameters for Configuring Incident Messages (Management Events)" (on page 736)

"Include Custom Incident Attributes in Your Message Format (Management Events)" (on page 741)

Valid Parameters for Configuring Incident Messages (Management Events)

When configuring incident messages, you may want to use incident information as part of the message. NNMi provides the following parameter values. Use these parameters as variables when formatting an incident message.

Tip: See the Using the Incident Form for more information about the parameter values.

Note: NNMi stores varbind values as custom incident attributes (CIAs).

See "Specify Your Incident Message Format (Management Events)" (on page 736) for more information about configuring messages.

Parameter strings are available for the following:

- Incident form attributes (Click here for a list of choices.)

Parameter Strings for Attributes from an Incident Form

Parameter String	Description
$category, $cat	Value of the Category attribute in the Incident form.
$count, $cnt	Value representing the number of Custom Incident Attributes that appear in the Incident form.
$family, $fam	Value from the Family attribute in the Incident form.
$firstOccurrenceTime, $fot	Value from the First Occurrence Time attribute in the incident form.
$lastOccurrenceTime, $lot	Value from the Last Occurrence Time attribute in the incident form.
$lifecycleState, $lcs	Value from the Lifecycle State attribute in the Incident form.
$name	Value of the Name attribute from the incident configuration.
$nature, $nat	Value from the Nature attribute in the Incident form.
$origin, $ori	Value from the Origin attribute in the Incident form.
$originOccurrenceTime, $oot	Value from the Origin Occurrence Time attribute in the incident form.
$priority, $pri	Value from the Priority attribute in the Incident form.
$sev, $severity	Value of the Severity attribute of the Incident form.

- Node form attributes (Click here for a list of choices.)

Parameter Strings for Attributes from a Node Form

Parameter String	Description
$managementAddress, $mga	Value from the Management Address attribute of the incident's source Node's form or SNMP Agent form.
$otherSideOfConnectionManagementAddress, $oma	If the incident's Source Node is part of a Layer 2 connection, this attribute is the value of the Management Address of a node on the other side of the Layer 2 connection.
$sourceNodeLongName, $sln	The fully-qualified DNS name as displayed in the Hostname attribute of the incident's source Node's form.
$sourceNodeName, $snn	Value from the Name attribute of the incident's source Node's form.
$sysContact, $sct	Value from the System Contact attribute of the incident's source Node form: General tab.
$sysLocation, $slc	Value from the System Location attribute of the incident's source Node form: General tab.

- Interface form attributes (Click here for a list of choices.)

Parameter Strings for Attributes from an Interface Form

Parameter String	Description
$ifAlias, $ifa	Value from the IfAlias attribute for the interface that is the incident's source object.
$ifConfigDupSetting, $icd	Configured Duplex Setting on the port associated with the interface that is the incident's source object.
$ifDesc, $idc	Value from the ifDesc attribute for the interface that is the incident's source object.
$ifIndex, $idx	Value from the ifIndex attribute for the interface that is the incident's source object.
$ifIpAddr, $iia	IP Address values associated with the interface that is the incident's source object. If multiple IPaddresses are associated with the interface, this parameter returns a comma-separated list.
$ifName, $ifn	Value from the ifName attribute for the interface that is the incident's source object.
$ifPhysAddr, $ipa	Value from the Physical Address attribute for the interface that is the incident's source object.
$ifSpeed, $isp	Value from the ifSpeed attribute for the interface that is the incident's souce object.
$ifType, $itp	Value from the ifType attribute for the interface that is the incident's souce object.

- Layer 2 Connection form attributes (Click here for a list of choices.)

Parameter Strings for Attributes from a Layer 2 Connection Form

Parameter String	Description
$otherSideOfConnectionConfigDupSetting, $ocd	If the incident's source Node is part of a Layer 2 connection, this parameter contains the Configured Duplex Setting on the port associated with the interface on the other side of the connection.
$otherSideOfConnectionIfAlias, $oia	If the incident's Source Node is part of a Layer 2 connection, this parameter is the value of the ifAlias of one of the interfaces on the other side of the Layer 2 connection.
$otherSideOfConnectionIfDesc, $odc	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifDescr attribute value for the interface on the other side of the Layer 2 connection.
$otherSideOfConnectionIfIndex, $odx	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifIndex attribute value for the interface on the other side of the connection.
$otherSideOfConnectionIfName, $ofn	If the incident's Source Node is part of a Layer 2

Parameter String	Description
	connection, this parameter contains the ifName attribute value for the interface on the other side of the connection.

- VLAN form attributes (Click here for a list of choices.)
  Parameter Strings for Attributes from a VLAN Form

Parameter String	Description
$impVlanIds, $ivi	Value from the VLAN Id attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Port tab and open the form for the VLAN of interest. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.
$impVlanNames, $ivn	Value from the VLAN Name attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Ports tab of the Interface form. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.

- Additional information that is not visible in any form (Click here for a list of choices.)
  Parameter Strings for Attributes not visible in any Form

Parameter String	Description
$id	Unique Object Identifier attribute value for the incident (unique across the entire NNMi Database).
$firstOccurrenceTimeMS, $fms	Value from the First Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$lastOccurrenceTimeMS, $lms	Value from the Last Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$oid	Value of the unique object identifier (oid) for the incident configuration that originated from either an SNMP trap, Management Event, or Remote NNM 6.x or 7.x event.
$otherSideOfConnection, $osc	If the incident's Source Node is part of a Layer 2 connection, this attribute is the following combination of values for the node and one of its interfaces on the other side of the Layer 2 connection: The fully-qualified DNS name of the node appended with the interface Name in the following format: <*fully-qualified DNS name*>[*interface_name*]
$originOccurrenceTimeMS, $oms	Value from the Origin Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$sourceNodeUuid, $snu	Universally Unique Object Identifier attribute value of the source node object for the incident (unique across all databases). This identifier distinguishes the source node object instance from all other node objects.

Parameter String	Description
$sourceObjectClass, $soc	Value of the object class for the object you want to include. Use this parameter when you need to request more details of a class of objects through a web service. Examples of object classes include: `com.hp.ov.nms.model.core.Interface` and `com.hp.ov.nms.model.snmp.SnmpAgent`.
$sourceObjectName, $son	Value from the Name attribute of the source object. For example, an interface object is named according to the MIB ifName. Each ifName varies according to the vendor's conventions. Using the name `4/1` as an example, `4` represents `the board number and 1` represents the port number.
$sourceObjectUuid, $sou	Universally Unique Object Identifier attribute value of the source object for the incident (unique across all databases). This identifier distinguishes the source object instance from all other similar object instances..
$uuid	Universally Unique Object Identifier attribute value of the incident (unique across all databases). This identifier distinguishes the incident object instance from all other incident objects.

- Information established in Custom Incident Attributes (Click here for a list of choices.)

Parameter Strings for Attributes Established in Custom Incident Attributes

Parameter String	Description
$<position_number>	Value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1` NNMi stores varbind values as Custom Incident Attributes. If you know the varbind position number, use this parameter.
$<CIA_name>	Value of the name that is used for the custom incident attribute. For example, `$mycompany.mycia`. NNMi provides CIA values for configuring Management Events. See Custom Incident Attributes Provided by NNMi for more information about custom incident attributes.
$<CIA_oid>	Value of the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this parameter when you are not certain of a custom incident attribute (varbind) position number.
$*	Used to indicate you want all of the custom incident attribute values originating as varbinds, to be passed to the action configuration. Each varbind is returned in the following format: `$<`*`CIA_name`*`>:<`*`CIA_value`*`>` in which the custom incident attribute name appears followed by the custom incident attribute value.

- Functions to generate values (Click here for a list of choices.)

  The function described in the following table replaces the specified numeric value with the associated text value stored in the CIA.

  Note: The associated MIB must have been loaded using the nnmloadmib.ovpl command.

Functions to Generate Values Within the Incident Message

Function	Description
$oidtext($<position_number>)	A <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, `$oidtext($2)`. **Note**: The position number you enter must represent a CIA that contains an Object Identifier (OID) value. NNMi returns the textual value of the OID for the CIA specified. Note the following: ■ If the MIB is not loaded, NNMi returns the numeric OID value. ■ If the OID has a MIB instance, the number representing the MIB instance is appended to the textual OID value.
$oidtext($<CIA_oid>)	The <*CIA_oid*> argument specifies the Object Identifier (OID) for any custom incident attribute that originated as a varbind. For example, `$oidtext($.1.3.6.1.6.3.1.1.5.1.)` Use this argument to the $oidtext() function when you are not certain of a custom incident attribute (varbind) position number. NNMi replaces the numeric value with the textual value of the OID you specify. Note the following: ■ If the MIB is not loaded, NNMi returns the numeric OID value. ■ If the OID has a MIB instance, the number representing the MIB instance is appended to the textual OID value.
$text($<position_number>)	The <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1`. NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.
$text($<CIA_oid>)	The <CIA_oid> argument specifies the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this argument to the $text function when you are not certain of a custom incident attribute (varbind) position number. NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.

Include Custom Incident Attributes in Your Message Format (Management Events)

NNMi includes two categories of CIAs:

- SNMP trap varbinds identified by the Abstract Syntax Notation value (ASN.1). NNMi turns varbinds into CIAs and maintains each varbind's position number. See "Load SNMP Trap Incident Configurations"

(on page 429).

- Custom attributes provided by NNMi. See "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312).

You can use CIAs in your message format to extend the amount of information presented. To determine which CIAs are available for any particular incident type, open an Incident view, locate the incident and open the Incident form. Navigate to the **Custom Attributes** tab. A complete list of available CIAs (for that incident type) appears in the table.

To include a CIA in your message format, type the dollar-sign character (`$`) plus any of the following:

- Varbind position number or asterisk (*) to include all varbind values
- Name of the CIA
- Object identifier (oid) of the CIA (useful when the varbind position number is not consistent among vendors)

Note: A single incident cannot include two CIAs with the same name. However, two incidents can contain CIAs having the same names and values.

The following table presents some example formats with the subsequent output.

Example Incident Message Formats

Example Message Format	Output in Incident View
Possible trouble with $3	`Possible trouble with` <varbind 3>
Possible trouble with $11	`Possible trouble with` <varbind 11>
Possible trouble with $77 (where the varbind position 77 does not exist)	`Possible trouble with <Invalid or unknown cia> 77`
Possible trouble with $*	`Possible trouble with` <cia1_name: cia_value>, <cia2_name; cia_value>,< cia*n*_name: cia_value>
Possible trouble with $3x	`Possible trouble with` <varbind 3>`x`
Possible trouble with $1.2.3.4.5	`Possible trouble with` <value of the CIA with oid of 1.2.3.4.5>
Possible trouble with $mycia.mycompany	`Possible trouble with` <value of the CIA with name of mycia.mycompany>

Tip: NNMi provides an error message when a CIA cannot be found. For example, if you enter an unavailable varbind position, name, or object identifier (oid), NNMi returns an "Invalid or unknown cia" error message.

Specify a Description for Your Incident Configuration (Management Events)

NNMi provides the Description attribute to help you further identify the current incident configuration.

Description

Use the description field to provide additional information that you would like to store about the current incident configuration. This description applies only to the configuration entry.

Type a maximum of 2048 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Configure Interface Settings for a Management Event Incident

Note: Interface Settings override any other Suppression, Enrichment, Dampen, or Actions settings for this incident, including those configured on the Node Settings tab.

NNMi enables you to apply an incident configuration to a Source Object based on the Source Object's participation in an Interface Group. If the Source Object is not a member of the Interface Group specified, the incident is neither displayed nor stored in the NNMi database

Tip: See "Create Interface Groups" (on page 197) for more information about Interface Groups.

For information about each Interface Settings tab:

For information about each Management Events tab:

To apply an incident configuration to a Source Object based on the Source Object's Interface Group:

1. Navigate to the **Management Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Management Events** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
4. Configure the desired Interface Settings (see table).
5. Configure any Suppression, Dampen, or Enrichment settings for this Interface Group.
6. Click **Save and Close** to save your changes and return to the previous form.

Interface Group Attributes

Name	Description
Interface Group	Click the Lookup icon and select Quick Find to select the Interface Group you want to use. See "Use the Quick Find Window" (on page 28) for more information about using Quick Find.
Ordering	Determines the priority order for those interfaces that appear in multiple Interface Groups. The lower the number, the higher the priority. For example, **1** is the highest priority. If an interface is in multiple Interface Groups and more than one of those Interface Groups have been

Name	Description
	specified in an incident configuration, only the incident configuration with the highest priority will be applied to the interface.
Enable	Use this attribute to temporarily disable an incident's configuration settings. To temporarily disable the Interface Group settings for the selected incident configuration, clear **Enable** ☐. To enable the Interface Group settings for the selected incident configuration, click **Enable** ☑.

Related Topics

"Configure Node Settings for a Management Event Incident" (on page 778)

Configure Incident Suppression Settings for an Interface Group (Management Events)

Note: Interface Settings override any other Suppression settings for this incident, including those from the Node Settings tab.

NNMi enables you to suppress a specified incident configuration based on the Source Object's participation in an Interface Group.

Note: You can also suppress the incident configuration based on the Source Node's participation in a Node Group. See "Configure Incident Suppression Settings for a Node Group (Management Events)" (on page 779) for more information.

Tip: See "Create Interface Groups" (on page 197) for more information about Interface Groups.

For information about each Interface Settings tab:

To suppress an incident configuration based on an Interface Group:

1. Navigate to the **Management Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Management Events** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit a configuration, click the Open icon in the row representing the configuration you want to edit.

4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Management Event Incident" (on page 743) for more information.
5. Select the **Suppression** tab.
6. Configure the desired Suppression behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

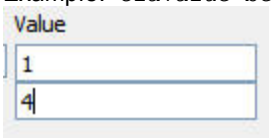
Interface Settings Suppression Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings for the specified Interface Group. To temporarily disable the Suppression settings for the selected incident configuration, clear **Enable** ☐. To enable the Suppression settings for the selected incident configuration, click **Enable** ☑.

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	
	When creating a Payload Filter, note the following:
	
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	
	The following example filters incidents on voltage state:
	
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`ciaValue = 5`
	
	NNMi evaluates the expression above as follows:
	
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Name	Description

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all values equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

  `ciaValue like .*Chicago.*` finds all traps or events that contain a

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like .*Chicago.*` finds all traps or events that do not

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	contain a varbind value that includes the string **Chicago**. contain a varbind value that includes the string **Chicago**. contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.
Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples. **Example 1**

Name	Description

Button	Description

```
AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
```

Name	Description

Button	Description
	ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Name Descr

Configure Incident Enrichment Settings for an Interface Group (Management Events)

Note: Interface Settings override any other Enrichment settings for this incident, including those from the Node Settings tab.

NNMi enables you to fine tune and enhance a specified incident configuration based on the Source Object's participation in an Interface Group. The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To

Note: You can also enhance the incident configuration based on the Source Node's participation in a Node Group. See "Configure Incident Enrichment Settings for a Node Group (Management Events)" (on page 787) for more information.

Tip: See Create Interface Groups for more information about Interface Groups.

For information about each Interface Settings tab:

For information about each Enrichment tab:

To enrich an incident configuration based on an Interface Group:

1. Navigate to the **Management Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Management Events** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.

4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Management Event Incident" (on page 743) for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the New icon, and continue.
   b. To edit an Enrichment configuration, click the Open icon, and continue.
   c. To delete an Enrichment configuration, click the Delete icon.
7. Configure the desired Enrichment behavior (see the "Enrich Configuration Attributes" table)
8. Click **Save and Close** to save your changes and return to the previous form.

Interface Settings Enrichment Configuration Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: • Accounting • Application Status • Configuration • Fault • Performance • Security • Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443) for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Possible values include: • Address • Aggregated Port • Board • Connection • Correlation • HSRP • Interface • Node • OSPF See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443) for more information.

Name	Description
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below: **Normal** - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there may or may not be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: **None** **Low** **Medium** **High** **Top** **Note**: The icons are displayed only in table views.
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact See Incident Form: General Tab for more information.

Name	Description
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right. You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (Management Events)" (on page 736) "Include Custom Incident Attributes in Your Message Format (Management Events)" (on page 741)
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 2048 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Configure Custom Incident Attributes to Enrich an Incident Configuration (Interface Settings) (Management Events)

The Custom Incident Attributes (CIAs) tab enables you to specify additional CIAs to be saved with an incoming incident. The persisted data might then be used as an argument to an action for the incident.

When creating a CIA for an incident configuration, you can specify any of the following values:

- Custom Attribute on the source node
- Custom Attribute on the interface (source object)
- Object Attribute (Object Attribute Path)

When specifying an Object Attribute, use the same syntax as when specifying a URL object for an Object-Type Attribute. See "Attributes per Object Type for Full URLs" (on page 883) for more information.

You also specify the Custom Incident Attribute name that will store this information.

For information about each Enrichment tab:

To create a Custom Incident Attribute to enrich an incident configuration:

1. Navigate to the **Management Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Management Events** tab.
    d. Do one of the following:

i. To create an incident configuration, click the New icon, and continue.

ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

iii. To delete an incident configuration, click the Delete icon.

2. Select **Interface Settings**.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Management Event Incident" (on page 743) for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the New icon, and continue.
   b. To edit an Enrichment configuration, click the Open icon, and continue.
   c. To delete an Enrichment configuration, click the Delete icon.
7. Make sure you configure the Enrichment settings. See "Configure Incident Enrichment Settings for an Interface Group (Management Events)" (on page 752) for more information.
8. Navigate to the **Custom Incident Attributes** tab.
9. Do one of the following:
   a. To create a Custom Incident Attribute, click the New icon, and continue.
   b. To edit a Custom Incident Attribute, click the Open icon, and continue.
   c. To delete a Custom Incident Attribute, click the Delete icon.
10. Configure the Custom Incident Attribute (see the "Custom Incident Attribute" table).
11. Click **Save and Close** to save your changes and return to the previous form.

Custom Incident Attribute

Name	Description
Custom Incident Attribute Name	Name used to identify the Custom Incident Attribute you are configuring. The name limit is 80 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. No spaces are allowed.
Type	Specifies whether you are using a Custom Attribute on a node, a Custom Attribute on an interface, or an Object Attribute. Possible values are: • Node Custom Attribute • Interface Custom Attribute • Object Attribute Path

Name	Description
Custom Attribute Name or Object Attribute Path	Used to determine the value to be assigned to the Custom Incident Attribute you are configuring. Enter either of the following: • Name of the Custom Attribute on the source node • Name of the Custom Attribute on the interface (source object) • Object Attribute Path When specifying an Object Attribute Path, use the same syntax as when specifying a URL object for an Object-Type Attribute; for example ${deviceProfile.SNMPObjectID}. See "Attributes per Object Type for Full URLs" (on page 883) for more information.

Configure a Payload Filter to Enrich an Incident Configuration (Interface Settings) (Management Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be enriched. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Management Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Management Events** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon that precedes the configuration you want to edit, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Management Event Incident" (on page 743) for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:

a. To create an Enrichment configuration, click the New icon, and continue.

b. To edit an Enrichment configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

c. To delete an Enrichment configuration, click the Delete icon.

7. Make sure you configure the Enrichment settings. See "Configure Incident Enrichment Settings for an Interface Group (Management Events)" (on page 752) for more information.
8. Select the **Payload Filter** tab.
9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

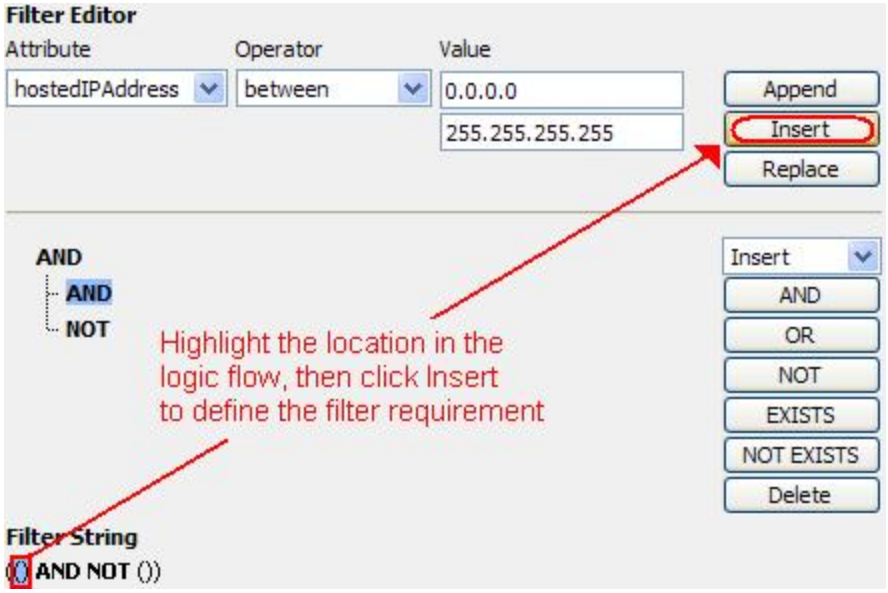
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



10. Click **Save and Close**.
11. Click **Save and Close** to save your changes and return to the previous form.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Operator	Valid operators are described below. ■ **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. ■ **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. ■ **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. ■ **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. ■ **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between` matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. ■ **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`

Attribute	Description

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not have a value.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Examples:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters.

  `ciaValue like *Chicago*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**.

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

Attribute	Description
	 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. ■ **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Value	The value for which you want NNMi to search. Note the following: ■ The values you enter are case sensitive. ■ NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. ■ The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Incident Dampening Settings for an Interface Group (Management Events)

Note: Interface Settings override any other Dampening settings for this incident, including those from the Node Settings tab.

NNMi enables you to delay the following for an incident configuration based on the Source Object's participation in an Interface Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

Note: You can also configure the Dampening settings based on the Source Node's participation in a Node Group. See "Configure Incident Dampening Settings for a Node Group (Management Events)" (on page 797) for more information.

Tip: See "Create Interface Groups" (on page 197) for more information about Interface Groups.

For information about each Interface Settings tab:

When using the Dampening configuration, note the following:

- By default, the Dampen Interval for NNMi Management Event Incidents is six minutes.

  Note: Any Dampen Interval set for less than six minutes might not provide enough time for the Causal Engine to complete its analysis. As a result, incidents associated with intermediate symptoms might appear as root cause incidents.

- NNMi initially assigns incidents with Dampening settings configured a Lifecycle State of DAMPENED.
- After the Dampen Interval has passed, NNMi changes the Lifecycle State to REGISTERED.

See About the Incident Lifecycle for more information about Lifecycle State.

To configure the Dampening settings based on an Interface Group:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **Management Events** tab.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete an incident configuration, click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Management Event Incident" (on page 743) for more information.
5. Select the **Dampening** tab.
6. Configure the desired Dampening behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Interface Settings Dampening Configuration Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's dampening settings. To temporarily disable the Dampening Configuration settings for the selected incident configuration, clear **Enable** ☐. To enable the Dampening Configuration settings for the selected incident configuration, click **Enable** ☑.
Hour	Specifies the number of hours to be used for the dampen interval.
Minutes	Specifies the number of minutes to be used for the dampen interval. **Note**: The maximum dampen interval is 60 minutes.
Seconds	Specifies the number of seconds to be used for the dampen interval.

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	
	When creating a Payload Filter, note the following:
	
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	
	The following example filters incidents on voltage state:
	
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`ciaValue = 5`
	
	NNMi evaluates the expression above as follows:
	
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Name	Description

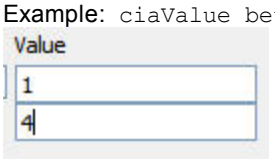
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all values equal to and between the two values specified. Click here for an example.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

Example: `ciaValue between`



matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line. selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line. selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.
Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples. **Example 1** `AND`

Name	Description

Button	Description

```
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Name	Description

Button	Description

	Example 2
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Example 2
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Example 2
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`

Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Name Desc

Configure Incident Actions for an Interface Group (Management Events)

Note: Interface Settings override any other Actions settings for this incident, including those from the Node Settings tab.

For information about each Interface Settings tab:

NNMi enables you to configure incident actions based on a Source Object's participation in an Interface Group.

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 (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

You can configure actions for incidents generated from SNMP traps, NNM 6.x or 7.x events, and the NNMi management events. Any time an incident configuration changes, the action directory is rescanned and any Jython files are reloaded to the NNMi database. See "Lifecycle Transition Action Form (Management Events)" (on page 851) for more information about the actions directory.

Tip: Copy any required Jython files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (Management Events)" (on page 851) for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. See "Verify that NNMi Services are Running" (on page 47) for more information about log files and where they are located.

To configure an automatic action for an incident:

1. Navigate to the **Management Event Configuration** tab.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration.**
    c. Select the **Management Events** tab.
    d. Do one of the following:
        i. To create a new incident configuration, click the New icon.
        ii. To edit an existing incident configuration, click the Open icon in the row representing the configuration you want to edit.
2. Select the **Interface Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Management Event Incident" (on page 743) for more information.
5. Select the **Actions** tab.

6. From the **Lifecycle Actions** table toolbar, do one of the following:
   - To create an Action configuration, click the New icon, and continue.
   - To edit an Action configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an Action configuration, click the Delete icon.
7. In the "Lifecycle Transition Action Form (Management Events)" (on page 851), provide the required information.
8. Click **Save and Close** to save your changes and return to the previous form.

   The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type.

Configure a Payload Filter for an Incident Action (Interface Settings) (Management Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **Management Events** tab.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete an incident configuration, click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Management Event Incident" (on page 743) for more information.
5. Select the **Actions** tab.
6. Do one of the following:

a. To create an Action configuration, click the New icon, and continue.

b. To edit an Action configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

c. To delete an Action configuration, click the Delete icon.

7. Make sure the Action settings are configured. See "Configure Incident Actions for an Interface Group (Management Events)" (on page 772) for more information.
8. Select the **Payload Filter** tab.
9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

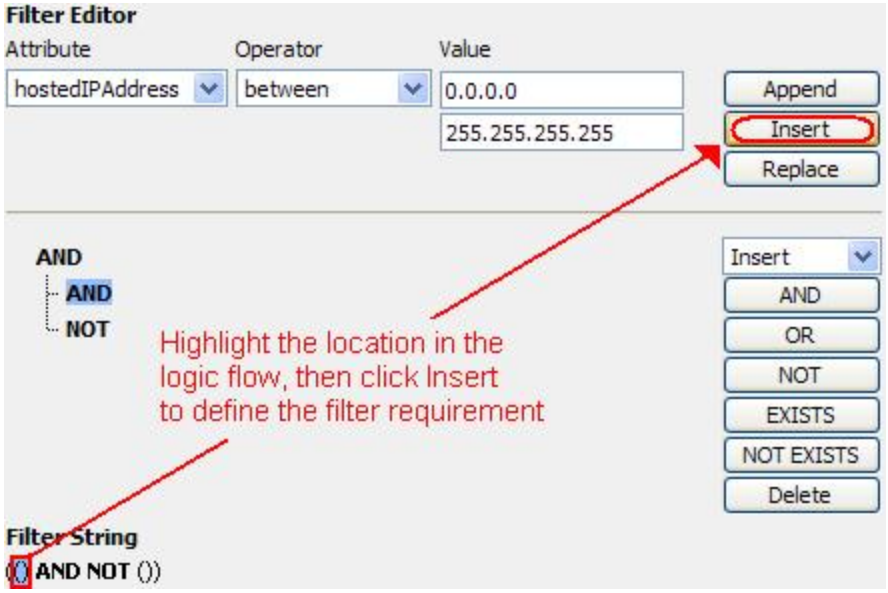
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



10. Click **Save and Close**.
11. Click **Save and Close** to save your changes and return to the previous form.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Operator	Valid operators are described below. ■ **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. ■ **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. ■ **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. ■ **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. ■ **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between` matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. ■ **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`

Attribute	Description

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not have a value.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Examples:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters.

  `ciaValue like *Chicago*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**.

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

Attribute	Description

matches any incident that contains a varbind with values other than **1** and **2**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**.

  `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Value	The value for which you want NNMi to search. Note the following: ■ The values you enter are case sensitive. ■ NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. ■ The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Node Settings for a Management Event Incident

Note: Node Settings override any other Suppression, Enrichment, Dampen, Action, or Diagnostics Selections configuration settings, except those configured on the Interface Settings tab.

NNMi enables you to apply an incident configuration to a Source Node based on the Source Node's participation in a Node Group. If the Source Node is not a member of the Node Group specified, the incident is neither displayed nor stored in the NNMi database.

Tip: See "Create Node Groups" (on page 181) for more information about Node Groups.

For information about each Node Settings tab:

For information about each Management Events tab:

To apply an incident configuration to a Source Node based on the Source Node's Node Group:

1. Navigate to the **Management Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Management Events** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.
4. Configure the desired Node Settings (see table).
5. Click **Save and Close** to save your changes and return to the previous form.

Node Group Attributes

Name	Description
Node Group	Click the Lookup icon and select Quick Find to select the Node Group you want to use. See "Use the Quick Find Window" (on page 28) for more information about using Quick Find.
Ordering	Determines the priority order for those nodes that appear in multiple Node Groups. The lower the number, the higher the priority. For example, **1** is the highest priority. If a node is in multiple Node Groups and more than one of those Node Groups have been specified in an incident configuration, only the incident configuration with the highest priority will be applied to the node.
Enable	Use this attribute to temporarily disable an incident's suppression settings. To temporarily disable the Node Group settings for the selected incident configuration, clear **Enable** ☐. To enable the Node Group settings for the selected incident configuration, click **Enable** ☑.

Configure Incident Suppression Settings for a Node Group (Management Events)

Note: Node Settings override any other Suppression settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to suppress a specified incident configuration based on the Source Node's participation in a Node Group.

Note: You can also suppress the incident configuration based on the Source Object's participation in an Interface Group. See "Configure Incident Suppression Settings for an Interface Group (Management Events)" (on page 744) for more information.

Tip: See "Create Node Groups" (on page 181)for more information about Node Groups.

For information about each Node Settings tab:

To suppress an incident configuration based on a Node Group:

1. Navigate to the **Management Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Management Events** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Node Settings** tab.
3. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Management Event Incident" (on page 778) for more information.
5. Select the **Suppression** tab.
6. Configure the desired Suppression behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Suppression Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings. To temporarily disable the Suppression settings for the selected incident configuration, clear **Enable**☐. To enable the Suppression settings for the selected incident configuration, click **Enable**☑.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor. When creating a Payload Filter, note the following: • Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class). • You must use a `ciaName` that already exists in the trap or event you are configuring. • Each set of expressions associated with a Boolean Operator is treated as if it were

Name	Description

enclosed in parentheses and evaluated together.

- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.

  The following example filters incidents on voltage state:

  ```
  AND
      ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
      ciaValue = 5
  ```

  NNMi evaluates the expression above as follows:

  ```
  (ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)
  ```

  NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

- **>** Finds all values greater than the value specified. Click here for an example.

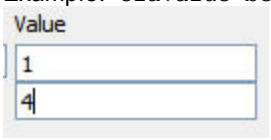
  Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

  `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.

Name	Description

Button	Description
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.

Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples.
	Example 1
	`AND`
	`ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6`
	`OR`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6`
	`OR`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3`
	Example 2
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`

Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Name	Description

Configure Incident Enrichment Settings for a Node Group (Management Events)

Note: Node Settings override any other Enrichment settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to enhanced a specified incident configuration based on the Source Node's participation in a Node Group. The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To

Note: You can also enhance the incident configuration based on the Source Object's participation in an Interface Group. See "Configure Incident Enrichment Settings for an Interface Group (Management Events)" (on page 752) for more information.

Tip: See "Create Node Groups" (on page 181) for more information about Node Groups.

For information about each Node Settings tab:

For information about each Enrichment tab:

To configure Enrichment settings for a Node Group:

1. Navigate to the **Management Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Management Events** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Management Event Incident" (on page 778) for more information.

5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the New icon, and continue.
   b. To edit an Enrichment configuration, click the Open icon, and continue.
   c. To delete an Enrichment configuration, click the Delete icon.
7. Configure the desired Enrichment behavior (see the "Enrich Configuration Attributes" table)
8. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Enrichment Configuration Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: • Accounting • Application Status • Configuration • Fault • Performance • Security • Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443) for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Possible values include: • Address • Aggregated Port • Board • Connection • Correlation • HSRP • Interface • Node • OSPF See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443) for more information.
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below:

Name	Description
	Normal - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there may or may not be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: 5 **None** 4 **Low** 3 **Medium** 2 **High** 1 **Top** **Note**: The icons are displayed only in table views.
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact See Incident Form: General Tab for more information.

Name	Description
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right. You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (Management Events)" (on page 736) "Include Custom Incident Attributes in Your Message Format (Management Events)" (on page 741)
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 2048 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Configure Custom Incident Attributes to Enrich an Incident Configuration (Node Settings) (Management Events)

The Custom Incident Attributes (CIAs) tab enables you to specify additional CIAs to be saved with an incoming incident. The persisted data might then be used as an argument to an action for the incident.

When creating a CIA for an incident configuration, you can specify any of the following values:

- Custom Attribute on the source node
- Custom Attribute on the interface (source object)
- Object Attribute (Object Attribute Path)

When specifying an Object Attribute, use the same syntax as when specifying a URL object for an Object-Type Attribute. See "Attributes per Object Type for Full URLs" (on page 883) for more information.

You also specify the Custom Incident Attribute name that will store this information.

For information about each Enrichment tab:

To create a Custom Incident Attribute to enrich an incident configuration:

1. Navigate to the **Management Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Management Events**.
    d. Do one of the following:

i. To create an incident configuration, click the New icon, and continue.

ii. To edit an incident configuration, click the Open that precedes the configuration you want to edit, and continue.

iii. To delete an incident configuration, click the Delete icon.

2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Management Event Incident" (on page 778) for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the New icon, and continue.
   b. To edit an Enrichment configuration, click the Open icon, and continue.
   c. To delete an Enrichment configuration, click the Delete icon.
7. Make sure the Enrichment settings are configure. See "Configure Incident Enrichment Settings for a Node Group (Management Events)" (on page 787) for more information.
8. Navigate to the **Custom Incident Attributes** tab.
9. Do one of the following:
   a. To create a Custom Incident Attribute, click the New icon, and continue.
   b. To edit a Custom Incident Attribute, click the Open icon, and continue.
   c. To delete a Custom Incident Attribute, click the Delete icon.
10. Configure the Custom Incident Attribute (see the "Custom Incident Attribute" table).
11. Click **Save and Close** to save your changes and return to the previous form.

Custom Incident Attribute

Name	Description
Custom Incident Attribute Name	Name used to identify the Custom Incident Attribute you are configuring. The name limit is 80 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. No spaces are allowed. **Note**: Make sure to note this name if you plan to filter on the value using the **Payload Filter** tab. See "Configure a Payload Filter to Enrich an Incident Configuration (Interface Settings) (Management Events)" (on page 757) for more information.
Type	Specifies whether you are using a Custom Attribute on a node, a Custom Attribute on an interface, or an Object Attribute. Possible values are: • Node Custom Attribute • Interface Custom Attribute

Name	Description
	• Object Attribute Path
Custom Attribute Name or Object Attribute Path	Used to determine the value to be assigned to the Custom Incident Attribute you are configuring. Enter either of the following: • Name of the Custom Attribute on the source node • Name of the Custom Attribute on the interface (source object) • Object Attribute Path When specifying an Object Attribute Path, use the same syntax as when specifying a URL object for an Object-Type Attribute; for example ${deviceProfile.SNMPObjectID}. See "Attributes per Object Type for Full URLs" (on page 883) for more information.

Configure a Payload Filter to Enrich an Incident Configuration (Node Settings) (Management Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be enriched. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Management Events Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Management Events** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Management Event Incident" (on page 778) for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:

a. To create an Enrichment configuration, click the New icon, and continue.

b. To edit an Enrichment configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

c. To delete an Enrichment configuration, click the Delete icon.

7. Make sure you configure the Enrichment settings. See "Configure Incident Enrichment Settings for a Node Group (Management Events)" (on page 787) for more information.
8. Select the **Payload Filter** tab.
9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

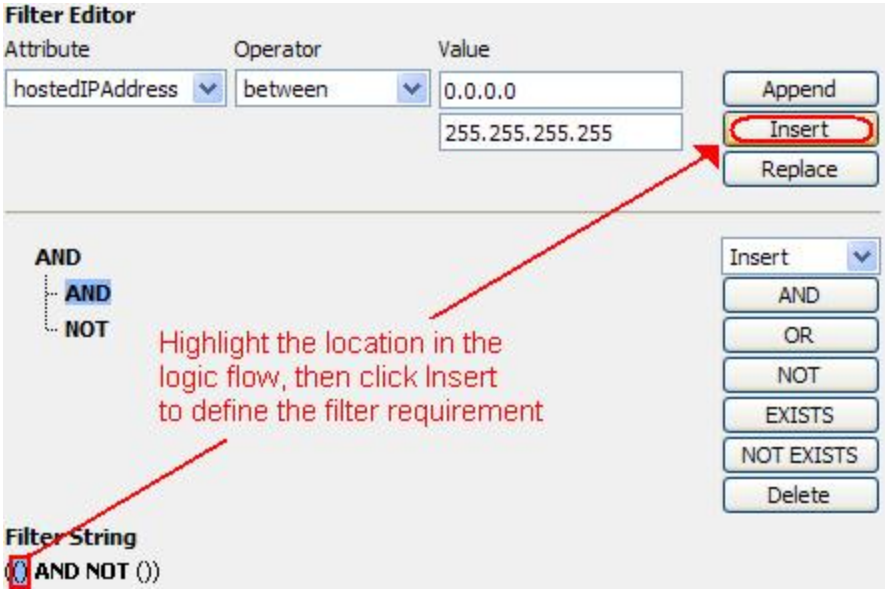
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



10. Click **Save and Close**.
11. Click **Save and Close** to save your changes and return to the previous form.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Operator	Valid operators are described below. ■ **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. ■ **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. ■ **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. ■ **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. ■ **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between` matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. ■ **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`

Attribute	Description

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not have a value.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Examples:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters.

  `ciaValue like *Chicago*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**.

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

Attribute	Description
	 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. ■ **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Value	The value for which you want NNMi to search. Note the following: ■ The values you enter are case sensitive. ■ NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. ■ The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Incident Dampening Settings for a Node Group (Management Events)

Note: Node Settings override any other Dampening settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to delay the following for an incident configuration based on the Source Node's participation in a Node Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

Note: You can configure the Dampening settings based on the Source Object's participation in an Interface Group. See "Configure Incident Dampening Settings for an Interface Group (Management Events)" (on page 762) for more information.

Tip: See "Create Node Groups" (on page 181) for more information about Node Groups.

For information about each Node Settings tab:

When using the Dampening configuration, note the following:

- By default, the dampen interval for NNMi Management Event Incidents is six minutes.

  Note: Any dampen interval set for less than six minutes might not provide enough time for the Causal Engine to complete its analysis. As a result, incidents associated with intermediate symptoms might appear as root cause incidents.
- NNMi initially assigns incidents with Dampening settings configured a Lifecycle State of DAMPENED.
- After the dampen interval has passed, NNMi changes the Lifecycle State to REGISTERED.

See About the Incident Lifecycle for more information about Lifecycle State.

To configure the Dampening settings based on a Node Group:

1. Navigate to the **Management Events Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **Management Events** tab.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete an incident configuration, click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Management Event Incident" (on page 778) for more information.
5. Select the **Dampen** tab.
6. Configure the desired Dampen behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Dampen Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's Dampening settings. To temporarily disable the Dampening settings for the selected incident configuration, clear **Enable** ☐. To enable the Dampening settings for the selected incident configuration, click **Enable** ☑.
Hour	Specifies the number of hours to be used for the dampen interval.
Minutes	Specifies the number of minutes to be used for the dampen interval. **Note**: Use a minimum of 6 minutes for the dampen interval. The maximum dampen interval is 60 minutes.
Seconds	Specifies the number of seconds to be used for the dampen interval.

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	
	When creating a Payload Filter, note the following:
	
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	
	The following example filters incidents on voltage state:
	
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`ciaValue = 5`
	
	NNMi evaluates the expression above as follows:
	
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Name	Description

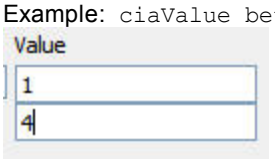
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all values equal to and between the two values specified. Click here for an example.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

Example: `ciaValue between`



matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line. selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line. selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.
Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples. **Example 1**

Name	Description

Button	Description

```
AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
```

Name	Description

Button	Description
	OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Name	Description

Configure Incident Actions for a Node Group (Management Events)

For information about each Node Settings tab:

Note: Node Settings override any other Actions settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to configure incident actions based on a Source Node's participation in a Node Group.

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 (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

You can configure actions for incidents generated from SNMP traps, NNM 6.x or 7.x events, and the NNMi management events. Any time an incident configuration changes, the action directory is rescanned and any Jython files are reloaded to the NNMi database. See "Lifecycle Transition Action Form (Management Events)" (on page 851) for more information about the actions directory.

Tip: Copy any required Jython files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (Management Events)" (on page 851) for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. See "Verify that NNMi Services are Running" (on page 47) for more information about log files and where they are located.

To configure an automatic action for an incident:

1. Navigate to the **Management Events Configuration** tab.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration.**
    c. Select the **Management Events**tab.
    d. Do one of the following:
        i. To create a new incident configuration, click the New icon.
        ii. To edit an existing incident configuration, click the Open icon in the row representing the configuration you want to edit.
2. Select the **Node Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Management Event Incident" (on page 778) for more information.
5. Select the **Actions** tab.

6. From the **Lifecycle Actions** table toolbar, do one of the following:
   - To create an Action configuration, click the New icon, and continue.
   - To edit an Action configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an Action configuration, click the Delete icon.
7. In the "Lifecycle Transition Action Form (Management Events)" (on page 851), provide the required information.
8. Click **Save and Close** to save your changes and return to the **Management Event Configuration** form.

   The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type.

Configure a Payload Filter for an Incident Action (Node Settings) (Management Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Management Events Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **Management Events** tab.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete an incident configuration, click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Management Event Incident" (on page 778) for more information.
5. Select the **Actions** tab.
6. Do one of the following:

a. To create an Action configuration, click the New icon, and continue.

b. To edit an Action configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

c. To delete an Action configuration, click the Delete icon.

7. Make sure the Action settings are configured. See "Configure Incident Actions for a Node Group (Management Events)" (on page 807) for more information.
8. Select the **Payload Filter** tab.
9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

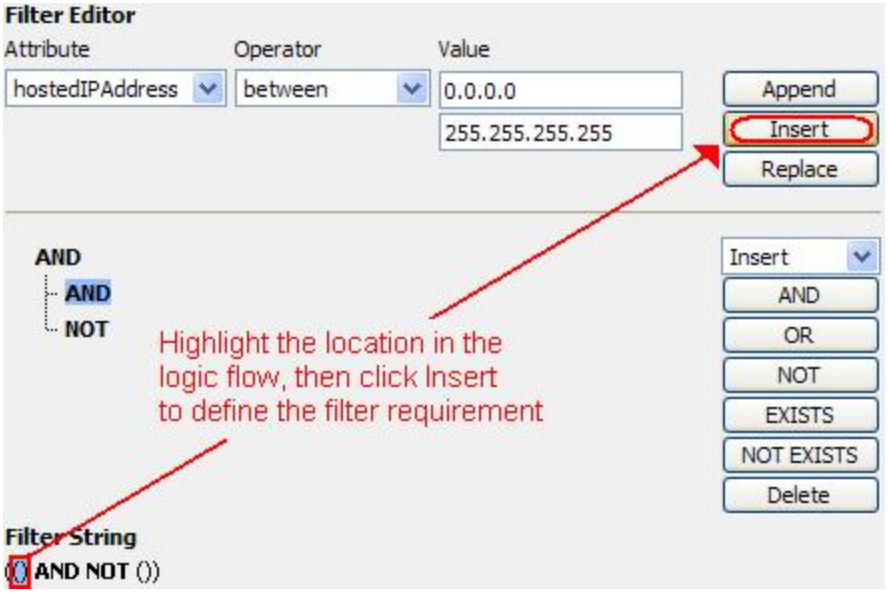
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



10. Click **Save and Close**.
11. Click **Save and Close** to save your changes and return to the previous form.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Operator	Valid operators are described below. ■ **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. ■ **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. ■ **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. ■ **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. ■ **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between` matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. ■ **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`

Attribute	Description

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not have a value.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Examples:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters.

  `ciaValue like *Chicago*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**.

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

Attribute	Description
	 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. ■ **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Value	The value for which you want NNMi to search. Note the following: ■ The values you enter are case sensitive. ■ NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. ■ The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Diagnostics Selections for a Node Group (Management Events)

For information about each Node Settings tab: .

Note: Node Settings override any other Diagnostics Selections settings for this incident, except those configured on the Interface Settings tab.

(*HP Network Node Manager iSPI Network Engineering Toolset Software*) The Diagnostic Selections form enables you to configure NNMi to automatically gather NNM iSPI NET diagnostic information for the

Incident you are configuring. When using this form, you specify the diagnostics you want to run on each applicable node in the specified Node Group.

To configure Diagnostics to run on a Source Node for an incident:

1. Navigate to the **Diagnostics Selection** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration.**
    c. Select the **Management Events** tab.
    d. Do one of the following:
        - To create an Incident configuration, click the New icon, and continue.
        - To edit an Incident configuration, select the Incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
    e. Navigate to **Node Settings** tab, and do one of the following:
        - To create a Node Settings configuration, click the New icon, and continue.
        - To edit a Node Settings configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        - To delete a Node Settings configuration, select the Node setting and click the Delete icon.
    f. Navigate to the **Diagnostic Selection** tab, and do one of the following:
        - To create a Diagnostic Selection setting, click the New icon, and continue.
        - To edit a Diagnostic Selection setting, click the Open icon in the row representing the Diagnostic Selection setting you want to edit, and continue.
        - To delete a Diagnostic Selection setting, select the Diagnostic Selection setting and click the Delete icon.
2. Provide the required information (see table).
3. Click **Save and Close** to save your changes and return to the previous form.

After you configure the Diagnostic for the incident and Node Group, the Diagnostic must match the following criteria before the Diagnostic runs:

- The Source Node must be in the specified Node Group.
- The Diagnostic must be valid for the Source Node. (For example, only Nortel switch Diagnostics are run on Nortel switches.)
- The incident's current lifecycle state must match a lifecycle state for which it was configured. (For example, if you configure the Incident to run a specified Diagnostic when the incident is Closed, then if the current Incident's Lifecycle State is Closed, NNMi runs that Diagnostic.)

Note: If a Source Node is in more than one Node Group, the Diagnostic is only run on the node the first time NNMi finds a match for that Source Node based on the configuration Ordering field.

If these criteria are met, NNM iSPI NET runs the Diagnostics and generates Diagnostic reports to help you solve the problem on the Source Node.

After you configure Diagnostics for an incident, you can also run Diagnostics and access the Diagnostics reports on demand, using **Actions → Run Diagnostics (iSPI NET only)** in the Incident form. The same criteria apply (see the criteria above). See Incident Form:Diagnostics Tab for more information.

You can also run and access Diagnostics reports from a Node form. See Node Form: Diagnostics Tab for more information.

Diagnostic Settings Attributes

Attribute	Description
Flow Definition	Select the Diagnostic (Flow Definition) you want to use for the specified Node Group. Click the Lookup icon and choose one of the following options: • Quick View to view summary information for the Flow Definition name displayed. • Quick Find to view the list of possible diagnostic Flow Definitions. NNMi provides diagnostics for the following types of devices: ▪ Cisco switch ▪ Cisco router ▪ Cisco switch/router ▪ Nortel switch See "Diagnostics (Flows) Provided by NNM iSPI NET" (on page 422) for more information about the diagnostics provided and the devices to which they apply.
Lifecycle State	Incident Lifecycle State of the target Incident. If the incident's Lifecycle State matches the value specified here, the Diagnostic runs. The Diagnostic automatically runs on each applicable Source Node in the specified Node Group if the incident has the Lifecycle State currently configured in this attribute of the Diagnostic (Flow Definition - set of automated commands). See About the Incident Lifecycle for more information about Lifecycle State.
Enable	Use this attribute to temporarily disable an incident's Diagnostics settings. To temporarily disable the selected Diagnostics settings, clear **Enable** ☐. To enable the selected Diagnostics settings, click **Enable** ☑.

Configure Suppression Settings for a Management Event Incident

For information about each Management Events tab:

NNMi enables you to suppress incidents based on Interface Group, Node Group, or default Suppression settings. NNMi applies your Suppression settings in the following order. Only the first match applies.

1. Interface Group (Management Event Configuration Form: Interface Settings tab)
2. Node Group (Management Event Configuration Form: Node Settings tab)
3. Suppression configuration settings without specifying an Interface Group or Node Group (Management Event Configuration Form: Suppresion tab)

A Payload Filter allows you to use the data that is included with any of the following items before they are stored as incidents in NNMi:

- Traps generated from an SNMP agent
- Management incidents that are generated by NNMi
- Events generated by NNM 6.x or 7.x management stations

Examples of the type of data that can be used as a Payload Filter include SNMP trap varbind names and values as well as CIA (Custom Incident Attribute) names and values. For example, you might want NNMi to suppress a particular status change notification trap for a specified Node Group or Interface Group. To do so, you could include the name of the trap varbind that stores this information as well as the particular status change value string the traps that you want to suppress should contain.

See "Configure Incident Suppression Settings for an Interface Group (Management Events)" (on page 744) for information about how to suppress an incident for an Interface Group with or without a Payload Filter.

See "Configure Incident Suppression Settings for a Node Group (Management Events)" (on page 779) for more information about how to suppress an incident for a Node Group with or without a Payload Filter.

To configure suppression for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **Management Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Management Events** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Suppression** tab.
3. Provide the required information (see table)
4. Click **Save and Close** to save your changes and return to the previous form.

Suppression Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings. To temporarily disable the Suppression settings for the selected incident configuration, clear **Enable** ☐. To enable the Suppression settings for the selected incident configuration, click **Enable** ☑.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor. When creating a Payload Filter, note the following:

Name	Description

	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`ciaValue = 5`
	NNMi evaluates the expression above as follows:
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

- **<=** Finds all values less than or equal to the value specified. Click here for an example.

  Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

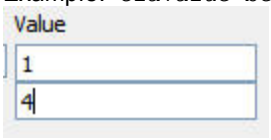
  Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

  `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

matches any incident that contains a varbind with values other than **1** and **2**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

matches any incident that contains a varbind with values other than **1** and **2**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

  `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

  in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.
Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click

Name	Description

Button	Description

here for examples.

Example 1

```
AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

here for examples.

Example 1

```
AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

here for examples.

Example 1

Name	Description

Button	Description

```
AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
     ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
          ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
```

Name	Description

Button	Description
	ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Configure Enrichment Settings for a Management Event Incident

For information about each Management Events tab:

NNMi enables you to fine tune and enhance incidents based on Interface Group, Node Group, or default Enrichment settings. NNMi applies your Enrichment settings in the following order. Only the first match applies.

1. Interface Group (Management Event Configuration Form: Interface Settings tab)
2. Node Group (Management Event Configuration Form: Node Settings tab)
3. Enrich configuration settings without specifying an Interface Group or Node Group (Management Event Configuration Form: Enrichment tab)

The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To

Note: Any configuration you specify for Severity, Priority, or Message overrides those values provided in the Management Event Configuration Form: Basics information.

A Payload Filter allows you to use the data that is included with any of the following items before they are stored as incidents in NNMi:

- Traps generated from an SNMP agent
- Management incidents that are generated by NNMi
- Events generated by NNM 6.x or 7.x management stations

Examples of the type of data that can be used as a Payload Filter include SNMP trap varbind names and values as well as CIA (Custom Incident Attribute) names and values. For example, you might want NNMi to suppress a particular status change notification trap for a specified Node Group or Interface Group. To do so, you could include the name of the trap varbind that stores this information as well as the particular status change value string the traps that you want to suppress should contain.

See "Configure Incident Enrichment Settings for an Interface Group (Management Events)" (on page 752) for information about how to enrich an incident for an Interface Group with or without a Payload Filter.

See "Configure Incident Enrichment Settings for a Node Group (Management Events)" (on page 787) for more information about how to enrich an incident for a Node Group with or without a Payload Filter.

To configure Enrichment settings for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **Management Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.

b. Select **Incident Configuration**.

c. Select the **Management Events** tab.

d. Do one of the following:

i. To create an incident configuration, click the New icon, and continue.

ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.

iii. To delete an incident configuration, click the Delete icon.

2. Select the **Enrichment** tab.

3. Do one of the following:

a. To create a new configuration, click the New icon.

b. To edit an existing configuration, select a row, and click the Open icon.

4. Provide the required information (see table)

5. Click **Save and Close** to save your changes and return to the previous form.

Enrichment Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: • Accounting • Application Status • Configuration • Fault • Performance • Security • Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443) for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Possible values include: • Address • Aggregated Port • Board • Connection • Correlation • HSRP • Interface • Node

Name	Description
	• OSPF See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" (on page 443) for more information.
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below: **Normal** - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there may or may not be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: **None** **Low** **Medium** **High** **Top** **Note**: The icons are displayed only in table views.

Name	Description
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact See Incident Form: General Tab for more information.
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right. You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (SNMP Trap Incident)" (on page 446) "Include Custom Incident Attributes in Your Message Format (SNMP Trap Incident)" (on page 452)
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 2048 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Configure Dampening Settings for a Management Event Incident

For information about each Management Events tab:

NNMi enables you to delay the following for an incident configuration based on the Source Object's participation in an Interface Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

You can configure the Dampening settings based on Interface Group, Node Group, or default Dampening settings. NNMi applies your Dampening settings in the following order. Only the first match applies.

1. Interface Group (Management Event Configuration Form: Interface Settings tab)
2. Node Group (Management Event Configuration Form: Node Settings tab)
3. Dampening configuration settings without specifying an Interface Group or Node Group (Management Event Configuration Form: Dampening tab)

When using the Dampening configuration, note the following:

- By default, the Dampen Interval for NNMi Management Event Incidents is six minutes.

  Note: Any Dampen Interval set for less than six minutes might not provide enough time for the Causal Engine to complete its analysis. As a result, incidents associated with intermediate symptoms might appear as root cause incidents.

- Duplicate and Rate Correlation incidents inherit the Dampening settings from their Correlated Children. If the Correlated Children are Closed while Dampened, and therefore deleted, NNMi retains the parent Duplicate or Rate Correlation incident. See "Correlate Duplicate Incidents (Deduplication Configuration)" (on page 346) and "Track Incident Frequency (Rate: Time Period and Count)" (on page 347) for more information about Duplicate and Rate Correlation incidents.

  Note: To view the number of incidents Closed and deleted while dampened, access the NNMi System Health Report Dampened Metrics (**Help** → **System Information** → **Health** tab).

- After the Dampen Interval has passed, NNMi changes the Lifecycle State to REGISTERED.

  See About the Incident Lifecycle for more information about Lifecycle State.

See "Configure Incident Dampening Settings for an Interface Group (Management Events)" (on page 762) for information about how to configure Dampening settings for an Interface Group with or without a Payload Filter.

See "Configure Incident Dampening Settings for a Node Group (Management Events)" (on page 797) for more information about how to configure Dampening settings for a Node Group with or without a Payload Filter.

To configure Dampening settings for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **Management Events** tab.
   d. Do one of the following:
      i. To create a configuration, click the New icon, and continue.
      ii. To edit configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete a configuration, click the Delete icon.
2. Select the **Dampening** tab.
3. Provide the required information (see table)
4. Click **Save and Close** to save your changes and return to the previous form.

Dampening Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's Dampening settings. To temporarily disable the Dampening configuration settings for the selected incident configuration, clear **Enable** ☐. To enable the Dampening configuration settings for the selected incident configuration, click **Enable** ☑.
Hour	Specifies the number of hours to be used for the Dampen Interval.
Minutes	Specifies the number of minutes to be used for the Dampen Interval. Note the following: ■ The maximum Dampen Interval is 60 minutes. ■ Any Dampen Interval set for less than six minutes might not provide enough time for the Causal Engine to complete its analysis. As a result, Incidents associated with intermediate symptoms might appear as Root Cause Incidents.
Seconds	Specifies the number of seconds to be used for the Dampen Interval.

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	
	When creating a Payload Filter, note the following:
	
	■ Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	
	■ You must use a `ciaName` that already exists in the trap or event you are configuring.
	
	■ Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	
	■ View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	
	■ The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	
	The following example filters incidents on voltage state:
	
	`AND`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`ciaValue = 5`
	
	NNMi evaluates the expression above as follows:
	
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	
	■ The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you

Name	Description

append to, replace, or change the indentation of the expression that is selected.

- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

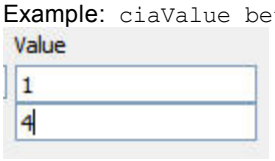
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
Operator	Valid operators are described below. ■ **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. ■ **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**. ■ **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**. ■ **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**. ■ **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. ■ **between** Finds all values equal to and between the two values specified. Click here for an example.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue

Example: `ciaValue between`



matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for more information.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue

The period asterisk (.*) characters mean *any number of characters of any type at this location*.

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue

defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

The period asterisk (.*) characters mean *any number of characters of any type at this location*.

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

The period asterisk (.*) characters mean *any number of characters of any type at this location*.

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at :

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue

ase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

The period asterisk (.*) characters mean *any number of characters of any type at this location*.

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

ase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

The period asterisk (.*) characters mean *any number of characters of any type at this location*.

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

ase/6/docs/api/java/util/regex/Pattern.html for more information. Click here for an example.

The period asterisk (.*) characters mean *any number of characters of any type at this location*.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
	The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search.

Name	Description

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ■ ciaName ■ ciaValue
	Note the following: ■ The values you enter are case sensitive. ■ NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. ■ The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.

Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples.
	Example 1
	`AND`
	`ciaName =.1.3.6.1.4.1.9.9.13.1.3.1.6`
	`OR`
	`ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3`

Name	Description

Button	Description

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6
     OR
     ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3
```

Example 2

```
AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in:

```
AND
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
```

Name	Description
	<table><tr><th>Button</th><th>Description</th></tr><tr><td>Delete</td><td>Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr></table>

Name	Description

Configure Deduplication for a Management Event Incident

For information about each Management Events tab:

The deduplication configuration determines what values NNMi should match to detect when an SNMP trap, management event, or remote NNM 6.x/7.x event is a duplicate.

Note the following:

- NNMi applies only one deduplication configuration per incident . If NNMi generates an incident using a specified deduplication configuration, NNMi continues to correlate duplicate incidents using the original configuration. To use a different deduplication configuration for an incident, first delete the current deduplication incident (created using the original deduplication configuration). NNMi generates the next deduplication incident according to the new deduplication configuration settings.
- By default, NNMi updates the **Duplicate Count** every 30 seconds. This interval cannot be changed.
- NNMi continues to update the duplicate count regardless of an incident's lifecycle state. For example, if an incident's **Lifecycle State** is set to **Closed**, the duplicate count continues to be incremented. See About the Incident Lifecycle for more information. This behavior helps you identify situations in which the incident is not yet fixed. Take note if the Duplicate Count is incremented after a lengthy time period has elapsed, which might indicate there is a new problem with the node, interface, or address.
- Each time you stop and restart ovjboss, any incidents that have not yet been correlated or persisted are lost. This means that after a restart of ovjboss, an incoming incident might not be correlated as expected. For example, after a restart of ovjboss, a duplicate incident might not be correlated under its original parent incident. Instead, a new parent incident might be generated. See "Stop or Start an NNMi Process" (on page 45)for more information about starting and stopping the ovjboss process.
- If a Duplicate Correlation Incident is dampened, note the following:
  - Duplicate Correlation Incidents inherit the Dampening settings from its Correlated Children.
  - NNMi always retains the Parent Duplicate Correlation incident, even if its Child Incidents are Closed and subsequently deleted.

    See "Dampening Incident Configurations" (on page 356) for more information about Dampening an Incident Configuration.

To specify or delete a deduplication configuration:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **Management Events tab**.
   d. Do one of the following:
      i. To create a deduplication configuration, click the New icon, and continue.
      ii. To edit a deduplication configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete a deduplication configuration, click the Delete icon.
2. Select the **Deduplication** tab.

3. Provide the required information (see "Deduplication Attributes" table).
4. Click **Save and Close** to save your changes and return to the previous form.

Deduplication Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's deduplication configuration. To temporarily disable the deduplication configuration setting, clear **Enable** ☐. To enable the deduplication configuration setting, click **Enable** ☑. **Note**: After a deduplication configuration is enabled, NNMi increments the **Duplicate Count** for an associated incident regardless of the **Lifecycle State** value. For example, if an incident's **Lifecycle State** is set to **Closed**, the duplicate count continues to be incremented. See About the Incident Lifecycle for more information.
Count	Specifies the number of duplicate incidents for the current configuration that NNMi stores at one time. For example, if the Count is 10, after NNMi receives 10 duplicate incidents, NNMi deletes the first (oldest) duplicate incident and keeps the eleventh. (NNMi stores ten maximum.)
Hour Interval	Used with the Minute and Second Intervals to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Hour Interval value is 1, and no Minute or Second Intervals are specified, and the duplicate incident is not generated within one hour, NNMi generates a new duplicate incident the next time it occurs.
Minute Interval	Used with the Hour and Second interval to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Minute Interval is 30 and no Hour or Second Intervals are specified, and the duplicate incident is not generated within 30 minutes, NNMi generates a new duplicate incident the next time it occurs.
Second Interval	Used with the Hour and Minute Intervals to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Second Interval is 120 and no Hour or Minute Intervals are specified, and the duplicate incident is not generated within 120 seconds, NNMi generates a new duplicate incident the next time it occurs.
Correlation Incident Config	Used to access the out-of-the box deduplicaton configuration provided by NNMi. Select the default value **Duplicate Correlation**. **Note**: You can choose to use this configuration as is or edit it. If you want to create a new deduplication configuration, you must create a new incident configuration. After you have created a new incident configuration, it appears in the **Quick Find** list of options. See "Lookup Fields" (on page 27) for more informationn about Quick Find.
Comparison Criteria	Specify the attribute values that must match before the incident is identified as a duplicate. The possible attributes consist of the following choices. • **Name** - The **Name** attribute value from the Incident form: General tab. • **CIA** - Represents any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " (on page 346): ▪ The **Value** attribute from the Incident form: Custom Attributes tab

Name	Description
	▪ An SNMP varbind Object ID ▪ An SNMP varbind position number If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " (on page 346)
	• **SourceNode** - The **Source Node** attribute value from the Basics attributes listed on the Incident form. The Source Node value is the IP Address or Name of the node for which the incident was generated. **Note**: The Source Node must be stored in the NNMi database. • **Source Object** - The **Source Object** attribute value from the Basics attributes listed on the Incident form. **Note**: The Source Object must be stored in the NNMi database. **Note**: Each attribute value in the option you select must match before the incident is identified as a duplicate. For example, if you select **Name**, only the Incident Name value must match. If you select **Name SourceNode SourceObject CIA**, the Incident Name, Source Node, Source Object, and all Custom Incident Attribute values that you configure as a Parameter Value must match before NNMi identifies the incident as a duplicate. Selecting an option that includes CIA enables you to further refine the deduplication criteria. For example, you might want to configure deduplication for incidents with CIA values that specify the same State attribute value for a particular network object. For a description of each Comparison Criteria option, click here.

Comparison Criteria	Description
Name	Value of the **Name** attribute from the Incident form: General tab must match.
Name CIA	Each of the following values must match: • **Name** attribute from the Incident form: General tab • **CIA** - Represents the Value associated with any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " (on page 346): ▪ Name of a Custom Incident Attribute (CIA) provided by NNMi. (See the Incident form: Custom Attributes tab ▪ An SNMP varbind Object ID ▪ An SNMP varbind position number If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " (on page 346)
Name SourceNode	**Note**: Select this option only if the Source Node is stored in the NNMi database. Each of the following values must match: • **Name** attribute from the Incident form: General tab • The **Source Node** attribute value from the Basics attributes listed on the Incident form
Name	**Note**: Select this option only if the Source Node is stored in the NNMi

Name attribute from the Incident form: General tab

Name	Description

Name	Description
Comparison Parameter List	*Optional*. If you selected a Comparison Criteria that includes CIA, you must populate one or more rows in this table. See "Deduplication Comparison Parameters Form " (on page 346).

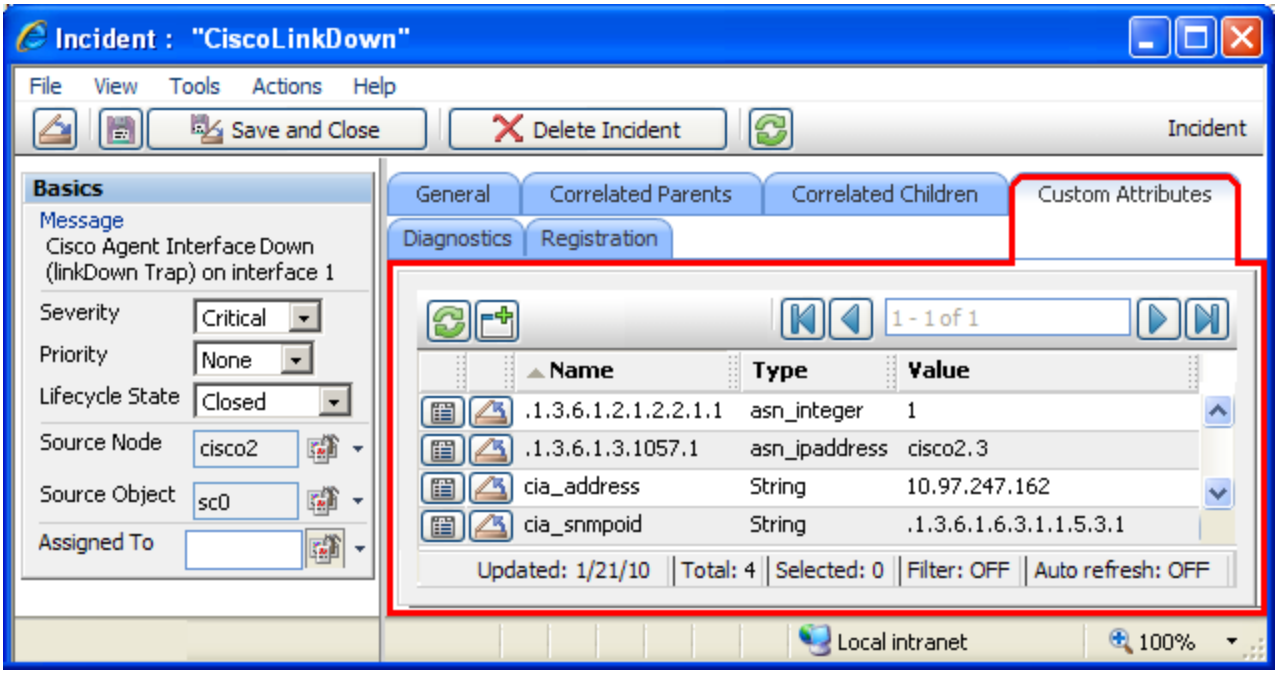
Deduplication Comparison Parameters Form (Management Events)

Comparison Parameter values enable accurate identification of duplicate incidents. Custom Incident Attributes (CIAs) are used as Comparison Parameter values. There are two categories of CIAs:

- SNMP trap varbind values (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312).

The group of available CIAs depends on which incident you are configuring for this Deduplication (for example, CiscoLinkDown). To see which CIAs are available, navigate to an Incident view, select an instance of that incident-type, click the Open icon and navigate to the Custom Attributes tab. The items listed in the table are the CIAs for that particular incident-type. For example, all CiscoLinkDown incidents would have the same group of CIAs shown in the illustration below.

Note: You can also use the CIA (varbind) position number.



To specify a CIA to use in the identification criteria for duplicate incidents:

1. Navigate to the **Deduplication Comparison Params** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Management Events** tab.
    d. Do one of the following:

- To create a new configuration, click the New icon.
- To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.

e. On the form that opens, navigate to the **Deduplication** tab.

f. Locate the **Comparison Parameter List** table.

g. Do one of the following to specify which CIA:

- To add a Custom Incident Attribute parameter specification, click the New icon, and continue.
- To edit an existing Custom Incident Attribute parameter specification, click the Open icon that precedes the configuration you want to edit, and continue.

2. In the Parameter Value field, type (or copy and paste) the exact text string from the Incident form, Custom Attribute tab, **Name** attribute value:
   - NNMi-provided CIA value (see "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312)).
   - SNMP trap varbind identified by the Abstract Syntax Notation value (ASN.1).
3. Click **Save and Close** to save your changes and return to the previous configuration form.

Configure Rate (Time Period and Count) for a Management Event Incident

For information about each SNMP Traps tab:

Use Rate configuration to track incident patterns *based on the number of incident reoccurrences within a specified time period*. After the count within the specified time period is reached, NNMi emits a Rate Correlation incident and continues to update the Correlation Notes with the number of occurrences within that rate.

As long as your defined criteria (Count and Hours, Minutes, Seconds) is sustained, the following information is updated in the Correlation Notes of the Rate Correlation incident:

- the actual number of occurrences of incidents for that sustained rate (Count)
- the sustained time interval (Hours, Minutes, Seconds)

For example, you can set a Rate configuration to track when a link is intermittently down at least three times in 30 minutes. NNMi shows the first occurrence of the rate incident in the incident view and uses Correlation Notes to update the number of incidents and time interval to reflect all the incremental incident occurrences and time periods. To continue the example, if the rate of three times in 30 minutes is sustained for 90 minutes, NNMi updates the Correlation Notes to specify that 9 incidents occurred in 90 minutes.

NNMi provides preconfigured Rate correlations. You can add new Rate correlations.

When you open the Incident form of the newest instance:

- On the General tab, two fields notify you that the Rate correlation is working:
  - **Correlation Nature**: Rate

- On the **Correlated Children** tab, each incident is listed in the table.
- If a Rate Correlation Incident is dampened, note the following:
  - Rate Correlation Incidents inherit the Dampening configuration settings from its Correlated Children.
  - NNMi always retains the Parent Rate Correlation Incident, even if its Child Incidents are Closed and subsequently deleted.

    See "Dampening Incident Configurations" (on page 356) for more information about Dampening an Incident Configuration.

To establish a rate correlation within an incident configuration:

1. Navigate to the **Rate** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Incident Configuration**.
   c. Select the **Management Events** tab.
   d. Do one of the following:
      - To create a new configuration, click the New icon.
      - To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
   e. On the form that opens, locate the **Rate** tab.
2. Provide the definition for this Rate Configuration (see the "Rate Configuration Definition" table).
3. *Optional*. If your Comparison Criteria includes custom incident attributes (CIA) to identify one specific incident, use the Comparison Parameter List table to define each CIA.See "Rate Comparison Parameters Form" (on page 347).
4. Click **Save and Close** to save your changes and return to the previous form.

Rate Configuration Definition

Attribute	Description
Enable	Use this attribute to temporarily disable an incident's rate settings.To temporarily disable the Dampen Configuration settings for the selected incident configuration, clear **Enable** ☐ .To enable the Dampen Configuration settings for the selected incident configuration, click **Enable** ☑. If enabled, NNMi actively tracks any reoccurrences of the designated incident within the time period you specify, and generates a Rate incident.
Count	Specify the number of reoccurrences required before your Rate Configuration starts working.
Set the Time Period	Specify a time duration within which the reoccurrences are measured. Fill in one or more of the following attribute fields: **Hours** **Minutes** **Seconds**
Correlation	Click the icon and select Quick Find. Select **Rate Correlation** from the list.

Attribute	Description
Incident Config	
Comparison Criteria	Specify which group of attributes must match before the incident is identified as a duplicate. The possible groups of attributes consist of the following choices. **Name** value of the Incident (from the General tab on the Incident form). **Source Node** value (from the Basics group on the Incident form). Address or name of the node for which the incident was generated. **Source Object** value (from the Basics group on the Incident form). For example, the Source Object for a LinkDown incident is **interface**. **CIA** custom incident attribute values (select from the list displayed on the Custom Attributes tab on the Incident form). If you want to use CIA as part of your comparison criteria, see "Rate Comparison Parameters Form (Management Events)" (on page 848).
Comparison Parameter List	*Optional*. If you selected a Comparison Criteria that includes CIA, you must populate one or more rows in this table. See "Rate Comparison Parameters Form (Management Events)" (on page 848).

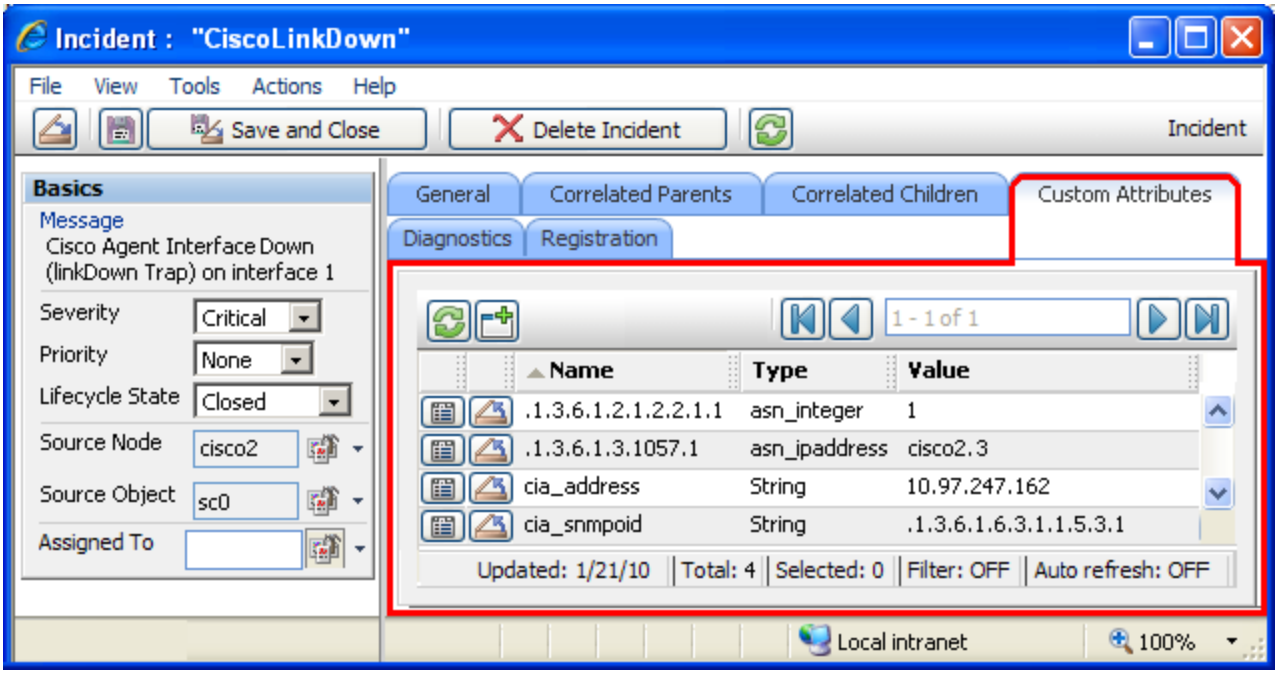
Rate Comparison Parameters Form (Management Events)

Custom Incident Attributes (CIAs) are used as parameter values. Parameter values enable accurate identification of duplicate incidents. There are two categories of CIAs:

- SNMP trap varbind values (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312).

The group of available CIAs depends on which incident you are configuring for this Rate (for example, CiscoLinkDown). To see which CIAs are available, navigate to an Incident view, select an instance of that incident-type, click the Open icon and navigate to the Custom Attributes tab. The items listed in the table are the CIAs for that particular incident-type. For example, all CiscoLinkDown incidents would have the same group of CIAs shown in the illustration below.

Note: You can also use the CIA (varbind) position number.



To specify a CIA to use in the identification criteria for duplicate incidents:

1. Navigate to the **Rate Comparison Params** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Management Events** tab.
    d. Do one of the following:
        - To create a new configuration, click the New icon.
        - To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
    e. On the form that opens, navigate to the **Rate** tab.
    f. Locate the **Comparison Parameter List** table.
    g. Do one of the following to specify which CIA:
        - To add a Custom Incident Attribute parameter specification, click the New icon.
        - To edit an existing Custom Incident Attribute parameter specification, click the Open icon in the row representing the configuration you want to edit.
2. In the Parameter Value field, type (or copy and paste) the exact text string from the Incident form, Custom Attribute tab, **Name** attribute value:
    - NNMi-provided CIA value (see "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312)).
    - SNMP trap varbind identified by the Abstract Syntax Notation value (ASN.1).
3. Click **Save and Close** to save your changes and return to the previous configuration form.

Configure Actions for a Management Event Incident

For information about each Management Events tab:

For information about each Actions tab:

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 (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

You can configure actions for incidents generated from SNMP traps, NNM 6.x or 7.x events, and the NNMi management events. Any time an incident configuration changes, the action directory is rescanned and any Jython files are reloaded to the NNMi database. See "Lifecycle Transition Action Form (Management Events)" (on page 851) for more information about the actions directory.

Tip: Copy any required Jython files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (Management Events)" (on page 851) for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. To view the contents of the Actions log, use the **Tools** → **Incident Actions Log** menu option.

See "Verify that NNMi Services are Running" (on page 47) for more information about log files and where they are located.

NNMi sets the default values described in the following table. For information about how to change the default values, see the HP Network Node Manager i Software Deployment Reference, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

Action Server Properties

Property	Description	Default Value
numProcess	Number of actions that can be run at one time.	150
processTimeout	Amount of time in seconds that an action can run before the action server terminates the action.	300
numJythonThreads	Number of threads the action server uses to run Jython scripts	10
userName	User name under which the action server runs.	bin

To configure an automatic action for an incident:

1. Navigate to the **Actions** tab.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration.**
    c. Select the **Management Events** tab.

d. Do one of the following:

- To create an incident configuration, click the New icon, and continue.
- To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue
- To delete an incident configuration, select a row, and click the Delete icon.

e. Select the **Actions** tab.

2. From the **Lifecycle Actions** table toolbar, do one of the following:

- To create an Action configuration, click the New icon, and continue.
- To edit an Action configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
- To delete an Action configuration, click the Delete icon.

3. In the "Lifecycle Transition Action Form (Management Events)" (on page 851), provide the required information.

4. Click **Save and Close** to save your changes and return to the previous form.

   The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type.

Lifecycle Transition Action Form (Management Events)

For information about each Actions tab:

Use this form to enter the command you want to run when an incident of the type you are configuring is at a particular Lifecycle State. For example, when an incident is generated (**Registered**), you might want to automatically open a trouble ticket or email or page your network operator.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable on the Actions tab or using the **Actions** → **Enable Configuration** option.

To configure an action for an incidents:

1. Navigate to the **Lifecycle Transition Actions** form:

   a. From the workspace navigation pane, select the **Configuration** workspace.

   b. Select **Incident Configuration**.

   c. Select the **Management Events** tab.

   d. Select the **Actions** tab.

   e. From the **Lifecycle Transition Action** table toolbar, do one of the following:

   - To create an Action configuration, click the New icon, and continue.
   - To edit an Action configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an Action configuration, click the Delete icon.

2. Make your configuration choices (see table).

Note: NNMi reloads the configuration information anytime the incident configuration is changed.

3. Click **Save and Close** to save your changes and return to the previous form.

Create Action Attributes

Attribute	Description
Lifecycle State	Select a Lifecycle State from the drop-down menu.
Command Type	If you provided a Jython command, select **Jython** from the drop-down list. If you are using an executable or bat file, select **ScriptOrExecutable** from the drop-down list.

Command	Enter one of the following:
	• A Jython method with the required parameters
	• Executable command for the current operating system with the required parameters.
	When entering a *Command* value, note the following:
	• Left or right bracket ([ ]) and backtick (` Unicode character: 0060 hex = 96 dec) characters are not allowed in the **Command** attribute. If you need these characters in your shell script, place them in a shell script file and reference that file from the **Command** attribute.
	• **Windows only**: Shell commands are not allowed in the **Command** attribute. If you need to use shell commands, place them in a shell script file and reference that file from the **Command** attribute.
	• Verify that you do not have two Jython methods with the same name. Otherwise, NNMi is not able to tell which is the correct method to load.
	• You can use the same Jython method for more than one incident configuration.
	• Python files or other executable scripts need to reside in the following directory:
	Note: Use absolute paths to executables instead of relying on the PATH variable as it might not be set correctly.
	Windows:
	`%NnmDataDir%\shared\nnm\actions`
	UNIX:
	`/var/opt/OV/shared/nnm/actions`
	• NNMi provides a set of parameters that can pass attribute values from an incident configuration. See "Valid Parameters for Configuring Incident Actions (Management Events)" (on page 859) for more information.

Configure a Payload Filter for an Action (Management Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline

as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

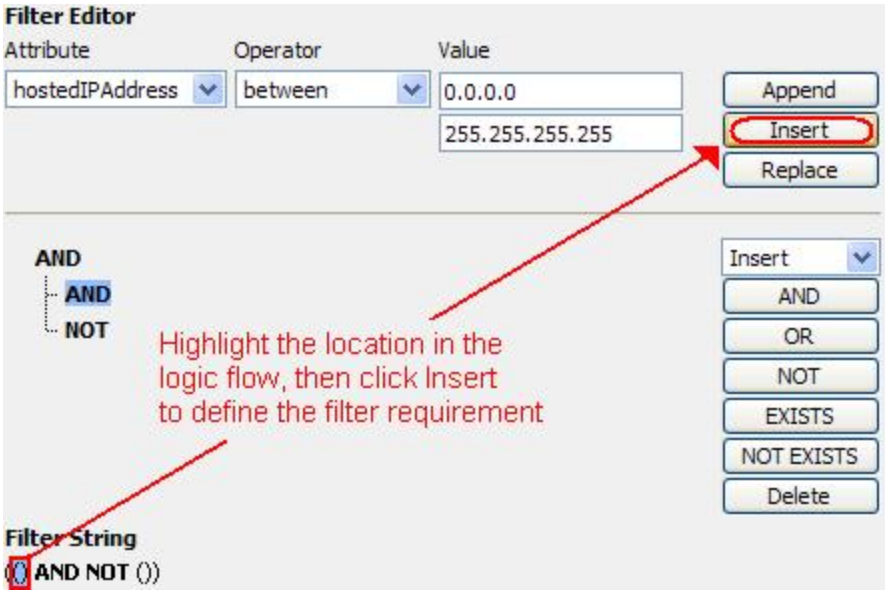
1. Navigate to the **Management Event Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Incident Configuration**.
    c. Select the **Management Events** tab.
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
        iii. To delete an incident configuration, click the Delete icon.
2. Select the **Actions** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, and click the Open icon.
4. Select the **Payload Filter** tab.
5. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".
    a. Plan out the logic needed for your Filter String.
    b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

        For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

        ```
        (( ) AND NOT ( ))
        ```

    c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



6. Click **Save and Close**.
7. Click **Save and Close** to save your changes and return to the previous form.

When creating a Payload Filter, note the following:

- Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class)
- You must use a `ciaName` that already exists in the trap or event you are configuring.
- Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.

  The following example filters incidents on voltage state. Using this Payload Filter, you could then configure the Basics settings of the Enrichment Configuration to set the severity and message format to all incidents that return a state value of `4` or `5`.

```
OR

  ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

   ciaValue = 4

  AND

   ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7

   ciaValue = 5
```

  NNMi evaluates the expression above as follows:

```
(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 4) OR (ciaName =
.1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)
```

NNMi finds all incidents with a varbind value of `.1.3.6.1.4.1.9.9.13.1.2.1.7` and CIA value of **4** or **5**.

- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName!=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind value less than **6** . • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind value less than or equal to **6** . • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind value greater than **4** . • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4** . • **between** Finds all traps or events that include a varbind value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`

Attribute	Description

matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4** .

Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  matches any incident that contains a varbind value of either **4** or **5** .

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with varbind values.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with no varbind values.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Examples:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters.

  `ciaValue like *Chicago*` finds all traps or events that contain a varbind

Attribute	Description
	value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in` matches any incident that contains a varbind with values other than **1** and **2** . **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **`.1.3.6.1.4.1.9.9`**. `ciaValue not like *Chicago*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
AND <--> OR	Switches the selected Boolean Operator to either AND or OR. If the selected Boolean Operator is AND, the value is changed to OR. If the selected Boolean Operator is OR, the value is changed to AND.

Outdent	Moves a nested expression up one expression level. If your filter string includes only one nested expression, **Outdent** replaces the top-level expression. Click here for examples.
	Example 1
	`AND`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6`
	`  OR`
	`  ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3`
	Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3` and selecting **Outdent**, results in:

Button	Description
	AND ciaName = .1.3.6.1.4.1.9.9.13.1.3.1.6 OR ciaName = .1.3.6.1.4.1.9.9.13.1.4.1.3 **Example 2** AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 Placing the cursor at `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` and selecting **Outdent**, results in: AND ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Valid Parameters for Configuring Incident Actions (Management Events)

When configuring incident actions, you may want to use incident information as part of the action. NNMi provides the following parameter values. Use these parameters as variables in your Jython or executable files.

Tip: See the Using the Incident Form for more information about the parameter values.

Note: NNMi stores varbind values as custom incident attributes (CIAs).

See "Lifecycle Transition Action Form" (on page 415) for more information about configuring incident actions.

Valid Parameters Visible From an Incident's Form

Parameter Value	Description
$category, $cat	Value of the Category attribute in the Incident form.
$count, $cnt	Value representing the number of Custom Incident Attributes that appear in the Incident form.
$family, $fam	Value from the Family attribute in the Incident form.
$firstOccurrenceTime, $fot	Value from the First Occurrence Time attribute in the incident form.
$lastOccurrenceTime, $lot	Value from the Last Occurrence Time attribute in the incident form.
$lifecycleState, $lcs	Value from the Lifecycle State attribute in the Incident form.
$name	Value of the Name attribute from the incident configuration.

Parameter Value	Description
$nature, $nat	Value from the Nature attribute in the Incident form.
$origin, $ori	Value from the Origin attribute in the Incident form.
$originOccurrenceTime, $oot	Value from the Origin Occurrence Time attribute in the incident form.
$priority, $pri	Value from the Priority attribute in the Incident form.
$severity, $sev	Value of the Severity attribute of the Incident form.

Valid Parameters Visible from a Node Form

Parameter Value	Description
$managementAddress, $mga	Value from the Management Address attribute of the incident's source Node's form or SNMP Agent form.
$otherSideOfConnectionManagementAddress, $oma	If the incident's Source Node is part of a Layer 2 connection, this attribute is the value of the Management Address of a node on the other side of the Layer 2 connection.
$sourceNodeLongName, $sln	The fully-qualified DNS name as displayed in the Hostname attribute of the incident's source Node's form.
$sourceNodeName, $snn	Value from the Name attribute of the incident's source Node's form.
$sysContact, $sct	Value from the System Contact attribute of the incident's source Node form: General tab.
$sysLocation, $slc	Value from the System Location attribute of the incident's source Node form: General tab.

Valid Parameters Visible from an Interface Form

Parameter Value	Description
$ifAlias, $ifa	Value from the IfAlias attribute for the interface that is the incident's source object.
$ifConfigDupSetting, $icd	Configured Duplex Setting on the port associated with the interface that is the incident's source object.
$ifDesc, $idc	Value from the ifDesc attribute for the interface that is the incident's source object.
$ifIndex, $idx	Value from the ifIndex attribute for the interface that is the incident's source object.
$ifIpAddr, $iia	IP Address values associated with the interface that is the incident's source object. If multiple IPaddresses are associated with the interface, this parameter returns a comma-separated list.
$ifName, $ifn	Value from the ifName attribute for the interface that is the incident's source object.

$ifPhysAddr, $ipa	Value from the Physical Address attribute for the interface that is the incident's source object.
$ifSpeed, $isp	Value from the ifSpeed attribute for the interface that is the incident's souce object.
$ifType, $itp	Value from the ifType attribute for the interface that is the incident's souce object.

Valid Parameters Visible from a Layer 2 Connection Form

Parameter Value	Description
$otherSideOfConnectionConfigDupSetting, $ocd	If the incident's source Node is part of a Layer 2 connection, this parameter contains the Configured Duplex Setting on the port associated with the interface on the other side of the connection.
$otherSideOfConnectionIfAlias, $oia	If the incident's Source Node is part of a Layer 2 connection, this parameter is the value of the ifAlias of one of the interfaces on the other side of the Layer 2 connection.
$otherSideOfConnectionIfDesc, $odc	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifDescr attribute value for the interface on the other side of the Layer 2 connection.
$otherSideOfConnectionIfIndex, $odx	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifIndex attribute value for the interface on the other side of the connection.
$otherSideOfConnectionIfName, $ofn	If the incident's Source Node is part of a Layer 2 connection, this parameter contains the ifName attribute value for the interface on the other side of the connection.

Valid Parameters Visible from a VLAN Form

Parameter Value	Description
$impVlanIds, $ivi	Value from the VLAN Id attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Port tab and open the form for the VLAN of interest. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.
$impVlanNames, $ivn	Value from the VLAN Name attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Ports tab of the Interface form. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.

Valid Parameters Not Visible From a Form

Parameter Value	Description
$id	Unique Object Identifier attribute value for the incident (unique across the entire NNMi Database).

Parameter Value	Description
$firstOccurrenceTimeMS, $fms	Value from the First Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$lastOccurrenceTimeMS, $lms	Value from the Last Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$oid	Value of the unique object identifier (oid) for the incident configuration that originated from either an SNMP trap, Management Event, or Remote NNM 6.x or 7.x event.
$otherSideOfConnection, $osc	If the incident's Source Node is part of a Layer 2 connection, this attribute is the following combination of values for the node and one of its interfaces on the other side of the Layer 2 connection: The fully-qualified DNS name of the node appended with the interface Name in the following format: <*fully-qualified DNS name*>[*interface_name*]
$originOccurrenceTimeMS, $oms	Value from the Origin Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$sourceNodeUuid, $snu	Universally Unique Object Identifier attribute value of the source node object for the incident (unique across all databases). This identifier distinguishes the source node object instance from all other node objects.
$sourceObjectClass, $soc	Value of the object class for the object you want to include. Use this parameter when you need to request more details of a class of objects through a web service. Examples of object classes include: `com.hp.ov.nms.model.core.Interface` and `com.hp.ov.nms.model.snmp.SnmpAgent`.
$sourceObjectName, $son	Value from the Name attribute of the source object. For example, an interface object is named according to the MIB ifName. Each ifName varies according to the vendor's conventions. Using the name `4/1` as an example, `4` represents `the board number and 1` represents the port number.
$sourceObjectUuid, $sou	Universally Unique Object Identifier attribute value of the source object for the incident (unique across all databases). This identifier distinguishes the source object instance from all other similar object instances..
$uuid	Universally Unique Object Identifier attribute value of the incident (unique across all databases). This identifier distinguishes the incident object instance from all other incident objects.

Valid Parameters Established in Custom Incident Attributes

Parameter Value	Description
$<position_ number>	Value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1`

Parameter Value	Description
	NNMi stores varbind values as Custom Incident Attributes. If you know the varbind position number, use this parameter.
$<CIA_name>	Value of the name that is used for the custom incident attribute. For example, `$mycompany.mycia`. NNMi provides CIA values for configuring Management Events. See Custom Incident Attributes Provided by NNMi for more information about custom incident attributes.
$<CIA_oid>	Value of the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this parameter when you are not certain of a custom incident attribute (varbind) position number.
$*	Used to indicate you want all of the custom incident attribute values originating as varbinds, to be passed to the action configuration. Each varbind is returned in the following format: `$<CIA_name>:<CIA_value>` in which the custom incident attribute name appears followed by the custom incident attribute value.

The function described in the following table replaces the specified numeric value with the associated text value stored in the CIA.

Note: The associated MIB must have been loaded using the nnmloadmib.ovpl command.

Functions to Generate Values Within Incident Messages

Function	Description
$text($<position_number>)	The *<position_number>* argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1`. After the function runs, NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.
$text($<CIA_oid>)	The <CIA_oid> argument specifies the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this argument to the $text function when you are not certain of a custom incident attribute (varbind) position number. After the function runs, NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.

Troubleshoot Incident Configurations

The NNMi **Actions** menu enables you to open an Incident Configuration from either an incident or an incident view. This feature is useful when you are monitoring incoming incidents to determine whether incidents are generated as expected. After you make any required changes, you can easily verify your changes the next time the incident occurs.

Note: You must have Administrator role to use this action.

To open an Incident Configuration form from an incident view:

1. Navigate to the incident view of interest. (For example, select the **Incident Browsing** workspace, **Root Cause Incidents** view.)
2. In the table view, click the ☑ selection box that precedes each incident of interest.
3. In the main toolbar, select **Actions** → **Open Incident Configuration Form**.

   NNMi opens one Incident Configuration form for each type of incident selected.

To open an Incident Configuration form from an Incident form:

1. Navigate to the incident view of interest. (For example, select the **Incident Browsing** workspace, **Root Cause Incidents** view.)
2. In the table view, click the Open icon in the row representing the configuration you want to edit.
3. In the main toolbar, select **Actions** → **Open Incident Configuration Form**.

   NNMi opens the Incident Configuration form for the current incident.

8. **Note**: Any configuration changes you make to an incident apply only to future incidents.

   The NNMi **Actions**→ **Incident Configuration Report** menu also enables you to view configuration reports for the following kinds of configurations for an incident:

   - Action Results
   - Dampen Results
   - Enrichments
   - Global Manager Forwarding (*NNMi Advanced -Global Network Management*) Available on Regional Managers.
   - Suppression Results

   See "View an Incident Configuration Report" (on page 864) for more information.

View an Incident Configuration Report

The NNMi **Actions** menu enables you to view a report of the following incident configurations:

- Action Results
- Dampen Results
- Enrichment
- Global Manager Forwarding (*NNMi Advanced - Global Network Management feature*)
- Suppression Results

Note: You must have Administrator role to use these actions.

Viewing an incident configuration report helps you determine the following:

- (*NNMi Advanced - Global Network Management feature*). On a Regional Manager, reports whether NNMi forwards occurrences of the selected incident configuration to Global Managers.
- The configuration settings (Interface, Node, or Default) NNMi is using for a selected incident.
- Whether the selected configuration (Suppression, Enrichment, or Dampening) is enabled.

- Whether NNMi found any matches for a Payload Filter for the selected configuration (Suppression, Enrichment, or Dampening).

These reports are useful when you want to change an incident configuration and need to determine which settings have been configured, and therefore which settings you might want to change, for the incident.

To view a configuration report for the selected incident:

1. Click the incident for which you want to view a configuration report.
2. Select **Actions→ Incident Configuration Reports.**
3. Select one of the following menu options to indicate the type of configuration report you want to view
   - **Action Results**
   - **Dampen Results**
   - **Report Enrichments**
   - **Global Manager Forwarding** (*NNMi Advanced*)
   - **Suppression Results**

See the Incident Configuration Actions table for a description of each incident configuration report.

Incident Configuration Actions

Action Menu Option	Information Displayed
Action Results	• If the Source Object is an interface, the Interface Group, if any, to which the Source Object belongs. If NNMi reports a matching Interface Group, this indicates it is using the Interface configuration settings for the selected incident. • The Node Group, if any, to which the Source Node belongs. If NNMi reports a matching Node Group, this indicates it is using the Node configuration settings for the selected incident. **Note**: If no matches for an Interface or Node Group are found, this indicates NNMi is using the Default configuration settings for the selected incident. • Whether the Action Configuration is enabled. • The action to be executed. • The Payload Filter, if configured for the incident and whether NNMi found any matches for the Payload Filter.
Dampen Results	• If the Source Object is an interface, the Interface Group, if any, to which the Source Object belongs. If NNMi reports a matching Interface Group, this indicates it is using the Interface configuration settings for the selected incident. • The Node Group, if any, to which the Source Node belongs. If NNMi reports a matching Node Group, this indicates it is using the Node configuration settings for the selected incident. **Note**: If no matches for an Interface or Node Group are found, this indicates NNMi is using the Default configuration settings for the selected incident. • Whether the Dampening configuration is enabled. • The Dampen Interval that is set. • The Payload Filter, if configured for the incident and whether NNMi found any

Action Menu Option	Information Displayed
	matches for the Payload Filter.
Report Enrichments	• If the Source Object is an interface, the Interface Group, if any, to which the Source Object belongs. If NNMi reports a matching Interface Group, this indicates it is using the Interface configuration settings for the selected incident. • The Node Group, if any, to which the Source Node belongs. If NNMi reports a matching Node Group, this indicates it is using the Node configuration settings for the selected incident. **Note**: If no matches for an Interface or Node Group are found, this indicates NNMi is using the Default configuration settings for the selected incident. • Whether the Enrichment configuration is enabled. • The Payload Filter, if configured for the incident and whether NNMi found any matches for the Payload Filter. • The current Severity, Priority, Message Format, and Custom Incident Attributes configuration settings for the incident.
Global Manager Forwarding	(*NNMi Advanced - Global Network Management feature*). Displays the following for each selected incident: • Whether the incident is an SNMP Trap, Remote NNM 6.x/7.x Management Event, or Management Event Configuration. • The name of the incident configuration. • Whether occurrences of the selected incident configuration will be forwarded to Global Managers in your network environment. • The Payload Filter, if configured for the incident and whether NNMi found any matches for the Payload Filter.

Action Menu Option	Information Displayed
Suppression Results	• If the Source Object is an interface, the Interface Group, if any, to which the Source Object belongs. If NNMi reports a matching Interface Group, this indicates it is using the Interface configuration settings for the selected incident. • The Node Group, if any, to which the Source Node belongs. If NNMi reports a matching Node Group, this indicates it is using the Node configuration settings for the selected incident. **Note**: If no matches for an Interface or Node Group are found, this indicates NNMi is using the Default configuration settings for the selected incident. • Whether the Suppress Configuration is enabled. • The Payload Filter, if configured for the incident, and whether NNMi found any matches for the Payload Filter.

Related Topics

"Troubleshoot Incident Configurations" (on page 863)

Using Route Analytics Management Software (RAMS) with NNMi Advanced

HP Route Analytics Management Software (RAMS) is an IP Route Analytics tool that listens to routing protocols and builds a real-time routing topology map. You can use RAMS data to enhance NNMi.

- After you configure RAMS as described in "Configure HP Route Analytics Management Software (NNMi Advanced)" (on page 868), the NNMi Path View provides the following enhanced information:
  - When accessing RAMS data, NNMi is able to provide a Path View more quickly than when using NNMi alone. This is because RAMS does not use SNMP to learn the routing paths. Therefore, it does not need to handle SNMP timeout issues.
  - Because RAMS uses real-time data, rather than data collected from SNMP MIBs, it may also be more accurate than the Path View data collected from NNMi alone.
  - When RAMS data is used to determine the router paths, NNMi ignores the `PathConnections.xml file` (see "Configure a Path View Map" (on page 291)).
- After you configure RAMS as described in "HP RAMS MPLS WAN Configuration (NNMi Advanced)" (on page 869), NNMi provides the following additional information:
  - The Inventory workspace's MPLS WAN Clouds (RAMS) table view shows data. Additional information is provided on each MPLS WAN Cloud (RAMS) form.
  - A new NNMi map, the MPLS WAN Cloud Map view, is available from the Actions menu for participating objects (see MPLS WAN Cloud Map).
  - Path View shows all Equal Cost Multi-Paths (ECMP) rather than being limited to one route.

Note: *NNMi Advanced*. Path View works only with IPv4 addresses. The NNMi Advanced IPv6 address values are not valid choices for Path View. Any devices in your network that are configured with IPv6 addresses cannot be displayed on Path View maps.

For more information on MPLS WAN, see the *HP Route Analytics Management Software User's Guide*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

Related Topics

Path Between Two Nodes

Path Calculation Rules

Path View Limitations

Configure HP Route Analytics Management Software (*NNMi Advanced*)

Route Analytics Management Software (RAMS) is an IP Route Analytics tool that listens to routing protocols and builds a real-time routing topology map. You can use RAMS data to enhance NNMi's ability to trace the route path between the source and destination node when displaying a Path View. You can also use RAMS with the HP RAMS MPLS WAN feature, which allows RAMS to support enterprises that have multiple sites connected by a WAN through Internet Service Providers (ISPs).

To enable NNMi to use RAMS data, you must use the RAMS form to configure each RAMS server you want to use. The RAMS form provides details about the RAMS appliance and the associated RAMS database to be used with NNMi.

Note: *NNMi Advanced*. Path View works only with IPv4 addresses. The NNMi Advanced IPv6 address values are not valid choices for Path View. Any devices in your network that are configured with IPv6 addresses cannot be displayed on Path View maps.

To configure a RAMS server:

1. Navigate to the **RAMS Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **RAMS Configuration**.
2. Do one of the following:
   - To establish a RAMS configuration, click the New icon, and continue.
   - To edit a RAMS configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete a RAMS configuration, select a row and click the Delete icon.
3. Provide the required information (see Basic Attributes table).
4. Click **Save and Close** to save your changes and return to the list of configured RAMS.

Basic Attributes

Attribute	Description
Host	Hostname (*not case-sensitive*) or IP address used to identify the RAMS appliance that you want NNMi to access.
Query Password	Query password configured for the RAMS appliance.
Database Name	Name of the database that NNMi should access. This database must reside on the RAMS appliance that you have identified in the Name attribute.
Priority	Used when you configure more than one RAMS appliance. Determines the order in which NNMi attempts to access the configured RAMS appliances. The lower the number, the higher the priority. For example, the number 1 is the highest priority.

Related Topics

"Using Route Analytics Management Software (RAMS) with NNMi Advanced" (on page 868)

"HP RAMS MPLS WAN Configuration (NNMi Advanced)" (on page 869)

HP RAMS MPLS WAN Configuration *(NNMi Advanced)*

For more information on MPLS WAN, see the *HP Route Analytics Management Software User's Guide*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

The HP NNMi – HP RAMS MPLS WAN integration provides actions for accessing several MPLS WAN tools from the NNMi console.

Enabling the HP NNMi – HP RAMS MPLS WAN Integration

This section describes the steps to enable the HP NNMi – HP RAMS MPLS WAN Integration.

Prerequisites:

- Configure the RAMS Server
- Create an NNMi Web Service Client for RAMS

To configure the connection between the NNMi and the HP RAMS MPLS WAN, follow these steps:

1. In the NNMi console, open the HP NNMi – HP RAMS MPLS WAN Integration Configuration form (Integration Module Configuration > HP RAMS MPLS WAN).
2. Select the **Enable Integration** check box to activate the integration fields on the form.
3. Enter the required information for connecting to the NNMi management server and to the RAMS server.
4. Click **Submit**.
   The status message is displayed. If the status message indicates a problem connecting to the NNMi management server, click **Return**, and then change the values as suggested in the message.

Changing the HP NNMi – HP RAMS MPLS WAN Integration Configuration

To change the connection between the NNMi and the HP RAMS MPLS WAN, follow these steps:

1. In the NNMi console, open the HP NNMi – HP RAMS MPLS WAN Integration Configuration form (Integration Module Configuration > HP RAMS MPLS WAN).
2. Modify the configuration values as appropriate.
3. Verify whether the **Enable Integration** check box in the form is selected, and then click **Submit**.
   The configuration settings are changed.

Disabling the HP NNMi – HP RAMS MPLS WAN Integration

To disable the connection between the NNMi and the HP RAMS MPLS WAN, follow these steps:

1. In the NNMi console, open the HP NNMi – HP RAMS MPLS WAN Integration Configuration form (Integration Module Configuration > HP RAMS MPLS WAN).
2. Clear the **Enable Integration** check box and then click **Submit.**
   The integration fields are disabled and the changes take effect immediately.

HP NNMi – HP RAMS MPLS WAN Integration Configuration Form Reference

The HP NNMi – HP RAMS MPLS WAN Integration Configuration form contains the parameters for configuring communications between NNMi and RAMS. This form is available from the Integration Module Configuration workspace.

Note: Only NNMi users with the Administrator role can access the HP NNMi – HP RAMS MPLS WAN Integration Configuration form.

The following table lists the parameters for connecting to the NNMi management server:

Attribute	Description
NNMi Host	The fully qualified domain name of the NNMi management server. This field is pre-filled with host name that was used to access the NNMi console. Ensure that

Attribute	Description
	this value is the name that is returned by the `nnmofficialfqdn.ovpl -t` command run on the NNMi management server.
NNMi Port	The port for connecting to the NNMi console. This field is pre-filled with the port that the JBoss application server uses for communicating with the NNMi console, as specified in the following file: **Windows**: %NnmDataDir%`\conf\nnm\props\nms-local.properties` **UNIX**: `/var/opt/OV/conf/nnm/props/nms-local.properties` For non-SSL connections, use the value of JBoss.http.port, which is 80 or 8004 by default (depending on the presence of another web server when NNMi was installed). For SSL connections, use the value of JBoss.https.port, which is 443 by default.
NNMi User	The user name for connecting to the NNMi console. This user must have the NNMi Web Service Client role. For information on Configuring the NNMi user interface, see Roles Provided in NNMi.
NNMi Password	The password for the specified NNMi user.
RAMS MPLS WAN Rediscovery Interval (hours)	The time interval in hours to run the RAMS MPLS WAN discovery process.

To apply changes to the integration configuration, update the values on the HP NNMi – HP RAMS MPLS WAN Integration Configuration form, and then click **Submit**.

Related Topics:

"HP RAMS MPLS WAN (NNMi Advanced)"

"Using Route Analytics Management Software (RAMS) with NNMi Advanced" (on page 868)

Extending NNMi Capabilities

NNMi enables you to extend its capabilities in the following ways:

- You can integrate other programs into the console through the Actions menu. See "Configure Menu Item Basic Details" (on page 874).
- "Configure Custom Polling" (on page 929) so that NNMi monitors additional information using MIB expressions.
- "Add Custom Attributes to a Node or Interface Object" (on page 213)
- HP offers extended features, see "Purchase an HP Network Node Manager i Smart Plug-in" (on page 953).

Control the Actions Menu

NNMi enables you to configure the following menu items in the Actions menu:

- SNMP Line Graph Actions

  When you configure SNMP Line Graphs, you specify the graph appearance, including the MIB expression used to determine the values to be graphed. See "Configure SNMP Line Graph Actions" (on page 890) for more information.

- Launch Actions

  When you configure Launch Actions, you provide access to in-house tools, Web sites, or a variety of other resources. URLs are used to configure this powerful feature of NNMi. You must follow "W3C Rules for URLs" (on page 882). See "Configure Launch Actions" (on page 878) for more information. The syntax used to define the URL provides variables that can incorporate real-time data from the NNMi database. Click here for a list of choices:

- Java Actions provided by NNMi. See "Configure Java Actions" (on page 897)
- JavaScript Actions provided by NNMi. See "Configure JavaScript Actions" (on page 896).

You control where each Action appears in the Actions menu structure:

- Choose an Ordering number for each Action. See "Configure Menu Item Basic Details" (on page 874).
- Establish a nested structure of Menu items. See "Create Menu Nesting (in the Actions Menu)" (on page 873).
- Control when the Menu Item is available. See "Configure Menu Item Context Basic Details" (on page 877) and "Specify Optional Menu Item Enablement Filters" (on page 898).

Behavior of the Menu Items

If you do not assign an SNMP Line Graph or Launch Action to a particular menu item, that menu item never appears in the Actions menu.

Some Menu Item Actions require that a particular Object Type be selected in order for the menu item to be available. If the required Object Type is not selected, the color of that Action turns from black to gray to indicate it is unavailable.

If you deselect the ☐ Enabled attribute on the Menu Item form, that menu item never appears in the Actions menu.

Create Menu Nesting (in the Actions Menu)

As an NNMi administrator, you configure how menu items are nested beneath the Actions menu. Menus can then contain menu items or other (cascading) menus.

To configure a Menu, beneath which other menu items can be nested:

1. Navigate to the **Menu** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **User Interface Configuration**.
    c. Navigate to the **Menus** tab.
    d. Do one of the following:
        - To edit an existing menu, click the Open icon in the row representing the configuration you want to edit, and continue.
        - To create a new menu, click the New icon, and continue.
        - To delete a menu, select a row, and click the Delete icon.
2. Provide the required information to define the new Parent-level Menu Item (see basics table).
3. Click **Save and Close** to save and apply your changes.
4. Assign other menu items to appear beneath the Menu. The Menu label from step 2 is now available in the Parent Menu attribute drop-down list for Menu Items. See "Configure Menu Item Basic Details" (on page 874).
5. To test your changes to the Actions menu:
    a. If required, access a view or form that contains the appropriate object type.
    b. If required, select an object instance.
    c. Select the **Actions** menu.
    d. Verify your changes are working.

Configuration Settings for a Menu Nesting

Attribute	Description
Menu Label	The text string that appears in the Actions Menu Items and that (when clicked) displays a submenu with a group of Action Menu Items. Ensure that your menu label is unique and provides an accurate indication about the group of submenu items that are found beneath this menu entry. If you add two Menu Labels with the same text string but different Unique Keys, both can show up beneath the Actions menu. The maximum length is 255 characters. Alpha-numeric, spaces, and underline characters are allowed.
Unique Key	**Caution**: This value cannot be changed after you click Save. Used as a unique identifier when exporting and importing menu definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space

Attribute	Description
	convention when providing this value. It is also useful to include the Menu Label value as part of the unique key as shown in the following example: `com.<company_name>.nnm.menu.<menu_label>` Type a maximum of 80 characters. Alpha-numeric and period characters are allowed. No spaces are allowed.
Author	Indicates who created or last modified the Menu nesting object. See Author form for important information. **Caution**: If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. Click the Lookup icon and select Quick Find to access the list of existing Author values or click New to create one.
Parent Menu	Refine the nested location of this Actions menu item. Click the Lookup icon next to the Parent Menu attribute, and do one of the following: • Default behavior. Select **Actions** if you want this Menu to appear in the main list. • To select an existing parent-level menu item from the drop-down list (nesting the new menu item at a lower level in the menu structure), click the Quick Find icon. • To create a new parent-level menu item for nesting, click the New icon.
Ordering	A numeric value. NNMi checks for configuration settings in the order you define (lowest number first). NNMi uses the first match found to determine the placement of this menu item within the Actions menu The Ordering numbers are calculated separately for each submenu (group of nested Actions menu items). **Tip**: It is recommended that ordering numbers are incremented by 10s or 100s to provide flexibility over time.

Configure Menu Item Basic Details

The **Menu Items** tab of the **User Interface Configuration** option enables you to make changes or additions to the items available in the Actions menu. For example, you can configure SNMP Line Graphs and provide menu items that display in-house tools, Web sites, or access a variety of other resources.

To make changes or additions to the items available in the Actions menu:

1. Navigate to the **Menu Item** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select the **User Interface Configuration**.
    c. Navigate to the **Menu Items** tab.
    d. Do one of the following:

- To edit an existing Actions menu item, click the Open icon in the row representing the configuration you want to edit, and continue.
- To create a new Actions menu item, click the New icon, and continue.
- To delete an Actions menu item, select a row, and click the Delete icon.

2. Provide the required information to define the Action (see Basics tables).

   Caution: If you make changes to a Menu Item provided by NNMi, those changes are at risk of being overwritten in the future. See Author form for important information.

3. Provide the required Context details (see "Configure Menu Item Context Basic Details" (on page 877)).
4. Click **Save and Close** to save and apply your changes.
5. To test your changes to the Actions menu:
   a. If required, access a view or form that contains the appropriate object type.
   b. If required, select an object instance.
   c. Click the **Actions** menu.
   d. Verify your changes are working.

Basics

Attribute	Description
Menu Item Label	The text string that appears as the menu link. Ensure that your menu label is unique and accurately reflects the intended use of the Menu Item. If you add two Menu Items with the same Menu Item Label string, both show up beneath the Actions menu.
Unique Key	**Caution**: This value cannot be changed after you click the Save icon. Used as a unique identifier when exporting and importing Action definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the Menu Item Label value as part of the unique key as shown in the following example: `com.<company_name>.nnm.menu.item.<menu_item_label>` Type a maximum of 80 characters. Alpha-numeric and period characters are allowed. Spaces and underline characters are not allowed.
Author	Indicates who created or last modified the Menu Item. See Author form for important information. **Caution**: If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. Click the Lookup icon and select Quick Find to access the list of existing Author values or click New to create one.
Parent Menu	Specify where this action appears under the Actions menu: • Select **Actions** if you want this action to appear in the main list. (Default behavior.) • Select any existing parent-level menu item from the drop-down list.

Attribute	Description
	• Create a new parent-level menu item. See "Create Menu Nesting (in the Actions Menu)" (on page 873) for more information.
Ordering	Valid entries are 1 to 100. This attribute controls where your menu item shows up in the list of available actions (lowest number appears at the top of the group of Menu Items).
Prepend Separator	Used when a previous menu item exists (based on the Ordering number). Inserts a separator line above the menu item . Use this attribute to separate unrelated menu items.
Enabled	Use to temporarily disable a Menu Item configuration. If ☑ enabled, the Menu Item appears under the Actions menu. If ☐ disabled, the Menu Item does not appear.

Selection

Attribute	Description
Selection Type	*Optional*. The default is Single Selection. The Menu Item is always available if you specify **No Selection** or **Any Selection**. • If you specify any of the following, an error message appears when the user launches the action prior to selecting an appropriate object or objects: ▪ **Any Selection** means zero or more selections required. ▪ **Single Selection** means exactly one selection required. ▪ **Multiple Selection** means one or more selections required. • If you specify **No Selection**, the user must launch the action without selecting any objects. An error message appears if any objects are selected.
Max Selection Count	*Only valid if Selection Type = Any Selection or Multiple Selection*. Zero means unlimited. Specify the maximum number of objects the user can select prior to launching this action.

Restrictions

Attribute	Description
Path View Only	If ☑ enabled, your Launch Action appears *only* in the Path View window's Actions menu. See "Attributes per Object Type for Full URLs" (on page 883) for additional information about Path View Full URL configuration choices. If ☐ disabled, your Launch Action can appear in the menu of multiple views.
Requires NNM 6.x/7.x Management Station	If ☑ enabled, the action appears only when NNM 6.x/7.x management stations are configured to communicate with NNMi within your environment. (See "Configure Remote NNM 6.x and 7.x Management Stations".) If ☐ disabled, the action always appears.

Description

Attribute	Description
Description	*Optional*. Provide a description of your action. Your description is visible only within this configuration form. Type a maximum of 1024 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Configure Menu Item Context Basic Details

NNMi enables you to configure Actions menu items using the Menu Item Context form.

To make changes or additions to the items available in the Actions menu:

1. Navigate to the **User Interface Configuration** form,**Context** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **User Interface Configuration**.
   c. Navigate to the **Menu Items** tab.
   d. Do one of the following:
      - To edit an existing menu item, select a row, click the Open icon in the row representing the configuration you want to edit, and continue.
      - To create a new menu item, click the New icon, and continue.
      - To delete a menu item, select a row, and click the Delete icon.
   e. Provide the Basics for this action (see "Configure Menu Item Basic Details" (on page 874)).

      Caution: If the **Author** value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. See Author form for important information.
   f. Navigate to the **Menu Item Contexts** tab.
   g. Do one of the following:
      - To edit an existing Context configuration, select a row, click the Open icon, and continue.
      - To create a new Context configuration, click the New icon, and continue.
      - To delete a Context configuration, select a row, and click the Delete icon.
2. Provide the Basic details for this Context configuration. (see the Basics table).
3. *Optional*. Limit the use of the menu item to a subset of the chosen object-type instances by defining filter criteria (see "Specify Optional Menu Item Enablement Filters" (on page 898)
4. Click **Save and Close** to save and apply your changes.
5. To test your changes to the Actions menu:
   a. If required, access a view or form that contains the appropriate object type.
   b. If required, select an object instance.

c. Click the **Actions** menu.

d. Verify your changes are working.

Basics

Attribute	Description
Action	Click the Lookup icon next to the Action attribute and select one of the following: • Select Open to view the current Action configuration. There are four types of actions: ▪ "Configure Launch Actions" (on page 878) ▪ "Configure SNMP Line Graph Actions" (on page 890) ▪ "Configure JavaScript Actions" (on page 896) ▪ "Configure Java Actions" (on page 897) • Select **New SNMP Line Graph Action** to create a Line Graph. See "Configure SNMP Line Graph Actions" (on page 890) for more information. • Select **New Launch Action** to create an Launch Action menu item (access in-house tools, Web sites, or a variety of other resources). See "Configure Launch Actions" (on page 878) for more information.
Object Type	*Optional*. If you select **All Object Types**, your Graph Action or Launch Action is visible within the Actions menu in all views and forms. If you want your menu item to be available only within a view or form of a particular object type, select the desired Object Type from the drop-down menu. You can further limit the Action to a subset of object instances:
Required Role	Specify the lowest level role allowed to access this Action. All roles above the role you select can also access this URL action. See "Determine which NNMi Role to Assign" (on page 263).

Configure Launch Actions

The **Launch Actions** option enables you to configure Menu Items that are available from the Actions menu. These additional Actions menu items can access in-house tools, Web sites, or a variety of other resources.

To configure Launch Actions:

1. Navigate to the **Menu Item Contexts** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **User Interface Configuration**.
   c. Navigate to the **Menu Items** tab.
   d. Do one of the following:
      ○ To edit an existing Launch Action menu item, click the Open icon in the row representing the configuration you want to edit, and continue.

- To create a new Launch Action menu item, click the New icon, and continue.
- To delete an Launch Action menu item, select a row, and click the Delete icon.

e. Provide the Basic details for this Menu Item (see "Configure Menu Item Basic Details" (on page 874)).

Caution: If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. See Author form for important information.

f. Navigate to the **Menu Item Contexts** tab.

g. Do one of the following:

- To edit an existing Menu Item Context, select a row, click the Open icon, and continue.
- To create a new Menu Item Context, click the New icon, and continue.
- To delete a Menu Item Context, select a row, and click the Delete icon.

2. Locate the **Menu Item Action** attribute. Click the Lookup icon, and click the **New Launch Action** icon.
3. In the required syntax for the **Full URL** and configuration any additional configuration rules (see Launch Action Basics.
4. Click **Save and Close** to apply your changes and return to the Menu Item Context form.
5. Limit the use of the Action menu item to a subset of the chosen object-type instances by defining filter criteria (see "Specify Optional Menu Item Enablement Filters" (on page 898))
6. Click **Save and Close** to save and apply your changes.
7. To test your changes to the Actions menu:
   a. If required, access a view or form that contains the appropriate object type.
   b. If required, select an object instance.
   c. Click the **Actions** menu.
   d. Verify your changes are working.

   Troubleshooting Tip: If a specified attribute does not exist (for example, you made a mistake when typing the attribute's name), the attribute passes through literally (unresolved). For example:

   A node named "mynode" is selected, and the URL is:

   http://example.com?name=${name}&error=${error}

   The output would be:

   http://example.com?name=mynode&error=${error}

Launch Actions Basics

Attribute	Description
Name	Type a meaningful and descriptive name to help you remember the type of Action.
Full URL	Add one or more definitions for the actual URL syntax. Type the full URL specification. The URL must comply with "W3C Rules for URLs" (on page 882). Include any required machine name and port number. Include any required parameters.

Attribute	Description

- You can begin with either http:// or https://

  For an example, click here:

  http://<
  serverName
  >:<
  portNumber
  >/<
  application>?<*yourURLparameter1*>=${<*attribute*>}&<*yourURLparameter2*>=${<*attribute*>}

  `<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

  `<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

  Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

- You can also use other common URL protocols such as ftp://, mailto://, news://, or telnet://.

  Tip: If the application that your URL calls is installed on the NNMi management server, the syntax can be as follows:
  /<
  application
  >?<*yourURLparameter1*>=${<*Attribute*>}&<*yourURLparameter2*>=${<*Attribute*>}

- The list of available parameters changes depending on which limiting factors you configure. The **&** is used as the separator between the <*yourURLparameter*> and ${<*attribute*>} pairs.

  For an example, click here:

  http://example.com/nodeReport.jsp?myNode=${hostname}&mySnmpOid=${systemObjectId}

  Tip: If the application that your URL calls is installed on the NNMi management server, the syntax can be as follows:
  /<
  application
  >?<*yourURLparameter1*>=${<*Attribute*>}&<*yourURLparameter2*>=${<*Attribute*>}

  - "Attributes per Object Type for Full URLs" (on page 883) (Limits the availability of the Action to a subset of one object type.)
  - "Database Object Identifiers for Full URLs" (on page 889) (Limits the availability of the Action to *one specific instance* of an object.)
  - "Capability Attributes in Full URLs" (on page 886) (Limits the availability of the Action to a subset of objects.)
  - "Custom Attributes in Full URLs" (on page 887) (Limits the availability of the Action to a subset of objects.)
  - "Custom Incident Attributes (CIAs) in Full URLs" (on page 888) (Limits the availability of the Action to a subset of Incidents.)

See "Attributes per Object Type for Full URLs" (on page 883) for more information about the

Attribute	Description
	valid attributes per Object Type.

Attribute	Description
Enable Cumulative Launch	If ☑ enabled, any object attribute references in the Full URL are populated with values from all selected objects. The multiple values are separated by a comma character. For example, if the attribute is "name", the URL results would be "name1,name2,name3". If ☐ disabled, the action launches a separate web page instance for each selected object. See "Attributes per Object Type for Full URLs" (on page 883) for details about including object attributes in your Full URL.
Browser Width	*Optional*. When empty, the default browser settings are used. If the value is 1 or more, the browser is launched this number of pixels wide.
Browser Height	*Optional*. When empty, the default browser settings are used. If the value is 1 or more, the browser is launched this number of pixels high.
Add Browser Decorations	If ☑ enabled, the web browser toolbar and menus appear when a user launches your URL. If ☐ disabled, the web browser has no toolbar or menu when a user launches your URL.

W3C Rules for URLs

The World Wide Web Consortium (W3C) allows only ASCII characters in URLs.

When configuring URLs, the following characters are always allowed:

- Alpha-numeric (A-Z a-z 0-9)
- - (hyphen)
- . (period)
- _ (underline)
- ~ (tilde)

Depending on the browser and the context, some characters require special formatting with Percent Encoding. A small number of possible values are shown in the quick reference table below.

You can designate the space character several ways:

- + (works in all browsers, recommended because it is easiest to read)
- %20 (Percent Encoded value, works in all browsers)
- space character (works in the browsers supported by NNMi, but is not guaranteed to work in all browsers)

RFC 3986 Characters Reserved as Delimiters
(If not specifying a delimiter, use Percent-Encoding value)

Character	:	/	?	#	[	]	@	!	$
Percent Encoded	%3A	%2F	%3F	%23	%5B	%5D	%40	%21	%24
Character	&	'	(	)	*	+	,	;	=
Percent Encoded	%26	%27	%28	%29	%2A	%2B	%2C	%3B	%3D

Additional Commonly Used Characters and Their Percent Encoding

Character	space	%	<	>
Percent Encoded	%20 (or + allowed)	%25	%3C	%3E

Attributes per Object Type for Full URLs

There are a variety of methods to limit Launch Actions:

${<*attribute*>} values can be included in the Full URL syntax for each Object Type. For example:

- To limit the use of an Action available only from an Interface form, include ${ifAlias} as an ${<*attribute*>} value.
- To limit the use of an Action available only from a Node form, specify hostname: http://${hostname}:<*portNumber*>/<*application*>?attributeName1=${sysContact}&attributeName2=${sysName}

The following list includes the possible ${<*attributes*>} that can be included in the Full URL for each Object Type:

- **Interface**[parameter list for interface]

  ${capabilities[capability.key=<*UniqueKey*>].capability.key} <value of one specific Capability, see "Capability Attributes in Full URLs" (on page 886) for more information>
  ${customAttributes[name=<*yourAttrName*>].value} <value of the matching Custom Attribute, see "Custom Attributes in Full URLs" (on page 887) for more information>
  ${ifAlias} <value from the ifAlias attribute>
  ${ifDescr} <value from the ifDescription attribute>
  ${ifIndex} <value from the ifIndex attribute>
  ${ifName} <value from the ifName attribute>
  ${ifType.label} <value from the ifType attribute>
  ${journal.notes} <value from the Notes attribute>
  ${managementMode} <value from the Management Mode attribute>
  ${name} <value from the Name attribute>
  ${overallStatus.lastChange} <value from the Status Last Modified attribute>
  ${overallStatus.status} <value from the Status attribute>
  ${physicalAddress} <value from the Physical Address attribute>
  ${speed} <value from the ifSpeed attribute>
  Access an attribute on the related Node form:
  ${hostedOn.hostname} <value from the Hosted On attribute, source Node's Hostname attribute>
  ${hostedOn.name} <value from the source Node's Name attribute>
  Access an attribute on the related Node's Device Profile form:
  ${deviceProfile.devCategoryInterface}<value from the Category attribute's Label value>
  ${deviceProfile.devFamilyInterface}<value from the Family attribute's Label value>
  ${deviceProfile.devVendorInterface}<value from the Vendor attribute's Label>
  Access an attribute on the related SNMP Agent form:
  ${hostedOn.snmpAgent.id} <value from the source Node's SNMP Agent Id attribute>${hostedOn.snmpAgent.agentSettings.agentEnabled} <value from the source Node's SNMP Agent Enabled attribute>

- **Interface Group** [parameter list for interfaceGroup]

  ${name} <value from the Name attribute>
  ${notes} <value from the Notes attribute>

${nodeGroup.name} <value from the Node Group attribute>

- **Node** [parameter list for node]

  ${capabilities[capability.key=<*UniqueKey*>].capability.key} <value of one specific Capability, see "Capability Attributes in Full URLs" (on page 886)for more information>
  ${customAttributes[name=<*yourAttrName*>].value} <value of the matching Custom Attribute, see "Custom Attributes in Full URLs" (on page 887) for more information>
  ${hostname} <value from the Hostname attribute>
  ${journal.notes} <value from the Notes attribute>
  ${managementMode} <value from the Management Mode attribute>
  ${name} <value from the Name attribute>
  ${overallStatus.lastChange} <value from the Status Last Modified attribute>
  ${overallStatus.status} <value from the Status attribute>
  ${systemContact} <value from the System Contact attribute>
  ${systemDescription} <value from the System Description attribute>
  ${systemLocation} <value from the System Location attribute, the current value of the sysLocation MIB variable>
  ${systemName} <value from the System Name attribute>
  ${systemObjectId} <value from the System Object ID attribute>
  Access an attribute on the related Device Profile form:
  ${deviceProfile.deviceModel} <value from the Device Model attribute>
  ${deviceProfile.SNMPObjectID}<value from the SNMP Object ID attribute>
  ${deviceProfile.devCategoryNode}<value from the Category attribute's Label value>
  ${deviceProfile.devFamilyNode}<value from the Family attribute's Label value>
  ${deviceProfile.devVendorNode}<value from the Vendor attribute's Label value>
  Access an attribute on the related SNMP Agent form:
  ${snmpAgent.id} <value from the Id attribute>
  ${snmpAgent.agentSettings.managementAddress} <value from the Management Address attribute>
  ${snmpAgent.agentSettings.agentEnabled} <value from the Agent Enabled attribute>

- **Node Group** [parameter list for nodeGroup]

  ${name} <value from the Name attribute>
  ${notes} <value from the Notes attribute>
  ${overallStatus.lastChange} <value from the Status Last Modified attribute>
  ${overallStatus.status} <value from the Status attribute>

- **Incident** [parameter list for incident]

  ${category.label} <value from the Category attribute>
  ${cias[name=<*cia.name*>].value} <value of one specific Custom Incident Attribute, see "Custom Incident Attributes (CIAs) in Full URLs" (on page 888) for more information>
  ${duplicateCount} <value from the Duplicate Count attribute>
  ${family.label} <value from the Family attribute>
  ${formattedMessage} <value from the Message attribute>
  ${getAttrOrName(<*attribute*>)} <value of the specified attribute of the Node associated with the Incident (if the Node exists in the database) or the *sourceNodeName* attribute of the Incident (if the Node was deleted from the database or never existed in the database). For example, ${getAttrOrName(hostname)}>
  ${journal.notes} <value from the Notes attribute>
  ${lifecycleState.label} <value from the Lifecycle State attribute>
  ${nature} <value from the Correlation Nature attribute>
  ${nodeUuid} <value of the uuid for the Source Node, see "Database Object Identifiers for Full URLs" (on page 889)>
  ${nodeUuid.id} <value of the id for the Source Node, see "Database Object Identifiers for Full URLs"

(on page 889)>
${notes} <value from the Correlation Notes attribute>
${origin} <value from the Origin attribute>
${priority.label} <value from the Priority attribute>
${registration.created} <value from Created attribute>
${registration.modified} <value from the Last Modified attribute>
${severity} <value from the Severity attribute>
${sourceName} <value from Name attribute of the source object>
${sourceNodeName} <value from the Name attribute of the source object>
${sourceUuid} <value of the uuid for the Source Object, see "Database Object Identifiers for Full URLs" (on page 889)>
${sourceUuid.id} <value of the source object's id attribute>
Access an attribute on the related source object form:
${sourceUuid.name} <value of the source object's Name attribute>
Access an attribute on the related Node form:
${nodeUuid.hostname} <<value from the source Node's Hostname attribute or IP address if no hostname is available>
${nodeUuid.name} <value of the Name attribute of the Source Node>

- **Layer 2 Connection** [parameter list for layer2Connection]

  ${journal.notes} <value from the Notes attribute>
  ${name} <value from the Name attribute of the connection>
  ${source} <value of the Topology Source attribute, the protocol used to create the connection>

- **IP Address**[parameter list for address]

  ${capabilities[capability.key=<*UniqueKey*>].capability.key} <value of one specific Capability, see "Capability Attributes in Full URLs" (on page 886) for more information>
  ${journal.notes} <value from the Notes attribute>
  ${managementMode} <value from the Direct Management Mode attribute>
  ${name} <value from the Name attribute>
  ${overallStatus.lastChange} <value from the Status Last Modified attribute>
  ${overallStatus.status} <value from the Status attribute>
  ${prefixLength} <value from the Prefix Length attribute>
  ${value} <value from the Address attribute>

- **IPSubnet**[parameter list for subnet]

  ${journal.notes} <value from the Notes attribute>
  ${name} <value from the Name attribute>
  ${prefix} <value from the Prefix attribute>
  ${prefixLength} <value from the Prefix Length attribute>

- **Card**[parameter list for subnet]

  ${capabilities[*CAPABILITY_NAME*]} <value of one specific Capability, see "Capability Attributes in Full URLs" (on page 886) for more information>
  ${capabilities[capability.key=<*UniqueKey*>].capability.key} <value of one specific Capability, see "Capability Attributes in Full URLs" (on page 886) for more information>
  ${entityPhysicalIndex} <value from the Physical Index attribute>
  ${firmwareVersion} <value from the Firmware Version attribute>
  ${hardwareVersion} <value from the Hardware Version attribute>
  ${hostingCard.name} <value from the Hosted On Card attribute>
  ${index} <value from the Index attribute>
  ${journal.notes} <value from the Notes attribute>
  ${managementMode} <value from the Management Mode attribute>
  ${modelName} <value from the Model Name attribute>
  ${monitoredAttributes.operationalState} <value from the Operational State attribute>

${overallStatus.status} <value from the Status attribute>
${overallStatus.lastChange} <value from the Status Last Modified attribute>
${redundantGroup.name} <value from the Redundant Group attribute>
${serialNumber} ${softwareVersion} <value from the Serial Number attribute>
${type} <value from the Type attribute>

- **Port**[parameter list for subnet]

  ${associatedInterface.name} <value from the Associated Interface attribute>
  ${configuredDuplexSetting}<value from the Configured Duplex Setting attribute>
  ${index} <value from the Index attribute>
  ${journal.notes} <value from the Notes attribute>
  ${speed} <value from the Speed attribute>
  ${type} <value from the Type attribute>

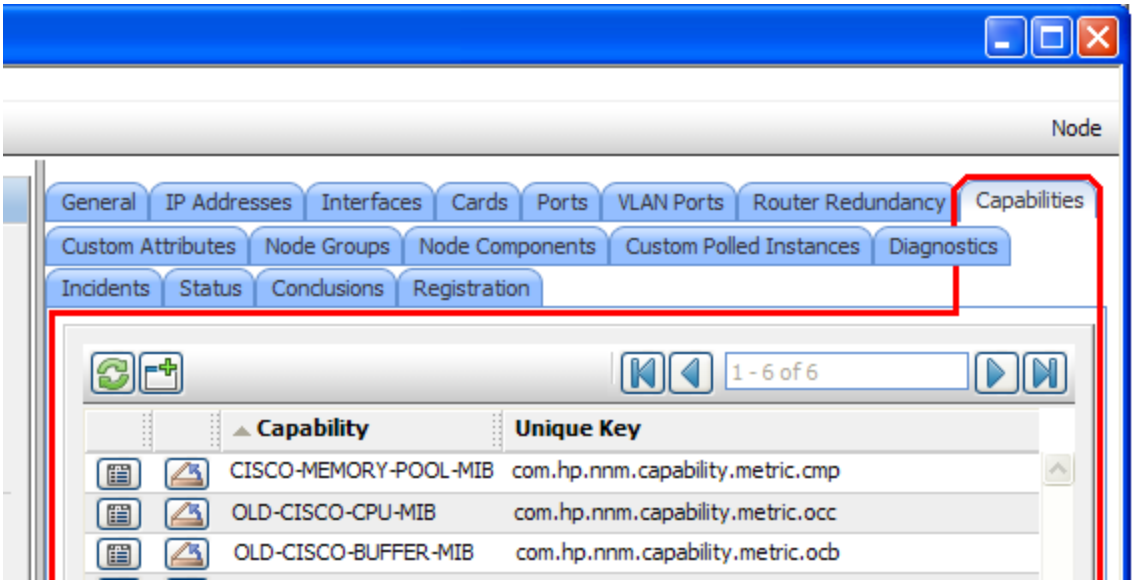
Capability Attributes in Full URLs

There are a variety of methods to limit Launch Actions:

NNMi node, interface, IP address, and card objects can have capability attributes:

Capabilities can be provided from HP Network Node Manager i Software Smart Plug-ins (iSPIs) or from integrations with other programs. See the documentation that came with any NNM iSPIs installed in your network environment.

To determine which group of capabilities are available for a specific object, navigate to a view for the object, select an instance of the object. Click the Open icon and navigate to the Capabilities tab. The items listed in the table are the Capabilities for that particular object instance. For example, the following illustration shows a Node form with three capability entries.



To pass Capability data within the Full URL, type (or copy and paste) the exact text string *from the object form, Capability tab*, **Unique Key** *attribute value*:

${**capabilities[capability.key=<*UniqueKeyValue*>].capability.key**}

Place the Capability into a location in the Full URL that enables the result your want:

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

http://<
serverName
>:<*portNumber*>/<*application*>?<*yourURLparameter1*>=${capabilities[capability.key=<*UniqueKey_*
1>].capability.key}&<*yourURLparameter2*>=${capabilities[capability.key=<*UniqueKey_2*>].capability.key}

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

Note: If the Capability that you request in the Full URL does not exist for the selected Node or Interface, the resulting URL passes an empty string.

Custom Attributes in Full URLs

There are a variety of methods to limit Launch Actions:

Custom Attributes enable an NNMi administrator to add information to the Node object or Interface object. Custom Attributes can also be set by external applications that have been integrated with NNMi.

The Node form: Custom Attributes tab and Incident form: Custom Attributes tab display a table view of any Custom Attributes that have been added to the selected object. See "Add Custom Attributes to a Node or Interface Object" (on page 213).

To determine which group of Custom Attributes are available for a specific Node or Interface, navigate to a Node view or Interface view, select an instance of the object, click the Open icon and navigate to the Custom Attributes tab. The items listed in the table are the Custom Attributes for that particular node or interface. For example, the following illustration shows a Node form with two Custom Attribute entries.

To pass Custom Attribute data within the Full URL, type (or copy and paste) the exact text string *from the Node or Interface form, Custom Attributes tab*:

${**customAttributes[name=<*yourAttrName*>].value**}

Place the Custom Attribute into a location in the Full URL that enables the result you want:

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

http://<
serverName
>:<
portNumber
>/<
application
>?
<*yourURLparameter1*>
=${customAttributes[value=<
yourAttrValue>].name}&<*yourURLparameter2*>=${customAttributes[name=<*yourAttrName*>].value}

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

<*portNumber*> = the port that the jboss application server uses for communicating with the NNMi console

- Example 1:

  mailto:${customAttributes[name=Admin].value}?subject=URGENT Action Required&body=${customAttributes[name=message].value}&${hostname} router needs attention.

  Resulting URL:

  mailto:JohnDoe@myCompany.com?subject=URGENT Action Required&body=Building-5:Floor-23.&cisco4.myCo.com router needs attention.

- Example 2:

  http://myCo.com/emailAdmin.jsp?name= ${hostname}&contact= ${customAttributes[name= Admin].value}&body= ${customAttributes[name=message].value}

  Resulting URL:

  http://myCo.com/emailAdmin.jsp?name= cisco4.myCo.com&contact= johnDoe@myCo.com&body= Building-5:Floor-23

Note: If the Custom Attribute that you request in the Full URL does not exist for the selected Node or Interface, the resulting URL passes an empty string.

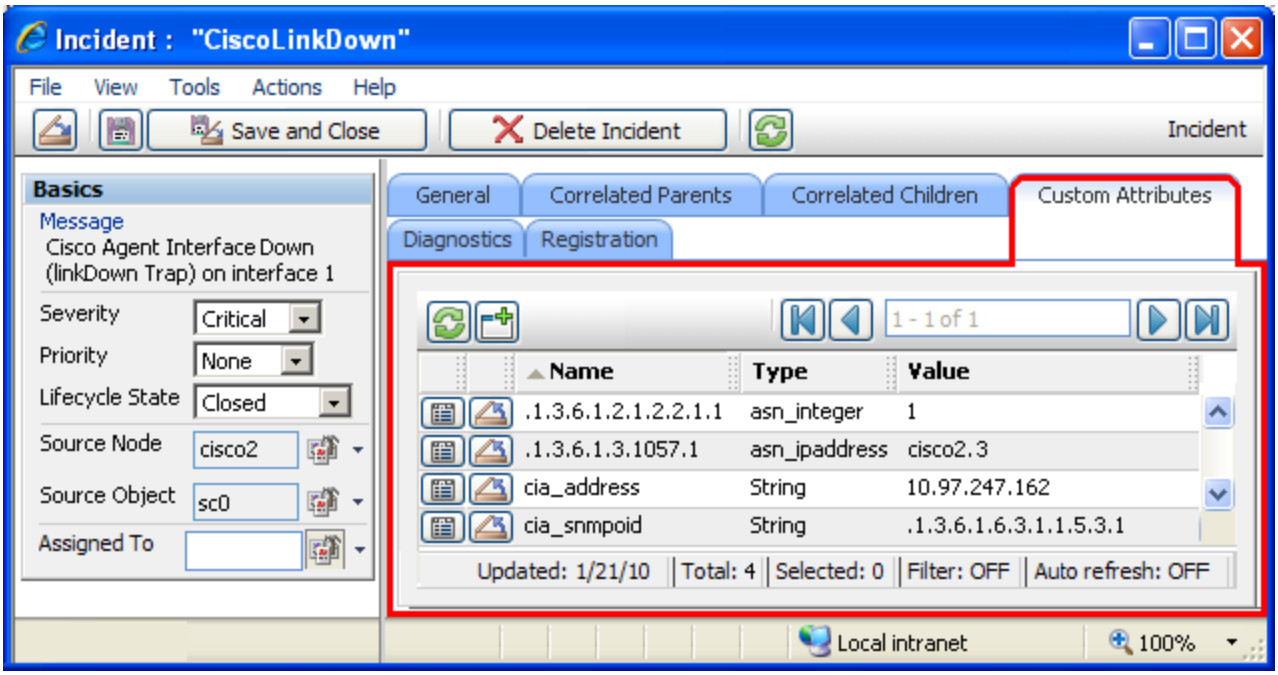
Custom Incident Attributes (CIAs) in Full URLs

There are a variety of methods to limit Launch Actions:

Custom Incident Attributes (CIAs) are used to provide the following types of information within incidents:

- SNMP trap varbinds identified by the Abstract Syntax Notation value, ASN.1 (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (for Administrators)" (on page 312).

To determine which group of CIAs is available for a specific incident-type (for example, CiscoLinkDown), navigate to an Incident view, select an instance of that incident-type, click the Open icon and navigate to the Custom Attributes tab. The items listed in the table are the CIAs for that particular incident-type. For example, all CiscoLinkDown incidents would have the same group of CIAs shown in the illustration below.



To pass CIA data within the Full URL, type (or copy and paste) the exact text string *from the Incident form, Custom Attribute tab*, **Name** *attribute value*:

${**cias[name=<*cia_name*>].value**}

Place the CIA into a location in the Full URL that enables the result your want:

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

http://<*serverName*>:<*portNumber*>/<*application*>?<*yourURLparameter1*>=${cias[name=<*cia_name_1*>].value}&<*yourURLparameter2*>=${cias[name=<*cia_name_2*>].value}

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

<*portNumber*> = the port that the jboss application server uses for communicating with the NNMi console

Note: If the CIA that you request in the Full URL does not exist for the selected Incident, the resulting URL passes an empty string.

Database Object Identifiers for Full URLs

There are a variety of methods to limit Launch Actions:

If you need the Full URL to identify one specific record in the NNMi database, and find that it is not possible to provide a unique set of attribute values that distinguish that object instance from all other similar object instances, the *database unique identifiers* are valuable parameters.

The ID and UUID attributes are valid for all object types. NNMi displays the ID and UUID attribute values on the object form's Registration tab:

- `${uuid}` Universally Unique Object Identifier -Unique across all databases.
- `${id}` Unique Object Identifier - Unique across the Entire NNMi Database.

For example, the user can select an Interface object in the console, and use this Action to open the form of the Node in which the Interface resides:

```
/nnm/launch?cmd=showForm&objtype=Node&objid=${hostedOn.id}
```

Path View Attributes for Full URLs

There are a variety of methods to limit Launch Actions:

If you specified that a Launch Action appears only in the Path View menu, additional parameters are available:

${pathStartNodeName} <value of the Source attribute>
${pathEndNodeName} <value of the Destination attribute>
${pathList} <list of objects traversed along the path, separated by commas>
${pathCalculationDate} <date and time the path was calculated>

Configure SNMP Line Graph Actions

The **User Interface Configuration** option enables you to configure Line Graphs that are available from the Actions menu. These graphs display real-time SNMP data for a selected node or interface. This feature is useful when you want to monitor a numeric MIB or MIB Expression value for a node or interface over a specified time interval. For example, you might want to monitor network traffic using the ifOutOctets MIB variable for a specified node. Or you might want to graph a MIB variable, such as Interface ifInOctets, to verify that a problem has been fixed for a specified interface before closing an incident.

Note: The node for which you want to display information must support SNMPv1, SNMPv2C, or SNMPv3.

NNMi provides a set of Line Graphs for nodes and for interfaces. See "s Provided by NNMi" for more information.

To configure additional Line Graphs:

1. Navigate to the **SNMP Line Graph** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **User Interface Configuration**.
    c. Navigate to the **Menu Items** tab.
    d. Do one of the following:
        - To edit an existing Graph Action, click the Open icon in the row representing the configuration you want to edit, and continue.

- To create a Graph Action, click the New icon, and continue.
- To delete a Graph Action, select a row, and click the Delete icon.

e. Provide the Basic details for this menu item (see "Configure Menu Item Basic Details" (on page 874)).

Caution: If the **Author** value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. See Author form for important information.

f. Navigate to the **Menu Item Contexts** tab.

g. Do one of the following:

- To edit an existing Menu Item Context, select a row, click the Open icon, and continue.
- To create a new Menu Item Context, click the New icon, and continue.
- To delete a Menu Item Context, select a row, and click the Delete icon.

2. Locate the **Menu Item Action** attribute. Click the Lookup icon next to the Action attribute, and click the **New SNMP Line Graph** icon.
3. Provide the graph details for this Graph (see Basics table).
4. Provide the MIB Specification information (see "MIB Specification Form" (on page 893)).
5. Click **Save and Close** to save and apply your changes and return to the Menu Item Context form.
6. Limit the use of the Action menu item:
   - By object type (see "Configure Menu Item Basic Details" (on page 874)).
   - By NNMi user role (see "Configure Menu Item Basic Details" (on page 874)).
   - By defining a filter for a subset of the chosen object-type instances (see "Specify Optional Menu Item Enablement Filters" (on page 898)).
7. Click **Save and Close** to save and apply your changes.

   To test your changes to the Actions menu:

   a. If required, access a view or form that contains the appropriate object type.
   b. If required, select an object instance.
   c. Click the **Actions** menu.
   d. Verify your Graph is working.

Basics

Attribute	Description
Graph Title	Type a meaningful and descriptive title to display above the graph. The maximum length is 255 characters. Alpha-numeric characters, spaces, and periods are allowed.
Y-axis Label	Enter the text string to describe the Y-axis data displayed. NNMi displays this label vertically along the left-side of the Y axis. The maximum length is 255 characters. Alpha-numeric characters, spaces, and

Attribute	Description
	periods are allowed. If you do not want to display a label for the Y-axis, leave this attribute blank.
Number of Lines	Specify the number of lines that will be initially displayed on the graph. An operator can display additional lines when viewing the Line Graph. To use the Default value specified in the User Interface Configuration, leave this attribute value blank. The default value that NNMi provides is 15.
Maximum Time Range (Hours)	The maximum time period in hours in which to retain the Line Graph data point sets. When the Maximum Time Range number is reached, NNMi discards the oldest data point sets so that it can display the most recent data for the time range you specify. For example, if you enter 24 hours, when 24 hours has passed, NNMi removes data starting with the initial data point set so that it can display data for the most recent 24-hour interval. Enter a decimal number indicating the maximum number of hours in which to retain the Line Graph data. If you specify 0 (zero), NNMi determines the best setting for the Maximum Time Range based on the Poling Interval specified.
Update Interval (Seconds)	The Update Interval in seconds to be used for collecting data to be displayed on the graph. **Note**: You can change the Update Interval for the current session when NNMi displays the graph. To use the Default value specified in the User Interface Configuration, leave this attribute value blank.
Fast Start	Select Fast Start when you want to increase the initial Polling Interval so that the initial data appears more quickly on the graph. When you select this option, NNMi increases the initial Polling Interval and then gradually decreases the Polling Interval until it reaches the Polling Interval configured for the graph.
Enable Cumulative Launch	If ☑ enabled, any object attribute references in the Full URL are populated with values from all selected objects. The multiple values are separated by a comma character. For example, if the attribute is "name", the URL results would be "name1,name2,name3". If ☐ disabled, the action launches a separate web page instance for each selected object. See "Attributes per Object Type for Full URLs" (on page 883) for details about including object attributes in your Full URL.
Browser Width	*Optional*. When empty, the default browser settings are used. If the value is 1 or more, the browser is launched with this number of pixels wide.
Browser Height	*Optional*. When empty, the default browser settings are used. If the value is 1 or more, the browser is launched with this number of pixels high.

Attribute	Description
Add Browser Decorations	If ☑ enabled, the web browser toolbar and menus appear when a user launches your URL. If ☐ disabled, the web browser has no toolbar or menu when a user launches your URL.

MIB Specification Form

The MIB Specification form enables you to indicate the following:

- The label to be displayed for each line that appears in the Line Graph Legend.
- The MIB Expression NNMi uses to gather the data shown in the graph.

To specify the Line Label and MIB Expression for an SNMP Line Graph Action:

1. Navigate to the **MIB Specification** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **User Interface Configuration**.
    c. Navigate to the **Menu Items** tab.
    d. Do one of the following:
        - To create a new menu, click the New icon.
        - To edit a menu, click the Open icon in the row representing the configuration you want to edit.
    e. Navigate to the **Menu Item Contexts** tab.
    f. Do one of the following:
        - To create a new Context configuration, click the New icon.
        - To edit an existing Context configuration, click the Open icon in the row representing the configuration you want to edit.
    g. In the **Menu Item Context** form, locate the **Action** attribute.
    h. Click the Lookup icon next to the **Action** attribute, and do one of the following:
        - To create a new Line Graph, click the **New SNMP Line Graph Action** icon.
        - To edit the Line Graph associated with the Graph Action name displayed, click the Open icon in the row representing the configuration you want to edit.
    i. Provide the Basic details for this Graph Action (see the "Configure SNMP Line Graph Actions" (on page 890)).
    j. Navigate to the **MIB Specifications** tab.
    k. Do one of the following:

- To create a new MIB Specification configuration , click the New icon.
- To edit an existing MIB Specification configuration, click the Open icon in the row representing the configuration you want to edit.

2. Provide the Basic details for this MIB Specification configuration. (see the MIB Specification Basics table).
3. Click **Save and Close** to save and apply your changes.

MIB Specification Basics

Attribute	Description
Line Label	Enter the label that you want to be displayed for each line that appears in the Graph legend. Type a maximum of 50 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. **Note**: When graphing multiple instances, the <*instance_string*> is appended to this value. See "MIB Expression Form (Line Graph)" (on page 920) for more information.
MIB Expression	Use this attribute to specify the MIB information that you want NNMi to poll. A MIB expression must include at least one MIB Variable. It can also include one or more of the following: • Constant • Arithmetic operator (+, -, *, /) If the MIB Expression does not include any arithmetic operators, valid types for any MIB Variable in the MIB Expression include the following: • Integer • Unsigned Integer • Octet String • Counter • Counter32 • Counter64 • Gauge • Time_Ticks If the MIB Expression contains any constants or arithmetic operators, the MIB Expression must evaluate to a numeric type. When evaluating MIB expressions that include MIB variables of type Counter (Counter, Counter32, Counter64, or Time_Ticks), NNMi evaluates the MIB Variable using the difference in value between the most recent poll and the poll before it. If you want NNMi to calculate a rate over time, divide the MIB Expression by sysUptime. For example: `(((ifInOctets+ifOutOctets)*8/ifSpeed)*100)/sysUpTime` **Note**: If you use a MIB variable of type Counter (Counter, Counter32, Counter64, or Time_Ticks) in the MIB Expression, NNMi automatically collects sysUpTime values if sysUpTime is not already in the MIB Expression. NNMi uses the sysUptime value to

Attribute	Description
	detect a system reboot. Any time a system reboot is detected, NNMi cannot determine the difference in values between polls for any Counter MIB variable and therefore does not calculate the MIB Expression for that poll. **If you select a MIB Variable from an Interface Table** to include in the MIB Expression, note the following: • When evaluating MIB Expressions that include MIB variables of type Counter or Counter64, NNMi requests both the high capacity and low capacity counter variable for any interface instance. If the high capacity counter (Counter 64) is enabled for any given interface instance, NNMi uses the high capacity counter. • When evaluating MIB Expressions that include MIB variables of type Counter32, NNMi requests only the low capacity counter information for any interface instance. You create a MIB Expression by using the MIB Expression form. To access the MIB Expression form, click the Lookup icon and do one of the following: • Select **Quick Find** to select an existing MIB expression. • Select **Open** to edit the current MIB expression. • Select **New** to create a MIB expression. See "MIB Expression Form (Line Graph)" (on page 920) for information about using the MIB Expression form.

Attribute	Description
Instance Selection Algorithm	Used to specify how you want NNMi to handle instance discovery for Line Graphs that display multiple instances. Possible values are: • **All** - Use when you want NNMi to graph each instance of the object selected by the user. Note the following: ▪ When a node is selected, NNMi discovers all instances for that node, including the interfaces. When an interface is selected, NNMi graphs all selected interfaces. ▪ NNMi ignores any values entered in the Instance List attribute. ▪ When the Line Graph menu item is launched, NNMI populates `${snmpAgent.id}` and ${hostedOn.snmpAgent.id} with the ID values from the selected objects. The multiple values are separated by a comma character. ▪ NNMi displays a maximum of 100 instances. NNMi determines which 100 instances to display using the following calculation: 100 instances/(number of nodes selected)*(number of MIB expressions for the Action) • **Instance List** - Use when you want to specify the instances to be included in the Line Graph **Note**: You must specify the Instance List when using this option.
Instance List (Comma Separated)	Used to identify the instances to be graphed for an object. If your Menu Item Context is Node and you want to specify which nodes should be included on this Line Graph, enter the instance number for each of the node instances to be included on this Line Graph. For example, to graph CPU values, enter the instance number representing each CPU on the node, separated by commas. If your Menu Item Context is an Interface and you want to specify the selected interfaces to be included on this Line Graph, enter the `${ifIndex}` value for each of the interfaces to be included in this Line Graph. See "Attributes per Object Type for Full URLs" (on page 883) for more information about ${<*attribute*>} values.

Configure JavaScript Actions

NNMi provides a set menu items implemented with JavaScript. These menu items do not generate a new browser window.

Do not make any changes to these other than:

- Hide the menu item from the NNMi console (see the ☐ Enabled attribute "Configure Menu Item Basic Details" (on page 874)).
- Change the Required Role setting (see "Configure Menu Item Context Basic Details" (on page 877)).

Caution: If you make changes to a Menu Item provided by NNMi, those changes are at risk of being overwritten in the future. See Author form for important information.

To view the settings for a JavaScript Action:

1. Navigate to the **JavaScript Action** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **User Interface Configuration**.
    c. Navigate to the **Menu Items** tab.
    d. Click the Open icon in the row representing the configuration you want to edit.
    e. In the Menu Item form, navigate to the **Menu Item Contexts** tab.
    f. Click the Open icon in the row representing the configuration you want to edit.
    g. Locate the **Menu Item Action** attribute. Click the Lookup icon next to the Action attribute, and click the **Open** icon.
2. For an explanation of the JavaScript Action attributes, see the Basics table.
3. Click **Close**to return to the Menu Item Context form.

   Tip: You can change the Required Role setting here.
4. Click **Close** to return to the Menu Item form.

   Tip: You can change the Enabled attribute here.

Java Script Basics

Attribute	Description
Name	The name that NNMi assigned to this menu item.
JavaScript	The actual JavaScript code that NNMi provided for this menu item

Configure Java Actions

NNMi provides a set of menu items implemented as Java classes. These menu items launch a new browser window.

Do not make any changes to these other than:

- Hide the menu item from the NNMi console (see the Enabled attribute "Configure Menu Item Basic Details" (on page 874)).
- Change the Required Role setting (see "Configure Menu Item Context Basic Details" (on page 877)).
- Change the width / height of the displayed window (see below).
- Enable / disable browser decorations for the displayed browser window (see below).

Caution: If you make changes to a Menu Item provided by NNMi, those changes are at risk of being overwritten in the future. See Author form for important information.

To view the settings for a Java Action:

1. Navigate to the **Java Action** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **User Interface Configuration**.
    c. Navigate to the **Menu Items** tab.

d. Click the Open icon in the row representing the configuration you want to edit.

e. In the Menu Item form, navigate to the **Menu Item Contexts** tab.

f. Click the Open icon in the row representing the configuration you want to edit.

g. Locate the **Menu Item Action** attribute. Click the Lookup icon next to the Action attribute, and click the **Open** icon.

2. For an explanation of the Java Action attributes, see the Basics table.

3. Click **Close**to return to the Menu Item Context form.

   Tip: You can change the Required Role setting here.

4. Click **Close** to return to the Menu Item form.

   Tip: You can change the Enabled attribute here.

Java Action Basics

Attribute	Description
Name	The name that NNMi assigned to this menu item.
Java Class	The Java class that NNMi implemented for this menu item
Parameters (Optional)	The list of any parameters used by the menu item.
Browser Width	Indicates the expected behavior of the menu item.
Browser Height	Indicates the expected behavior of the menu item.
Add Browser Decorations	Indicates the expected behavior of the menu item.

Specify Optional Menu Item Enablement Filters

If your SNMP Graph Action or Launch Action applies to Nodes, Interfaces, or Incidents, you can use the Filters Editor to create expressions that further define the context in which this Graph Action or Launch Action is available within NNMi. A Menu Enablement Filter limits the use of the Menu Item which uses this context. In the Actions menu, the Menu Item is disabled unless the selected object passes this filter.

Design complex Filters on paper as a Boolean expression first to minimize errors when entering your expressions using this Filters editor.

To create any Filter expressions:

1. Navigate to the **Menu Item Context** form.

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Select **User Interface Configuration**

   c. Navigate to the **Menu Items** tab.

   d. Do one of the following:

- To create a Menu Item definition, click the New icon.
- To edit a Menu Item definition, click the Open icon in the row representing the configuration you want to edit.

e. Navigate to the **Menu Item Contexts** tab.

f. Do one of the following:

- To create a Context configuration, click the New icon.
- To edit a Context configuration, select a row, click the Open icon.

2. Navigate to the **Menu Item Enablement Filter** tab.
3. Establish the appropriate settings for the filter you want to create. (See the Custom Filter Editor Components table.)

   When creating any filters, note the following:

   - Boolean Attributes begin with "is" and must contain the value `true` or `false`.
   - Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together. View the expression displayed under **Filter String** to see the logic of the expression as it is created.
   - The AND Boolean Operators must contain at least two expressions.
   - The placement of your cursor and the subsequent text that is selected is important when performing operations using the Additional Filters Editor. For example, you append to or replace, the expression that is selected.
   - The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators. See "Add Boolean Operators in the Additional Filters Editor" (on page 193) for more information.
   - You can drag any of the following items to a new location in the Filter String:
     - Filter Editor Options: AND, OR, NOT, EXISTS, NOT EXISTS
     - Filter Expression (Attribute, Operator and Value)
   - When moving items in the Filter String, note the following:
     - Click the item you want to move before dragging it to a new location.
     - As you drag a selected item, an underline indicates the target location.
     - If you are moving the selection up, NNMi places the item above the target location.
     - If you are moving the selection down, NNMi places the item below the target location.
     - If you attempt to move the selection to an invalid target location, NNMi displays an error message.
4. Click **Save and Close** to save and apply your changes.

Custom Filter Editor Components

Attribute	Description
Attribute	The attribute name NNMi should use as the filter criteria. Possible attributes include the following: **Note**: Boolean Attributes begins with "is" and must contain the value `true` or `false`. Interface [click here for a list of attribute values] **Unique Keys from the Interface Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Interface Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) Node [click here for a list of attribute values] **Values from the Basics information on the Node Form:** • isSnmpNode (Agent Enabled) • isSnmpInterface (Agent Enabled) • isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: • sysOidNode (System Object ID) • sysOidInterface (System Object ID) **Unique Keys from the Node Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Node Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) **Values from the Basics information on the Device Profile Form:** • devVendorNode (Device Vendor) • devFamilyNode (Device Family) • devVendorInterface (Device Vendor) • devFamilyInterface (Device Family) Incident [click here for a list of attribute values] **Values from the Incident Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value)
Operator	The standard query language (SQL) operations to be used for the search. Valid operators are described below. **Note**: Only the `is null` Operator returns null values in its search.

Attribute	Description
Attribute	The attribute name NNMi should use as the filter criteria. Possible attributes include the following: **Note**: Boolean Attributes begins with "is" and must contain the value `true` or `false`. Interface [click here for a list of attribute values] **Unique Keys from the Interface Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Interface Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) Node [click here for a list of attribute values] **Values from the Basics information on the Node Form:** • isSnmpNode (Agent Enabled) • isSnmpInterface (Agent Enabled) • isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: • sysOidNode (System Object ID) • sysOidInterface (System Object ID) **Unique Keys from the Node Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Node Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) **Values from the Basics information on the Device Profile Form:** • devVendorNode (Device Vendor) • devFamilyNode (Device Family) • devVendorInterface (Device Vendor) • devFamilyInterface (Device Family) Incident [click here for a list of attribute values] **Values from the Incident Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value)
	• **=** Finds all values equal to the value specified. • **!=** Finds all values not equal to the value specified. • **<** Finds all values less than the value specified.

Attribute	Description
Attribute	The attribute name NNMi should use as the filter criteria. Possible attributes include the following: **Note**: Boolean Attributes begins with "is" and must contain the value `true` or `false`. Interface [click here for a list of attribute values] **Unique Keys from the Interface Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Interface Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) Node [click here for a list of attribute values] **Values from the Basics information on the Node Form:** • isSnmpNode (Agent Enabled) • isSnmpInterface (Agent Enabled) • isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: • sysOidNode (System Object ID) • sysOidInterface (System Object ID) **Unique Keys from the Node Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Node Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) **Values from the Basics information on the Device Profile Form:** • devVendorNode (Device Vendor) • devFamilyNode (Device Family) • devVendorInterface (Device Vendor) • devFamilyInterface (Device Family) Incident [click here for a list of attribute values] **Values from the Incident Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value)
	• **<=** Finds all values less than or equal to the value specified. • **>** Finds all values greater than the value specified. • **>=** Finds all values greater than or equal to the value specified.

Attribute	Description
Attribute	The attribute name NNMi should use as the filter criteria. Possible attributes include the following: **Note**: Boolean Attributes begins with "is" and must contain the value `true` or `false`. Interface [click here for a list of attribute values] **Unique Keys from the Interface Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Interface Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) Node [click here for a list of attribute values] **Values from the Basics information on the Node Form:** • isSnmpNode (Agent Enabled) • isSnmpInterface (Agent Enabled) • isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: • sysOidNode (System Object ID) • sysOidInterface (System Object ID) **Unique Keys from the Node Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Node Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) **Values from the Basics information on the Device Profile Form:** • devVendorNode (Device Vendor) • devFamilyNode (Device Family) • devVendorInterface (Device Vendor) • devFamilyInterface (Device Family) Incident [click here for a list of attribute values] **Values from the Incident Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value)
	• **between** Finds all values equal to and between the two values specified. • **in** Searches for a match in at least one of a series of values. • **is not null** Searches for all non-blank values.

Attribute	Description
Attribute	The attribute name NNMi should use as the filter criteria. Possible attributes include the following: **Note**: Boolean Attributes begins with "is" and must contain the value `true` or `false`. Interface [click here for a list of attribute values] **Unique Keys from the Interface Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Interface Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) Node [click here for a list of attribute values] **Values from the Basics information on the Node Form:** • isSnmpNode (Agent Enabled) • isSnmpInterface (Agent Enabled) • isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: • sysOidNode (System Object ID) • sysOidInterface (System Object ID) **Unique Keys from the Node Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Node Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) **Values from the Basics information on the Device Profile Form:** • devVendorNode (Device Vendor) • devFamilyNode (Device Family) • devVendorInterface (Device Vendor) • devFamilyInterface (Device Family) Incident [click here for a list of attribute values] **Values from the Incident Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value)
	• **is null** Searches for all blank values. • **like** Enables you to find matches using the asterisk (*) and question mark (?) as wildcard characters. Question mark character means "any single character of any type at this

Attribute	Description
Attribute	The attribute name NNMi should use as the filter criteria. Possible attributes include the following: **Note**: Boolean Attributes begins with "is" and must contain the value `true` or `false`. Interface [click here for a list of attribute values] **Unique Keys from the Interface Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Interface Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) Node [click here for a list of attribute values] **Values from the Basics information on the Node Form:** • isSnmpNode (Agent Enabled) • isSnmpInterface (Agent Enabled) • isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: • sysOidNode (System Object ID) • sysOidInterface (System Object ID) **Unique Keys from the Node Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Node Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) **Values from the Basics information on the Device Profile Form:** • devVendorNode (Device Vendor) • devFamilyNode (Device Family) • devVendorInterface (Device Vendor) • devFamilyInterface (Device Family) Incident [click here for a list of attribute values] **Values from the Incident Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value)
	location". Asterisk character means "any number of characters of any type at this location". • **not between** Finds all values except those between the two values specified. • **not in** Finds all values except those included in the list of values.

Attribute	Description
Attribute	The attribute name NNMi should use as the filter criteria. Possible attributes include the following: **Note**: Boolean Attributes begins with "is" and must contain the value `true` or `false`. Interface [click here for a list of attribute values] **Unique Keys from the Interface Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Interface Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) Node [click here for a list of attribute values] **Values from the Basics information on the Node Form:** • isSnmpNode (Agent Enabled) • isSnmpInterface (Agent Enabled) • isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: • sysOidNode (System Object ID) • sysOidInterface (System Object ID) **Unique Keys from the Node Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Node Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) **Values from the Basics information on the Device Profile Form:** • devVendorNode (Device Vendor) • devFamilyNode (Device Family) • devVendorInterface (Device Vendor) • devFamilyInterface (Device Family) Incident [click here for a list of attribute values] **Values from the Incident Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value)
	• **not like** Finds all values except those included in the value specified. The not like operator enables you to use the asterisk (*) and question mark (?) as wildcard characters.

Attribute	Description
Attribute	The attribute name NNMi should use as the filter criteria. Possible attributes include the following: **Note**: Boolean Attributes begins with "is" and must contain the value `true` or `false`. Interface [click here for a list of attribute values] **Unique Keys from the Interface Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Interface Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) Node [click here for a list of attribute values] **Values from the Basics information on the Node Form:** • isSnmpNode (Agent Enabled) • isSnmpInterface (Agent Enabled) • isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: • sysOidNode (System Object ID) • sysOidInterface (System Object ID) **Unique Keys from the Node Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Node Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) **Values from the Basics information on the Device Profile Form:** • devVendorNode (Device Vendor) • devFamilyNode (Device Family) • devVendorInterface (Device Vendor) • devFamilyInterface (Device Family) Incident [click here for a list of attribute values] **Values from the Incident Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value)
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive.

Attribute	Description
Attribute	The attribute name NNMi should use as the filter criteria. Possible attributes include the following: **Note**: Boolean Attributes begins with "is" and must contain the value `true` or `false`. Interface [click here for a list of attribute values] **Unique Keys from the Interface Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Interface Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) Node [click here for a list of attribute values] **Values from the Basics information on the Node Form:** • isSnmpNode (Agent Enabled) • isSnmpInterface (Agent Enabled) • isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: • sysOidNode (System Object ID) • sysOidInterface (System Object ID) **Unique Keys from the Node Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Node Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) **Values from the Basics information on the Device Profile Form:** • devVendorNode (Device Vendor) • devFamilyNode (Device Family) • devVendorInterface (Device Vendor) • devFamilyInterface (Device Family) Incident [click here for a list of attribute values] **Values from the Incident Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value)
	Note: When entering the Boolean values, `true` or `false`, use all lowercase. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed.

Attribute	Description
Attribute	The attribute name NNMi should use as the filter criteria. Possible attributes include the following: **Note**: Boolean Attributes begins with "is" and must contain the value `true` or `false`. Interface [click here for a list of attribute values] **Unique Keys from the Interface Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Interface Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) Node [click here for a list of attribute values] **Values from the Basics information on the Node Form:** • isSnmpNode (Agent Enabled) • isSnmpInterface (Agent Enabled) • isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: • sysOidNode (System Object ID) • sysOidInterface (System Object ID) **Unique Keys from the Node Form: Capabilities Tab**: • capability (Unique Key of the Capability) **Values from the Node Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value) **Values from the Basics information on the Device Profile Form:** • devVendorNode (Device Vendor) • devFamilyNode (Device Family) • devVendorInterface (Device Vendor) • devFamilyInterface (Device Family) Incident [click here for a list of attribute values] **Values from the Incident Form: Custom Attributes Tab**: • customAttrName (Custom Attribute Name) • customAttrValue (Custom Attribute Value)
	• The `in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude nodes with values that pass the expression that immediately follows the NOT. For example, when evaluating the following Filter String, NNMi includes all nodes that have SNMP enabled and excludes any nodes with a Device Profile attribute value that includes **Cisco** as the Vendor value: `(isSysName = true AND NOT (devVendorNode=Cisco))`
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider nodes that do not have any Capabilities or Custom Attributes when evaluating the Filter String. For example, when evaluating the following Filter String, NNMi includes all nodes with a Capability having the Unique Value of **com.hp.nnm.capability.metric.cse**and ImportantRouters value of **Building5**: `(capability = com.hp.nnm.capability.metric.cse AND EXISTS (customAttrName=ImportantRouters AND customAttrValue=Building5))`
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the objects that match the expression that follows the NOT EXISTS. For example, when evaluating the following Filter String, NNMi includes all nodes with a hostname that includes **router**, followed by any number of characters, followed by **hp.com**and excludes any nodes with a Custom Attribute named **ImportantRouters** with the value of **Building5**: `(hostname like router*.hp.com AND NOT EXISTS (customAttrName=ImportantRouters AND customAttrValue=Building5))`
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Examine Available MIBs and MIB Variables

NNMi enables you to take a proactive approach to network management by using MIB Expressions to specify additional information that NNMi should poll. See "Configure MIB Expressions" (on page 919) for more information.

Note: The MIB files that define the MIB variables included in the MIB Expression that you want NNMi to poll must be loaded on the NNMi management server.

Before you create your MIB Expressions, examine the available MIBs and MIB variables using the following methods:

- "Loaded MIBs View" (on page 912)
- MIB Browser

Determine the MIBs Supported for a Node (for Administrators)

Tip: See MIB Browser Keyboard Navigation for a description of the keyboard navigation you can use in the MIB Browser.

To view the MIBs (Management Information Base) supported by a selected Node, use the **Tools → List Supported MIBs** option from the MIB Browser. This option is useful when configuring MIB Expressions so that you can determine the MIBs and associated MIB variables available for use. It can also help you to determine what additional MIBs you might want to load on the NNMi management server. See "Loaded MIBs View" (on page 912) for more information.

Note: You can also select a Node or Incident from an Inventory view and use the **Actions→List Supported MIBs** option to view the MIBs supported for a Node without accessing the MIB Browser.

To view the MIBs supported for a Node from the MIB Browser:

1. Do one of the following:
   - Select **Tools → MIB Browser**.
   - Open a MIB Variable, MIB Notifcation, Table Index, or Enumerated Value form from the Loaded MIBs view and select **Actions → Browse MIB**.

     Note: You can also access the MIB Browser from a Node or Incident view or form. See Determine a Node's MIB Variable Values (MIB Browser) for more information.

     NNMi displays the MIB Browser.
2. In the **Node** attribute, enter the Node Name or IP address of the Node for which you want to view the MIB Variable values.
3. *Optional*. In the **Community String** attribute, enter a valid SNMPv1 *read community string* for the Node. If you provide a *read community string*, NNMi uses SNMPv1 communication protocol. If none is provided, NNMi checks the NNMi database for any configured SNMP read community string for that Node.
4. Select **Tools → List Supported MIBs**.

   NNMi displays the textual representation of the OID (Object Identifier) for each MIB that is supported by the Node's SNMP Agent. NNMi also lists any MIB tables that reside in each MIB. When displaying the list, NNMi indicates the MIBs that are supported, but not loaded.

   To access the MIB form for a supported MIB, click the MIB name, for example `ENTITY-MIB`.

To view the MIBs supported for a Node without accessing the MIB Browser:

1. Do one of the following:
   - Select a Node from an Inventory view.
   - Select an Incident from an Incident view.
   - Open a Node or Incident form.

     Note: NNMi uses the Incident's Source Node as the selected Node.
2. Select **Actions** → **List Supported MIBs**.

   NNMi displays the textual representation of the OID (Object Identifier) for each MIB that is supported by the Node's SNMP Agent. NNMi also lists any MIB tables that reside in each MIB. When displaying the list, NNMi indicates the MIBs that are supported, but not loaded.

   To access the MIB form for a supported MIB, click the MIB name, for example `ENTITY-MIB`.

Related Topics

"Display a MIB Table (MIB Browser)"

"Display a MIB File's Contents (Administrators)" (on page 915)

"Determine the MIB Variables Supported for a Node (for Administrators)" (on page 913)

Check SNMP Support for a Node (MIB Browser)

Find an Entry in the MIB Browser Output

Export MIB Browser Output

Copy MIB Browser Output (MIB Browser)

Print SNMP MIB Browser Output (MIB Browser)

View the MIBs Loaded on the NNMi Management Server

To view the MIBs stored in the NNMi Database, do either of the following:

Use the "Loaded MIBs View" (on page 912)

Use the nnmloadmib.ovpl command. Also see"Load MIBs from the Command Line" (on page 919) for more information.

After a MIB is loaded from the NNMi management server, you can also view the MIB when creating a Custom Poller Collection. See "Create a Policy" (on page 947) for more information.

Loaded MIBs View

Use the **Loaded MIBs** option of the **Configuration** workspace to determine the MIBs loaded on the NNMi management server.

Note: The MIB containing a variable you want to use in a MIB Expression must be loaded on the NNMi management server.

To view the MIBs Loaded on the NNMi management server:

1. Navigate to the **Configuration** workspace.
2. Select **Loaded MIBs**.

NNMi displays the Name of the MIB and the relative MIB file name for each of the MIBs available.

See "Load MIBs" (on page 916) for information about how to load MIBs.

See nnmloadmibs.ovpl for information about how to unload MIBs.

MIB Form

The MIB form provides details about the selected MIB that is loaded on the NNMi management server.

For information about each tab:

MIB Basics Attributes

Attribute	Description
Name	Name from the DEFINITIONS clause in the MIB file.
MIB File	Relative location of the MIB file, starting with `snmp-mibs` or `user-snmp-mibs`.

.

Determine the MIB Variables Supported for a Node (for Administrators)

To view the MIB Variables supported for a node, use the **Tools** → **MIB Browser** menu option. This option is useful for determining the following:

- What is possible to graph for a specified node. For example, you might want to determine whether a Node supports MIB Variables in the RMON2-MIB so that you can decide whether to configure a Line Graph using one or more of the RMON2-MIB's Variables.
- How often the MIB Variable values change. This information helps to determine whether a Line Graph would be a useful tool for monitoring the MIB Variable's values.
- Determine MIB Variables to use for Custom Polling. See "Configure Custom Polling" (on page 929) for more information.

To view the MIB Variables supported for a specified node:

1. Do one of the following:
   - Select **Tools** → **MIB Browser**.
   - Open a MIB Variable form from the Loaded MIBs view and select **Actions** → **Browse MIB**.

     Note: You can also access the MIB Browser from a Node or Incident view or form. See Determine a Node's MIB Variable Values for more information.

     NNMi displays the MIB Browser.
2. In the **Node** attribute, enter the Node Name or IP address of the Node for which you want to view the MIB Variable values.
3. *Optional*. In the **Community String** attribute, enter a valid SNMPv1 *read community string* for the Node. If you provide a *read community string*, NNMi uses SNMPv1 communication protocol. If none is provided, NNMi checks the NNMi database for any configured SNMP read community string for that Node.
4. If you accessed the MIB Browser from a MIB Variable form, NNMi provides the OID attribute value using the selected MIB Variable. Otherwise, NNMi provides `mib-2.system` (the root of the MIB-2

branch). To change the OID:

- Type additional numbers or text strings for a specific MIB-2 area.
- Replace the default OID numbers to issue an SNMP getNext request for another area in the Internet MIB tree.

Click here for more information. Note the following:

- You can obtain a MIB Variable OID value using the **Loaded MIBs** view. See "Loaded MIBs View" (on page 912) for more information.
- The OID must begin with a dot (.).
- NNMi automatically completes the OID name for you. The name you begin to enter must be one of the following:
  - A valid textual or numeric OID.
  - An OID alias provided by NNMi. To obtain a list of valid OID aliases, use the **Tools** → **OID Aliases** option from the SNMP MIB Browser.

    Note: If you begin with a space, NNMi displays the list of possible values.

5. Press `Enter`. NNMi does the following:

   If the associated MIB file is loaded on the NNMi management server, NNMi displays the textual representation of the OID (Object Identifier) for the MIB Variable as well as its associated value. If the associated MIB file is not loaded on the NNMi management server, NNMi displays the numeric representation of the OID.

   Note: You can also click the Walk button to display MIB Browser output.
6. To expand a MIB or MIB Variable entry, do one of the following:
   - Click the Expand icon that precedes the entry you want to expand.
   - Click **Expand All**.
7. To collapse a MIB or MIB Variable entry, do one of the following:
   - Click the Collapse icon that precedes the entry you want to collapse.
   - Click **Collapse All**.
8. To stop gathering the MIB Variable information before NNMi reaches the end of the Internet MIB tree, click the Stop button.

   When all available MIB Variable values are displayed, the Stop button is disabled.

Related Topics

"Display a MIB Table (MIB Browser)"

"Display a MIB File's Contents (Administrators)" (on page 915)

"Determine the MIBs Supported for a Node (for Administrators)" (on page 911)

Check SNMP Support for a Node (MIB Browser)

Find an Entry in the MIB Browser Output

Export SNMP MIB Browser Output

Copy MIB Browser Output (MIB Browser)

Print MIB Browser Output (MIB Browser)

Display a MIB File's Contents (Administrators)

To view a MIB file's contents, use the **Actions** → **Display MIB File** menu option. This option is useful for examining the contents of an entire MIB file to determine all of the MIB Variables and associated values contained in a MIB. You might want to use this option to familiarize yourself with a MIB before creating a MIB expression.

To view a MIB file's contents:

1. Do one of the following:
   - Select a MIB from the **Configuration** → **Loaded MIBs** view.
   - Open a MIB form.

     Note: You can also access a MIB form using **Tools** → **List Supported MIBs** from the MIB Browser.
   - Open a MIB Variable form.
2. Select **Actions** → **Display MIB File**.

   NNMi displays the MIB file's contents.

You can also view a MIB file's contents from the MIB Browser. See Display a MIB File's Contents (MIB Browser) for more information.

Related Topics

"Display a MIB Table (SNMP MIB Browser)"

"Determine the MIBs Supported for a Node (for Administrators)" (on page 911)

"Determine the MIB Variables Supported for a Node (for Administrators)" (on page 913)

Check SNMP Support for a Node (MIB Browser)

Find an Entry in the MIB Browser Output

Export MIB Browser Output

Copy MIB Browser Output (MIB Browser)

Print MIB Browser Output (MIB Browser)

Upload MIB Files from the Console

To upload local MIBs files so they are available to load into the NNMi database, use the **Tools** → **Upload Local MIB File** menu. The **Upload Local MIB File** enables you to browse to a vendor's site and upload the specified MIB for subsequent MIB loading.

See "Load MIBs from the Console" (on page 916) for more information about loading MIBs from the NNMi console.

You can also use the **Tools** → **Load MIB..** to load any incident configuration associated with the MIB. See "Load SNMP Trap Incident Configurations using the Console" (on page 430) for more information.

To upload local MIB files from the NNMi console:

1. Do one of the following:
   a. Navigate to the MIB view or form. For example, select **Configuration** → **Loaded MIBs**.
   b. Navigate to the MIB Variable view or form. For example, select **Inventory** → **MIB Variables**.
2. Select **Tools** → **Upload Local MIB File**.
3. Click **Browse** to locate the MIB file you want to upload.
4. Click **Upload** to upload the MIB file to the following directory:

   Windows:
   `%NnmDataDir%\shared\nnm\user-snmp-mibs`

   UNIX:
   `/var/opt/OV/shared/nnm/user-snmp-mibs`
5. NNMi displays the following information:
   - The full path to the MIB file.
   - Instructions for loading and listing MIB files.

Load MIBs

NNMi requires that a MIB be loaded on the NNMi management server before you can specify that you want to poll a MIB Expression that includes one of that MIB's variables.

You might also want to load MIBs when creating Graphs.

NNMi automatically stores a set of MIB files on the NNMi management server during installation. These files are located in the following directory:

Windows

`%NnmInstallDir%\misc\nnm\snmp-mibs`

UNIX

`/opt/OV/misc/nnm/snmp-mibs`

To view the MIBs loaded on the NNMi management server, see "View the MIBs Loaded on the NNMi Management Server" (on page 912)

To load additional MIBs, do one or more of the following:

- In the console, load MIBs
- "In a command line, load MIBs"

If you are using MIBs with Custom Poller, see "Enable or Disable Custom Poller" (on page 930) and "Create a Custom Poller Collection" (on page 930)

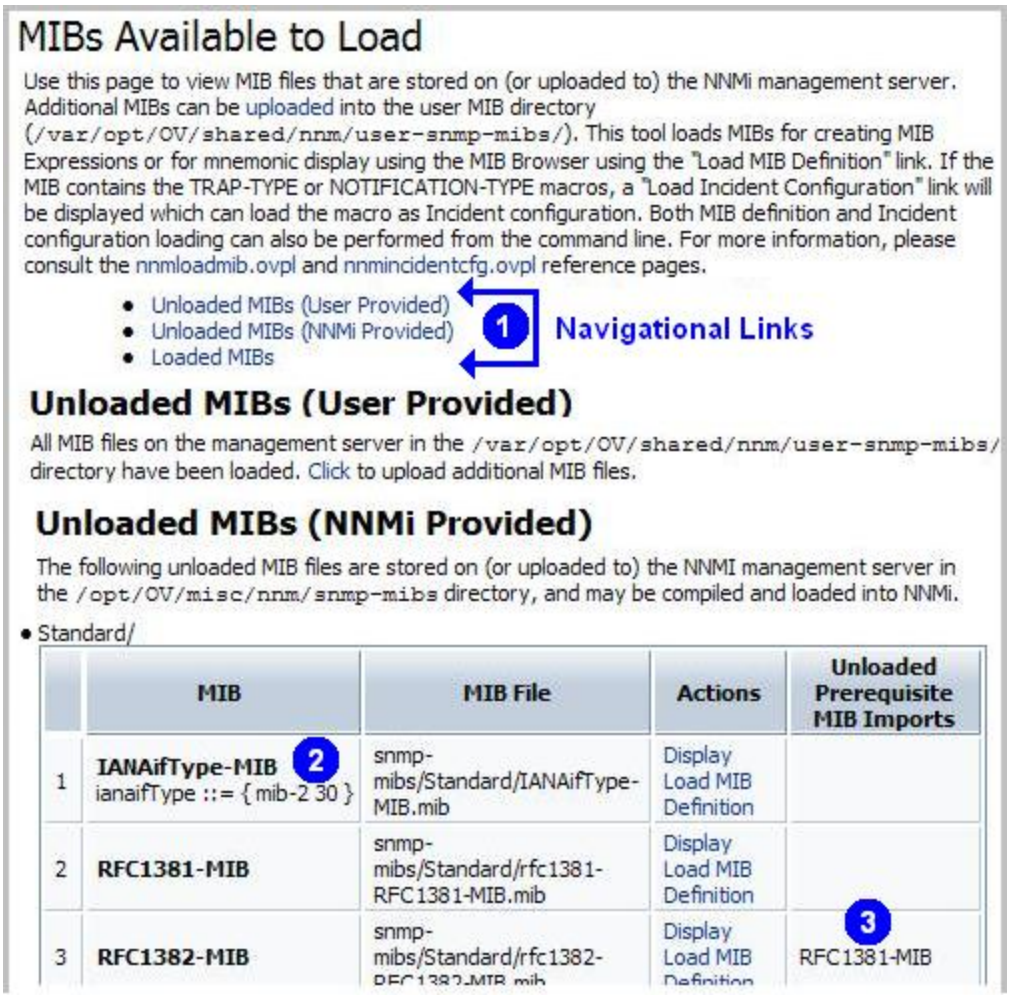
If you are using MIBs to create Graphs, see "Configure SNMP Line Graph Actions" (on page 890)

To unload a MIB file, see the nnmloadmib.ovpl command.

Load MIBs from the Console

To load additional MIBs from the NNMi console, use the **Tools** → **Load MIB** menu. The **Load MIBs** option also enables you to view the MIBs that are available to load.

Click here for details.



Load MIBs Web Page

Feature	Description
1	If any MIBs are stored on the NNMi management server (available for loading or already loaded), click the link to display the appropriate table.
2	If any MIB includes a conforming SNMPv2 SMI *MODULE-IDENTITY*, a text string displays that describes the MODULE-IDENTITY. For example, the MODULE-IDENTITY in row 1 is `ianaifType`
3	This column displays any MIBs that are "prerequisites" for the listed MIB, and still need to be manually loaded before you can load the listed MIB. These dependencies are gathered from the MIB's IMPORTS statement. For example:

```
RFC1382-MIB DEFINITIONS ::= BEGIN

IMPORTS
      Counter, Gauge, TimeTicks
              FROM RFC1155-SMI
      OBJECT-TYPE
              FROM RFC-1212
      DisplayString, transmission
              FROM RFC1213-MIB
      TRAP-TYPE
              FROM RFC-1215
      EntryStatus
              FROM RFC1271-MIB
      PositiveInteger,
      IfIndexType
              FROM RFC1381-MIB;
```

Note: Currently, the NNMi console does not display any `TEXTUAL-CONVENTION` entries from loaded MIBs.

You can also use the **Tools** → **Load MIB** to load any incident configuration associated with the MIB. See "Load SNMP Trap Incident Configurations using the Console" (on page 430) for more information.

To load additional MIBs from the NNMi console:

1. Do one of the following:
   a. Navigate to the MIB view or form. For example, Select **Configuration** → **Loaded MIBs**.
   b. Navigate to the MIB Variable view or form. For example, Select **Inventory** → **MIB Variables**.
2. Select **Tools** → **Load MIB**.

   NNMi displays the following information:

   - Unloaded MIBs (user provided) that are stored on the NNMi management server and that were provided by the NNMi administrator.
   - Unloaded MIBs (NNMi provided) that NNMi has stored on the NNMi management server during installation.
   - MIBs that are loaded in the NNMi database.

     See Click here for more details for more information.
3. Navigate to the Unloaded MIB view of interest. For example, **Unloaded MIBs (NNMi Provided)**.
4. In the MIB colum, find the MIB you want to load. For example, **RFC1381-MIB**.
5. To view the MIB before loading, in the Actions column, click **Display**.

   NNMi displays the MIB file contents.
6. To load the MIB, in the Actions column, click **Load MIB Definition**.

   NNMi displays the MIB File load progress including the following:

- The MIB root object identification (OID) number
- Number of MIBs, MIB variables, enumerated values, table indices, and parent/child hierarchies created
- Whether the MIB successfully loaded

To upload a local MIB file so that it is stored on the NNMi management server and available for loading, see "Upload MIB Files from the Console" (on page 915).

To unload a MIB file, see the nnmloadmib.ovpl command.

Load MIBs from the Command Line

To load additional MIBs from the command line, use the nnmloadmib.ovpl command.

Note: You can also use the nnmloadmib.ovpl command with the `-list` option to view the list of MIBs stored in the NNMi database.

To load additional MIBs from the command line:

1. Locate the MIB file you want to use.

   Note: You can use the device vendor's website to locate the MIBs available for your devices.

2. Copy the MIB file to the location of your choice. In the example used in the next step, the MIB file is copied to a `/temp` directory.

3. Use the nnmloadmib.ovpl command to load the MIB on the NNMi management server.

   For example, to load the HOST-RESOURCES-MIB that was copied to the `/temp` directory, you would enter a command similar to the following:

   If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access" (on page 275) for more information.

   ```
   nnmloadmib.ovpl -u <NNMiadminUsername> -p <NNMiadminPassword> -load
   /temp/HostResources.mib
   ```

If you are using MIBs to create MIB Expressions for Custom Poller, also see "Enable or Disable Custom Poller" (on page 930)and "Create a Custom Poller Collection" (on page 930)

If you are using MIBs to create Graphs, see "Configure SNMP Line Graph Actions" (on page 890)

To unload a MIB, use the nnmloadmib.ovpl command.

Configure MIB Expressions

NNMi enables you to take a proactive approach to network management by using SNMP MIB Expressions to specify additional information that NNMi should poll. After you create the MIB Expression, you can display this information in Graphs or use it with the NNMi Custom Poller feature.

To specify a MIB Expression, provide the required information within one of the following contexts:

"MIB Expressions Form (Custom Poller)" (on page 936)

"MIB Expression Form (Line Graph)" (on page 920)

See "MIB Expressions in Full URLs" for more information about using MIB Expressions in Graphs.

See "Configure Custom Polling" (on page 929) for more information about using MIB Expressions with Custom Poller.

The MIB Expressions view in the Configuration workspace includes the MIB Expressions provided by NNMi. See "MIB Expressions View" (on page 920) for more information.

MIB Expressions View

Use the MIB Expressions view to determine the MIB Expressions available for use. You can use MIB Expressions when configuring Custom Poller and Graph Actions. See and "MIB Expressions Form (Custom Poller)" (on page 936) and "MIB Expression Form (Line Graph)" (on page 920) for more information.

Note: All MIB Expressions provided by NNMi use the Author value **HP Network Node Manager.**

To view the MIB Expressions available:

1. Navigate to the **Configuration** workspace.
2. Select **MIB Expressions**.

   The columns in this table view show the Name, Author, and Description for each available MIB Expression.

MIB Expression Form (Line Graph)

You can access the MIB Expression form in either of the following ways:

- From the **Configuration** workspace, **MIB Expressions** view.
- From the **Configuration** workspace, **Custom Poller Configuration** form.
- From the **MIB Specification** form. (Used when configuring SNMP Graph actions.)

When you want to create a MIB Expression to be used in Line Graphs, use the **MIB Expressions** view. See "Configure MIB Expressions" (on page 919) for more information about configuring Line Graph. See "MIB Expressions Form (Custom Poller)" (on page 936) for more information about using the **Custom Poller Configuration** form.

Note: You can re-use any MIB Expression that you create for NNMi Line Graphs or for Custom Poller. Use "MIB Expressions View" (on page 920) to see a list of the available MIB Expressions. Use the "Loaded MIBs View" (on page 912) to see a list of the MIBs loaded on the NNMi management server.

To create a MIB Expression using the MIB Expression form:

1. From the workspace navigation panel, select the **Configuration** workspace.
2. Select **MIB Expressions**.
3. Do one of the following:
   - To create a MIB Expression, click the **New** icon, and continue.
   - To edit a MIB Expression, select a row, click the **Open** icon, and continue.
4. Provide the required basic settings (see the MIB Expression Basic Attributes table).
5. *Only for Multiple Instance MIB Expressions*. Line Graphs that display multiple instances use the

following syntax for the line label that appears in the Graph legend:

<node_name> <Line_Label>.<instance_string>

In this instance, *<Line_Label>* is the Line Label value specified when using the MIB Specification for.

Use the **Instance Display Configuration** section of the MIB Expression form to specify the configuration for the *<instance_string>* values (see the Instance Display Configuration table).

See "Use the MIB Expression Editor (Line Graph)" (on page 925) for more information about multiple instance MIB Expressions.

6. Click **Save and Close**.
7. To test your MIB Expression, select **Actions** → **Graph MIB Expression**. See "Test a MIB Expression (Line Graph)" (on page 924) for more information.

   Note: You must save the MIB Expression before you use **Actions** → **Graph MIB Expression**.

MIB Expression Basic Attributes

Attribute	Description
Unique Key	Used as a unique identifier when exporting and importing MIB Expression definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the label value as part of the unique key as shown in the following example: `com.<your_company_name>.nnm.mibexp.<mib_expression_name>` The maximum length is 80 characters. **Note**: Unlike the Unique Key attributes associated with other objects, you can change the MIB Expression configuration's Unique Key value at any time.
Name	The name you want to use for the MIB information being polled. This name can be the same name as a MIB Variable used in the MIB Expression, or you can enter a name of your choice. Type a maximum of 50 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. No spaces are allowed.
Author	Indicates who created or last modified the MIB Expression. See Author form for important information. **Caution**: If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. Click the Lookup icon and select Quick Find to access the list of existing Author values or click New to create one.
Expression	Click the icon to access the MIB Expression editor. See "Use the MIB Expression Editor (Line Graph)" (on page 925) for information about using the MIB Expression editor. **Note**: The MIB containing the variable must be loaded on the NNMi management server.
Display numeric MIB OIDs in the Expression	Enables you to display the MIB object identifier (OID) rather than the MIB variable name in the MIB Expression. Select **Display MIB OIDs in the Expression** to replace any MIB variable name with the MIB OID value in the MIB Expression.

Attribute	Description
	Clear **Display MIB OIDs in the Expression** ☐ to display the MIB variable names rather than the MIB OIDs within the MIB Expression.
Description	NNMi provides the Description attribute to help you further identify the current MIB Expression configuration. Use the description field to provide additional information that you would like to store about the current MIB expression configuration. Type a maximum of 2000 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Instance Display Configuration

Attribute	Description
Conversion Algorithm	Used to determine the format in which the instance portion (<*instance_string*>) of the line label appears in the Line Graph legend. Line labels in a Line Graph use the following syntax: <*node_name*> <*Line_Label*>.<*instance_string*> In this instance, <*Line_Label*> is the Line Label value specified when using the MIB Specification form. Possible Conversion Algorithms are: • **Numeric** - Use this option to display the instance number returned by the SNMP query as the <*instance_string*> value. This format is useful when no meaningful name is available in the MIB. For example, Line Graphs that display CPU information might use this format. • **MIB Variable** - Use this option to display the value that is stored in the MIB variable you specify. To obtain each MIB variable value, NNMi appends the instance number to the MIB variable specified. The result from the SNMP query is converted to a text string and displayed as the <*instance_string*> value of the line label in the Line Graph legend. • **Alphabetic** - Use this option to display information for legacy Cisco Arrow Point load balancers. When using this algorithm, each instance number returned by the SNMP query is treated as a set of ASCII characters instead of numbers. For example, the instance 101.120.97.109.112.108.101 would be displayed as 'example' in the <*instance_string*> of the line label.. • **Interface Name** - Use this option to display the interface name as the <*instance_string*> in the Line Graph legend. **Note**: The Interface Name option is only valid when an IfIndex value is returned as the instance number. The ifIndex value is then used to determine the Interface Name value. • **Interface Name Indirect** - Use this option to display the Interface Name value obtained from an indirect reference in the MIB table. For example, if the MIB variable you specify resides in an RMON MIB table, use this algorithm. **Note:** The **Interface Name Indirect** option is only valid when an OID is returned from an SNMP query that, when queried, returns an ifIndex value. The ifIndex value is then used to determine the Interface Name value using the "Interface Name"

Attribute	Description
	algorithm.
Display Variable	Select the MIB variable you want to display as the <*instance_string*> value in the line label of the Line Graph legend. NNMi uses the Conversion Algorithm you specify to determine the <*instance_string*> format.

Attribute	Description
Display Filter	Enter a valid regular expression that specifies the pattern you want NNMi to match when determining the values to display in the <*instance_string*> value of each line label. **Note**: NNMi uses the syntax defined for java regular expressions (java.util.regex Pattern class). NNMi finds the first character sequence that matches the Display Filter expression. If NNMi does not find a match for the Display Filter, it returns the Display Variable name. For example, if you have several interfaces with an ifDescr set to "FastEthernet" followed by a unique set of numbers for each interface (such as FastEthernet0/1, FastEthernet0/2, FastEthernet0/3, and so on), you can use the following Display Filter to display "Ethernet" followed by the unique set of numbers: `(Ethernet.*[0-9]+){1}` In the example, the following matches occur: • `Ethernet` matches Ethernet • The `.*` matches 0/ • The `[0-9]+` matches any sequence of numbers • The `{1}` specifies to match the expression exactly one time

Test a MIB Expression (Line Graph)

The Actions menu enables you to test the results of a MIB Expression using a Line Graph.

Note: You must save the MIB Expression before you use **Actions** → **Graph MIB Expression**.

To graph the results for a MIB Expression:

1. Navigate to the **MIB Expression** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **MIB Expressions**.

      Note: You can also access the MIB Expression form when creating Line Graphs and when creating Custom Poller Collections. See "MIB Expression Form (Line Graph)" (on page 920) and "MIB Expressions Form (Custom Poller)" (on page 936) for more information.
2. Click ☑ to select the MIB Expression you want to graph.
3. Select **Actions** → **Graph MIB Expression**.

   The dialog for selecting a node appears.
4. Click the **Lookup** icon and select **Quick Find**.
5. Select the node you want to use to test your MIB Expression results.

   NNMi displays a Line Graph using the selected node and calculating the results for the MIB Expression you selected.

   Note the following:

- When evaluating MIB Expressions that include MIB variables of type Counter or Counter64, NNMi requests both the high capacity and low capacity counter variable for any interface instance. If the high capacity counter (Counter 64) is enabled for any given interface instance, NNMi uses the high capacity counter.
- When evaluating MIB Expressions that include MIB variables of type Counter32, NNMi requests only the low capacity counter information for any interface instance.

Use the MIB Expression Editor (Line Graph)

Use the MIB Expression Editor to specify the MIB Variables and any Constant values or arithmetic operators you want to include in your MIB Expression.

For example, disk utilization could be calculated and polled using a MIB Expression similar to the following:

(`hrStorageAllocationUnits` * `hrStorageSize`) / (`hrStorageUsed` * `hrStorageAllocation`)

See the MIB Expression Editor Options table for a description of each of the MIB Expression Editor options.

When using the MIB Expression Editor, note the following:

- As a general guideline, begin by writing out the MIB Expression. Then in the MIB Expression Editor, begin creating your MIB Expression by selecting your arithmetic operators (+, -, *, or /) from the outermost parenthesis to the innermost parenthesis. Each time you specify an arithmetic operator (+, -, *, or /), NNMi creates a set of parenthesis to specify the ordering of the mathematical calculation.
- When adding arithmetic operators (+, -, *, or /) to a MIB Expression, first click to select the location in the MIB Expression at which you want to add the arithmetic operator.
- Click to select the arithmetic operator (for example +) in the MIB Expression, before selecting the MIB variable or Constant value that you want to add, subtract, multiply or divide.

  You can also use the following key bindings to add arithmetic operators:

  - ALT+ (plus button)
  - ALT- (minus button)
  - ALT/ (divide button)
  - ALT* (multiply button)
- NNMi inserts arithmetic operators, MIB Expressions, and Constant values from the left to right.
- To replace an arithmetic operator use the <> (Change Operator) button (see table).
- To replace a MIB Variable or Constant value, click to select the existing value in the MIB Expression and then select the new MIB variable or enter the new Constant value.

  Note: You can replace a MIB Variable with another MIB Variable or with a Constant value. You can replace a Constant value with a MIB Variable or Constant value.
- You can drag any of the following items to a new location in the MIB Expression:
  - MIB variable
  - Constant value
  - An operation, such as **(ifInOctets + ifOutOctets)**

Click here for more information about moving items in the MIB Expression to a new location.

When moving items in the MIB Expression, note the following:

- To move an arithmetic operation (for example, **(ifInOctets + ifOutOctets)**), click the arithmetic operator before dragging it to a new location.
- To move a MIB Variable or Constant Value, click the MIB Variable or Constant Value you want to move before dragging it to a new location.
- If you are moving the selected item to the right, NNMi places the item to the right of the new location.
- If you are moving the selected item to the left, NNMi places the item to the left of the new location.
- As you drag a selected item, an underline indicates the current target location.
- If you drag a selected item past the outermost parenthesis, it is deleted. If desired, you can re-enter the value in the new location.

MIB Expression Example

To create a MIB Expression that calculates the percentage of available bandwidth on a half-duplex interface, you might create the following MIB Expression:

((((ifInOctets + ifOutOctets) * 8) / ifSpeed) * 100)

See Creating a MIB Expression Animation for an animated demonstration of creating the MIB Expression above.

Click here for a step-by-step textual example of creating the same MIB Expression:

To create a MIB Expression that calculates the percentage of available bandwidth on a half-duplex interface, you might create the following MIB Expression:

((((ifInOctets + ifOutOctets) * 8) / ifSpeed) * 100)

To create the expression above, begin by specifying each arithmetic operator from the outermost parenthesis to the innermost parenthesis.

1. Click * (multiply).
2. Click / (divide).

   Now that you have multiple entries in your MIB Expression, you need to click to select the location in the MIB Expression to which you want to add each remaining arithmetic operators.
3. In the MIB Expression, click / (divide).

   The divide (/) arithmetic operator and its surrounding parenthesis should appear highlighted. Because NNMi inserts arithmetic operators, MIB variables, and Constant values from left to right, selecting / (divide) places the next arithmetic operator to the left of the divide arithmetic operator.
4. Click * (multiply).

   The multiply (*) arithmetic operator and its parenthesis should appear to the left of the divide arithmetic operator you previously selected.
5. In the MIB Expression, click the leftmost * (multiply).

   The multiply (*) arithmetic operator and its surrounding parenthesis should appear highlighted.
6. Click + (add).

   The add (+) arithmetic operator and its parenthesis should appear to the left of the multiply (*) arithmetic operator you previously selected.

Now that you have specified the arithmetic operators, you are ready to add the MIB variables and Constant values. Begin by selecting the arithmetic operator in the MIB Expression to which you will add MIB variables, Constant values, or both. We will begin with the leftmost arithmetic operation.

Note: As you add your MIB variables or Constant values, make sure you first select the corresponding arithmetic operator within the MIB Expression.

7. In the MIB Expression attribute, click + (add).
8. Select the ifInOctets MIB Variable:
   a. Click to open the MIB Variable Tree.
   b. Navigate to **ifInOctets**.
   c. Select **ifInOctets**.
   d. Click **OK**.

      The ifInOctets MIB variable should appear to the left of the add (+) arithmetic operator.
9. Select the ifOutOctets MIB Variable:
   a. Click to open the MIB Variable Tree.
   b. Navigate to **ifOutOctets**.
   c. Select **ifOutOctets**.
   d. Click **OK**.

      The ifOutOctets MIB variable should appear to the right of the add (+) arithmetic operator.

      You are ready to specify the Constant value 8 that corresponds with the leftmost multiply (*) arithmetic operator.
10. Click the leftmost * multiply.
11. In the Constant attribute, enter 8 and click Enter.

    The value 8 should appear to the right of the multiply (*) arithmetic operator that you previously selected.
12. In the MIB Expression, click divide (/).
13. Select the IfSpeed MIB Variable:
    a. Click to open the MIB Variable Tree.
    b. Navigate to ifSpeed.
    c. Double-click ifSpeed.
    d. Click **OK**.

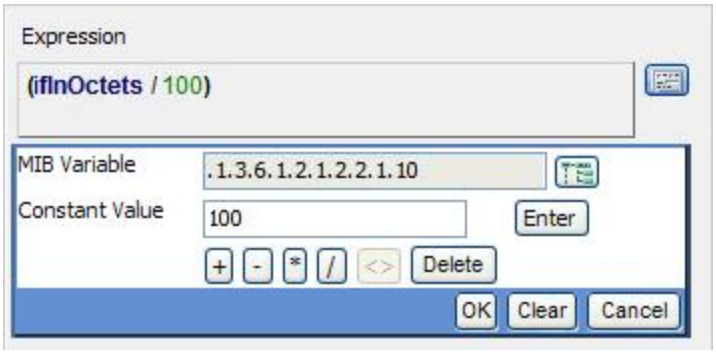
       The ifSpeed MIB Variable name should appear to the right of the divide (/) arithmetic operator you previously selected.
14. Click the rightmost * (multiply)
15. The Constant value 100 should appear to the right of the divide (/) arithmetic operator you previously selected.
16. In the Constant attribute, enter 100 and then click Enter.

17. Click **OK** to save your MIB Expression.

The following table describes each of the MIB Expression Editor options.

MIB Expression Editor Options

Attribute	Description
MIB Expression	Displays the MIB Expression as it is created. You can place the cursor in the MIB Expression field to specify where you want to add or replace an entry.
MIB Variable	You must select a MIB Variable using the MIB tree. Click the icon to access the MIB tree and navigate to the MIB variable of interest. **Note**: If you do not see a MIB that you recently loaded, close the Custom PollerCollection form, allow 1 minute for NNMi to cache the new MIB information, and then open the MIB tree again. After you select a MIB Variable, NNMi displays the MIB Variable's name. If you choose a MIB Variable that has multiple instances, you MUST specify a MIB Filter Variable and MIB Filter. For example, because a node can have multiple interfaces, MIB Variables containing interface information have multiple instances, one for each interface. You are required to provide a MIB Filter value to select the interfaces you want NNMi to poll. If you do not specify a MIB Filter Variable and MIB Filter, NNMi assumes the MIB variable is non-repeating. Click here for more information. For example, if you want to always gather additional HOST-RESOURCES-MIB status information about COM (communication) port devices, you would define the following: • MIB Expression: `hrDeviceStatus` • MIB Filter Variable: `hrDeviceDescr` • MIB Filter: `COM*` See "Create a Policy" (on page 947) for more information about the MIB Filter.
Constant Value	A numeric value to be used in the calculation for the MIB Expression. For example, you might want to include 100 as a constant when calculating percentages.
Enter	Includes the Constant Value in the MIB Expression.
	Adds the results.
	Subtracts the results.
	Multiplies the results.
	Divides the results.
	Changes the selected operator (+, -, *, and /) to the operator that appears next in sequence (from left to right) in the MIB Expression Editor. (The example below shows the operator sequence in the MIB Expression Editor.) For example, if you place your cursor at an add (+) operator in the MIB Expression, the MIB Expression Editor changes the add (+) operator to the minus (-) operator. If you place your cursor at the divide (/) operator in the MIB Expression as shown in the example below, the MIB Expression Editor changes the operator to the add (+) operator.

Attribute	Description
	 When using the <> (Change Operator) button, note the following: • You must select an operator in the MIB Expression before using the Change Operator (<>) button. • You can replace a MIB Variable with another MIB Variable or with a Constant. You can replace a Constant value with a MIB Variable or Constant.
Delete	Deletes the entry that is selected. If no entry is selected, NNMi deletes the last entry in the MIB Expression.
OK	Closes the MIB Expression Editor and saves your changes.
Clear	Removes any entries in the MIB Expression.
Cancel	Closes the MIB Expression Editor without saving your changes.

Configure Custom Polling

The Custom Poller feature enables you to take a proactive approach to network management by using SNMP MIB Expressions to specify additional information that NNMi should poll. You can also specify States that should be assigned to polled MIB Expression values, including any thresholds that should be set and monitored.

For example, if you have the HOST-RESOURCES-MIB loaded on your NNMi management server, you might want to monitor additional information using a single MIB variable, such as `hrDeviceStatus`, so that you can monitor information about a COM (communication) port, Loopback interface, or Ethernet Adapter Status (hrDeviceStatus). You might also want to monitor additional information using multiple MIB variables. For example, disk utilization could be calculated and polled using a MIB Expression similar to the following:(`hrStorageAllocationUnits` * `hrStorageSize`) / (`hrStorageUsed` * `hrStorageAllocation`)

Note: The MIB variables included in the MIB Expression that you want NNMi to poll must be loaded on the NNMi management server.

As an Administrator, to configure Custom Polling you want to perform the following tasks:

"Load MIBs" (on page 916)

"Enable or Disable Custom Poller" (on page 930)

"Create a Custom Poller Collection" (on page 930)

"Configure Basic Settings for a Custom Poller Collection" (on page 932)

"Specify the MIB Variable Information for a Custom Poller Collection" (on page 935)

"Configure Threshold Information for a Custom Poller Collection" (on page 943)

"Configure Comparison Maps for a Custom Poller Collection" (on page 945)

"Create a Policy" (on page 947)

"Create a Report Group (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 950)

For more details about configuring `Custom Poller`, see "Using Custom Poller by Example" , which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

Enable or Disable Custom Poller

The Custom Poller Configuration form enables you to enable or disable your Custom Poller Collections. You can also view the Custom Poller Collections and Policies that have been created.

Note: Custom Poller is not enabled by default. When Custom Polling is disabled, the State of Polled Instances retain the most recent value before Custom Poller was disabled.

To enable Custom Poller:

1. Navigate to the **Configuration** workspace.
2. Select **Custom Poller Configuration**.
3. Click **Enable Custom Poller** ☑.
4. Click the **Save** icon.

   To verify that Custom Poller is working as expected, see the report on the Custom Poller tab in **Help → System Information**.

To disable Custom Poller:

1. Navigate to the **Configuration** workspace.
2. Select **Custom Poller Configuration**.
3. Click to clear **Enable Custom Poller** ☐.
4. Click the **Save** icon.

   To verify that Custom Poller is working as expected, see the report on the Custom Poller tab in **Help → System Information**.

The Custom Poller Collections tab enables you to create a Custom Poller Collection. See "Create a Custom Poller Collection" (on page 930) for more information.

The Policies tab enables you to create one or more policies for a Collection. See "Create a Policy" (on page 947) for more information.

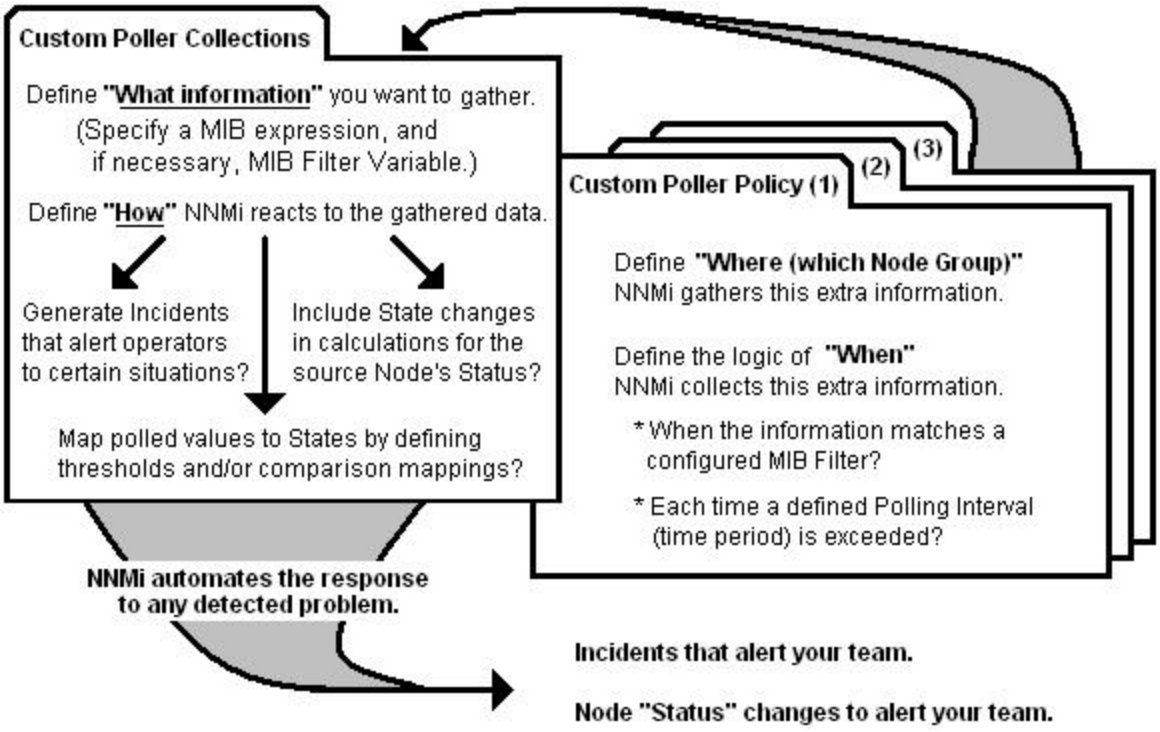
Create a Custom Poller Collection

A Custom Poller Collection defines the information you want to gather (poll) as well as how NNMi reacts to the gathered data. For example, you can specify whether you want to do either of the following:

- Configure Thresholds or Comparison Maps that map polled MIB Expression values to States and optionally causes incidents to be generated
- Include State changes in calculations for the source Node's Status.

Each Custom Poller Collection can have one or more Policies. Each Policy specifies the Node Group from which you want to gather the additional information. The first time a MIB Expression is validated with discovery information, the results appear in a Polled Instance object. The Polled Instance object is updated whenever a change in State occurs and includes the most recent polled value that caused the State to change.

Click here for a diagram that describes Custom Poller Collections and their associated Policies:



To create a Custom Poller Collection, do the following:

1. Navigate to the **Custom Poller Collections** tab.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Custom Poller Configuration**.
    c. Select the **Custom Poller Collections** tab.
    d. Do one of the following:
        - To create a Custom Poller Collection, click the New icon, and continue.
        - To edit a Custom Poller Collection, click the Open icon that precedes the Custom Poller Collection you want to edit, and continue.
2. Make your configuration choices (see table).
3. Click **Save and Close**.

Note: When you save a Collection configuration, each Policy for that Collection changes to Active State**Suspended**. When you are finished making your Custom Poller Configuration changes, set the Active State to **Active** for each of the policies in the Custom Poller Collection that you want to be in use. To make a Policy active, access the Custom Poller Configuration: Policy tab, open each associated Policy, and change the Active State to **Active**. See "Create a Policy" (on page 947) for more information.

To verify that Custom Poller is working as expected, see the report on the Custom Poller tab in **Help → System Information**.

Custom Poller Collection Configuration Tasks

Task	How
"Configure Basic Settings for a Custom Poller Collection" (on page 932)	Provide the basic information for a Custom Poller Collection configuration.
"Specify the MIB Variable Information for a Custom Poller Collection" (on page 935)	You specify the MIB Expression you want to poll. Use the MIB Expression editor to specify the MIB Variable and any constant or arithmetic operator you want to use in the MIB Expression. Navigate the MIB tree to select each MIB Variable.
"Configure Threshold Information for a Custom Poller Collection" (on page 943)	*Optional*. Specify minimum and maximum threshold values for the MIB Expression results and assign these thresholds to States.
"Configure Comparison Maps for a Custom Poller Collection" (on page 945)	*Optional*. Use Comparison Maps to assign a State value to a potential polled value of a MIB Expression.

Note: Thresholds and Comparison Maps contribute to State calculations. If you configure both Thresholds and Comparison Maps, NNMi first checks Threshold settings to determine State values. If the Threshold evaluates to non-Normal, NNMi uses the Threshold settings to determine State values. If the Threshold evaluates to Normal, NNMi checks for a non-Normal State using any Comparison Maps configuration.

Configure Basic Settings for a Custom Poller Collection

The Basic settings for a Custom Poller Collection include the Name of the Custom Poller Collection as well as whether to have this Collection affect a Node's Status or generate incidents under specified conditions. You also use the Basic settings to configure whether NNMi exports Custom Poller Collection metrics to a comma-separated values (CSV) file for use in other applications.

To configure the Basic settings for a Custom Poller Collection:

1. Navigate to the **Custom Poller Collection** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Custom Poller Configuration**.
    c. Locate the **Custom PollerCollections** tab.
    d. Do one of the following:

- To create a Collection, click the New icon, and continue.
- To edit a Collection, select a row, click the Open icon, and continue.

2. Provide the required basic settings (see the Basics for this Custom Polling Collection table).
3. Click **Save and Close** to return to the **Custom Poller Configuration** form.

   To verify that Custom Poller is working as expected, see the report on the Custom Poller tab in **Help → System Information**.

Basics for this Custom Poller Collection

Attribute	Description
Name	The name for the Custom Poller Collection configuration. The name can be up to 255 alphanumeric characters. Spaces are allowed. The following special characters (<, >, ", ', &, \, \, #) are not allowed. **Note**: The Custom Poller Collection name appears in any incidents generated as a result of the collection. Specify a name that will help you to identify the MIB information being polled.
Affect Node Status	Used to indicate whether you want each Polled Instance to affect the associated Node's Status. The first time a MIB Expression is validated with discovery information, the results appear in a Polled Instance object. The Polled Instance object is updated whenever a change in State occurs and includes the most recent polled value that caused the State to change. For example, you can specify that each time the `hrDeviceStatus` for a COM (communication) port returns a value of **5** (**down**), the status of the Polled Instance becomes **Critical** and affects the Status of the topology node, according to the Comparison Map configuration.
Generate Incident	Used to indicate whether you want NNMi to generate an incident when a threshold is reached or exceeded, or when a specified MIB value is returned and the resulting State is other than **Normal**.

Export Custom Poller Collection	If enabled ☑, NNMi exports the Custom Poller Collection to a comma-separated values (CSV) file that is written to the following directory:
	
	Windows:
	
	`%NnmDataDir%\shared\nnm\databases\custompoller\export\final`
	
	Unix:
	
	`$NnmDataDir/shared/nnm/databases/custompoller/export/final`
	
	When exporting Custom Poller Collections, note the following:

Attribute	Description

	• NNMi includes the following information in the CSV file:
	  ▪ Node UUID
	  ▪ IP address
	  ▪ Node Name
	  ▪ The MIB expression or the numeric Object Identifier of the MIB variable
	  ▪ Time stamp (in milliseconds)
	  ▪ Poll interval (in milliseconds)
	  ▪ MIB Instance (number used to identify the row in the MIB table)
	  ▪ Metric value
	• By default, NNMi accumulates the data and writes the metrics to the CSV file, one metric per Custom Poller Collection instance, every 5 minutes.
	• NNMi names each CSV file using the Custom Poller Collection name, appended with the timestamp (yyyymmddHHmmssSSS).
	• NNMi monitors the `custompoller` directory to ensure that the Custom Poller metrics do not fill the disk. By default, after the custompoller directory has consumed more than one gigabyte of disk space, NNMi removes the oldest metric files as it writes new files to the disk.
	• See the HP Network Node Manager i Software Deployment Reference, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`, for information about how to change default values, including the directory name, disk size, and the interval at which NNMi accumulates the data before writing the metric files to the disk.
	• If you have a High Availability (HA) environment, NNMi places the CSV files on the shared disk.
	• If you are using Application Failover, NNMi replicates these files to the failover system.
	  For more information about HA and Application Failover, see the HP Network Node Manager i Software Deployment Reference, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals` .
	• If you change the name of a Custom Poller Collection or the MIB Expression associated with a Custom Poller Collection that is exported, NNMi removes all of the historical data for that Custom Poller Collection.
	If disabled ☐, NNMi does not export the Custom Poller Collection information.

Attribute	Description
Compress Export File	If enabled ☑, NNMi exports the Custom Poller Collection in compressed format and appends .gz to the .csv file suffix. If you have more than one Custom Poller Collection with the same name, note the following: • If at least one of those Custom Poller Collections has Compress Export File enabled, NNMi compresses all of the exported Custom Poller Collections with the same name. • NNMi writes the Custom Poller Collection information to the same CSV file. If disabled ☐, NNMi does not compress the CSV file.

See "Specify the MIB Variable Information for a Custom Poller Collection" (on page 935) for information about the Variable attributes.

See "Configure Threshold Information for a Custom Poller Collection" (on page 943) for information about configuring Thresholds.

Specify the MIB Variable Information for a Custom Poller Collection

When specifying the MIB variable information, note the following:

- Each MIB Variable included in the MIB Expression must be loaded on the NNMi management server.
- You specify only one MIB Expression per Custom Poller Collection.
- You navigate the MIB tree to select a MIB Variable to include in a MIB Expression.

Variable Attributes

Attribute	Description
MIB Expression	You create a MIB Expression by using the MIB Expression form. If your NNMi role allows, to access the MIB Expression form, click the Lookup icon and do one of the following: • Select **Quick Find** to select an existing MIB expression. • Select **Open** to edit the current MIB expression. • Select **New** to create a MIB expression. See "MIB Expressions Form (Custom Poller)" (on page 936) for more information.
MIB Filter Variable	The MIB Filter Variable is the MIB variable value you want to use as a filter to determine which instances of the MIB expression to Custom Poll. If you specify a MIB Filter Variable, you must also specify a MIB Filter (value). For example, because a node can have multiple interfaces, MIB expressions containing interface information have multiple instances and require you to use a MIB Filter Variable and MIB Filter (value) to specify which interfaces you want NNMi to poll. You might use a MIB Filter Variable of `ifIndex` and a MIB Filter (value) of `1`. In this example, NNMi creates a Polled Instance for each interface with an (interface index) ifIndex value of 1 in the Node Group or Interface Group specified by the associated Custom Poller Policy. See "Create a Policy" for more information about Custom Poller Policies. Valid types for MIB Filter Variables include the following: • Integer • Unsigned Integer

Attribute	Description
	• Gauge • Octet String • IpAddress (IPv4 only) **Note**: The MIB Filter Variable must also be a MIB variable that has multiple instances (Table Entry MIB). Click the icon to open the MIB tree and select the MIB variable you want to use. When using MIB Filter Variables, note the following: • If you do not see a MIB that you recently loaded, close the Custom Poller Collection form, allow 1 minute for NNMi to cache the new MIB information, then open the MIB tree again. • To remove an unwanted MIB Filter Variable: a. Delete any MIB Filter Values from all Policies associated with the Custom Poller Collection. b. Edit the Custom Poller Collection to remove the MIB Filter Variable.

To specify Threshold information for the Custom Poller Collection, see "Configure Threshold Information for a Custom Poller Collection" (on page 943)

MIB Expressions Form (Custom Poller)

You can access the MIB Expression form in either of the following ways:

- From the **Configuration** workspace, **MIB Expressions** view.
- From the **Configuration** workspace, **Custom Poller Configuration** form.
- From the **MIB Specification** form. (Used when configuring SNMP Graph actions.)

When you want to create a MIB Expression to be used in Graphs, use the **MIB Expressions** view. See "MIB Expression Form (Line Graph)" (on page 920) for more information.

When you want to create a MIB Expression to be used in a Custom Poll, use the **Custom Poller Configuration** form.

Note: You can re-use any MIB Expression that you create for NNMi graphs or for Custom Poller. Use "MIB Expressions View" (on page 920) to see a list of the available MIB Expressions. Use the "Loaded MIBs View" (on page 912) to see a list of the MIBs loaded on the NNMi management server.

To create a MIB Expression using the Custom Poller Configuration form:

1. Navigate to the **Custom Poller Collection** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Custom Poller Configuration**.
   c. Locate the **Custom Poller Collections** tab.
   d. Do one of the following:

- To create a Collection, click the **New** icon, and continue.
- To edit a Collection, select a row, click the **Open** icon, and continue.

2. In the MIB Expression attribute, click the **Lookup** icon and do one of the following:
   - Select **Quick Find** to select and edit an existing MIB expression.
   - Select **Open** to edit the current MIB expression.
   - Select **New** to create a MIB expression.
3. Provide the required basic settings (see the MIB Expression Basic Attributes table).
4. Click **Save and Close** to return to the **Custom Poller Configuration** form.
5. To test your MIB Expression, select **Actions** → **Graph MIB Expression**. See "Test a MIB Expression (Custom Poller)" (on page 938) for more information.

   Note: You must save the MIB Expression before you use **Actions** → **Graph MIB Expression**.

MIB Expression Basic Attributes

Attribute	Description
Unique Key	Used as a unique identifier when exporting and importing MIB Expression definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the label value as part of the unique key as shown in the following example: `com.<your_company_name>.nnm.mibexp.<mib_expression_name>` The maximum length is 80 characters. **Note**: Unlike the Unique Key attributes associated with other objects, you can change the MIB Expression configuration's Unique Key value at any time.
Name	The name you want to use for the MIB information being polled. This name can be the same name as a MIB Variable used in the MIB Expression, or you can enter a name of your choice. Type a maximum of 50 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. No spaces are allowed.
Author	Indicates who created or last modified the MIB Expression. See Author form for important information. **Caution**: If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. Click the Lookup icon and select Quick Find to access the list of existing Author values or click New to create one.
Expression	Click the icon to access the MIB Expression editor. See "Use the MIB Expression Editor (Custom Poller)" (on page 939) for information about using the MIB Expression editor. Note the following: • The MIB containing the variable must be loaded on the NNMi management server. • If a MIB Expression includes more than one MIB Variable that has multiple instances

Attribute	Description
	(Table Entry MIB), select a MIB Filter and MIB Filter Variable that can be consistently applied to each Table Entry MIB in the expression. • Although it is strongly discouraged, to configure Custom Polling for all instances of a repeating MIB, you can use the same MIB variable for both the MIB Expression and the MIB Filter Variable. • If your MIB Expression contains an invalid MIB Variable, NNMi is not able to create an associated Polled Instance. If Polled Instances are not created as expected, check the Custom Node Collection view for Discovery State and Discovery State Information values. • If Polled Instances are created, but errors occur while processing the MIB Expression data from a device's SNMP Agent, information is logged to the analysis.0.0.log file. Examples of possible errors include divide by zero (0) or data unavailable. See "Verify that NNMi Services are Running" (on page 47)for more information about log files.
Display numeric MIB OIDs in the Expression	Enables you to display the MIB object identifier (OID) rather than the MIB variable name in the MIB Expression. Select **Display MIB OIDs in the Expression** ☑ to replace any MIB variable name with the MIB OID value in the MIB Expression. Clear **Display MIB OIDs in the Expression** ☐ to display the MIB variable names rather than the MIB OIDs within the MIB Expression.
Description	NNMi provides the Description attribute to help you further identify the current MIB Expression configuration. Use the description field to provide additional information that you would like to store about the current MIB expression configuration. Type a maximum of 2000 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed.

Note: The **Instance Display Configuration** section of the MIB Expression form is used when configuring MIB Expressions for Line Graphs.

Test a MIB Expression (Custom Poller)

The Actions menu enables you to test the results of a MIB Expression using a Line Graph.

Note: You must save the MIB Expression before you use **Actions** → **Graph MIB Expression**.

To graph the results for a MIB Expression:

1. Navigate to the **MIB Expression** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **MIB Expressions**.
      Note: You can also access the MIB Expressions form when creating Line Graphs and when creating Custom Poller Collections. See "MIB Expression Form (Line Graph)" (on page 920) and "MIB Expressions Form (Custom Poller)" (on page 936) for more information.
2. Click ☑ to select the MIB Expression you want to graph.

3. Select **Actions** → **Graph MIB Expression**.

   The dialog for selecting a node apears.

4. Click the **Lookup** icon and select **Quick Find**.

5. Select the node you want to use to test your MIB Expression results.

   NNMi displays a Line Graph using the selected node and calculating the results for the MIB Expression you selected.

   Note the following:

   - When evaluating MIB Expressions that include MIB variables of type Counter or Counter64, NNMi requests both the high capacity and low capacity counter variable for any interface instance. If the high capacity counter (Counter 64) is enabled for any given interface instance, NNMi uses the high capacity counter.
   - When evaluating MIB Expressions that include MIB variables of type Counter32, NNMi requests only the low capacity counter information for any interface instance.

Use the MIB Expression Editor (Custom Poller)

Use the MIB Expression Editor to specify MIB Variables and any Constant values or arithmetic operators in your MIB Expression.

- For a description of each MIB Expression Editor option, see the table below.
- Before you start, review the MIB Expression Editor guidelines, click here.
  - As a general guideline, begin by writing out the MIB Expression. Then in the MIB Expression Editor, begin creating your MIB Expression by selecting your arithmetic operators (+, -, *, or /) from the outermost parenthesis to the innermost parenthesis. Each time you specify an arithmetic operator (+, -, *, or /), NNMi creates a set of parenthesis to specify the ordering of the mathematical calculation.
  - When adding arithmetic operators (+, -, *, or /) to a MIB Expression, first click to select the location in the MIB Expression at which you want to add the arithmetic operator.
  - Click to select the arithmetic operator (for example +) in the MIB Expression, before selecting the MIB variable or Constant value that you want to add, subtract, multiply or divide.
  - NNMi inserts arithmetic operators, MIB Expressions, and Constant values from the left to right.
  - To replace an arithmetic operator use the <> (Change Operator) button (see table).
  - To replace a MIB Variable or Constant value, click to select the existing value in the MIB Expression and then select the new MIB variable or enter the new Constant value.

    Note: You can replace a MIB Variable with another MIB Variable or with a Constant value. You can replace a Constant value with a MIB Variable or Constant value.
  - You can drag any of the following items to a new location in the MIB Expression:
    - MIB variable
    - Constant value
    - An operation, such as **(IfInOctets + IfOutOctets)**
- For information about moving items to a new location within your MIB Expression, click here.
  - To move an arithmetic operation (for example, **(IfInOctets + IfOutOctets)**), click the arithmetic

operator before dragging it to a new location.

- To move a MIB Variable or Constant Value, click the MIB Variable or Constant Value you want to move before dragging it to a new location.
- If you are moving the selected item to the right, NNMi places the item to the right of the new location.
- If you are moving the selected item to the left, NNMi places the item to the left of the new location.
- As you drag a selected item, an underline indicates the current target location.
- If you drag a selected item past the outermost parenthesis, it is deleted. If desired, you can re-enter the value in the new location.

MIB Expression Example

To poll or graph a MIB Expression that calculates the percentage of available bandwidth on a half-duplex interface, create the following MIB Expression:

((((ifInOctets + ifOutOctets) * 8) / ifSpeed) * 100)

For an animated demonstration of creating this MIB Expression, click here.

For step-by-step instructions about creating this MIB Expression, click here.

To create the expression above, begin by specifying each arithmetic operator from the outermost parenthesis to the innermost parenthesis.

1. Click * (multiply).
2. Click / (divide).

   Now that you have multiple entries in your MIB Expression, you need to click to select the location in the MIB Expression to which you want to add each remaining arithmetic operators.
3. In the MIB Expression, click **/** (divide).

   The divide (/) arithmetic operator and its surrounding parenthesis should appear highlighted. Because NNMi inserts arithmetic operators, MIB variables, and Constant values from left to right, selecting / (divide) places the next arithmetic operator to the left of the divide arithmetic operator.
4. Click * (multiply).

   The multiply (*) arithmetic operator and its parenthesis should appear to the left of the divide arithmetic operator you previously selected.
5. In the MIB Expression, click the leftmost ***** (multiply).

   The multiply (*) arithmetic operator and its surrounding parenthesis should appear highlighted.
6. Click + (add).

   The add (+) arithmetic operator and its parenthesis should appear to the left of the multiply (*) arithmetic operator you previously selected.

   Now that you have specified the arithmetic operators, you are ready to add the MIB variables and Constant values. Begin by selecting the arithmetic operator in the MIB Expression to which you will add MIB variables, Constant values, or both. We will begin with the leftmost arithmetic operation.

   Note: As you add your MIB variables or Constant values, make sure you first select the corresponding arithmetic operator within the MIB Expression.
7. In the MIB Expression attribute, click + (add).

8. Select the IfInOctets MIB Variable:
    a. Click to open the MIB Variable Tree.
    b. Navigate to **ifInOctets**.
    c. Select **ifInOctets**.
    d. Click **OK**.

    The ifInOctets MIB variable should appear to the left of the add (+) arithmetic operator.
9. Select the IfOutOctets MIB Variable:
    a. Click to open the MIB Variable Tree.
    b. Navigate to **ifOutOctets**.
    c. Select **ifOutOctets**.
    d. Click **OK**.

    The ifOutOctets MIB variable should appear to the right of the add (+) arithmetic operator.

    You are ready to specify the Constant value 8 that corresponds with the leftmost multiply (*) arithmetic operator.
10. Click the leftmost * multiply.
11. In the Constant attribute, enter 8 and click Enter.

    The value 8 should appear to the right of the multiply (*) arithmetic operator that you previously selected.
12. In the MIB Expression, click divide (/).
13. Select the IfSpeed MIB Variable:
    a. Click to open the MIB Variable Tree.
    b. Navigate to ifSpeed.
    c. Double-click ifSpeed.
    d. Click **OK**.

    The ifSpeed MIB Variable name should appear to the right of the divide (/) arithmetic operator you previously selected.
14. Click the rightmost * (multiply)
15. In the Constant attribute, enter 100 and then click Enter.
16. The Constant value 100 should appear to the right of the divide (/) arithmetic operator you previously selected.
17. Click **OK** to save your MIB Expression.

The following table describes each of the MIB Expression Editor options.

MIB Expression Editor Options

Attribute	Description
MIB Expression	Displays the MIB Expression as it is created. You can place the cursor in the MIB Expression field to specify where you want to add or replace an entry.

Attribute	Description
MIB Variable	You must select a MIB Variable using the MIB tree. Click the icon to access the MIB tree and navigate to the MIB variable of interest. **Note**: If you do not see a MIB that you recently loaded, close the Custom PollerCollection form, allow 1 minute for NNMi to cache the new MIB information, and then open the MIB tree again. After you select a MIB Variable, NNMi displays the MIB Variable's name. If you choose a MIB Variable that has multiple instances, you MUST specify a MIB Filter Variable and MIB Filter. For example, because a node can have multiple interfaces, MIB Variables containing interface information have multiple instances, one for each interface. You are required to provide a MIB Filter value to select the interfaces you want NNMi to poll. If you do not specify a MIB Filter Variable and MIB Filter, NNMi assumes the MIB variable is non-repeating. Click here for more information. For example, if you want to always gather additional HOST-RESOURCES-MIB status information about COM (communication) port devices, you would define the following: • MIB Expression: `hrDeviceStatus` • MIB Filter Variable: `hrDeviceDescr` • MIB Filter: `COM*` See "Create a Policy" (on page 947) for more information about the MIB Filter.
Constant Value	A numeric value to be used in the calculation for the MIB Expression. For example, you might want to include 100 as a constant when calculating percentages.
Enter	Includes the Constant Value in the MIB Expression.
	Adds the results.
	Subtracts the results.
	Multiplies the results.
	Divides the results.

Attribute	Description
<>	Changes the selected operator (+, -, *, and /) to the operator that appears next in sequence (from left to right) in the MIB Expression Editor. (The example below shows the operator sequence in the MIB Expression Editor.) For example, if you place your cursor at an add (+) operator in the MIB Expression, the MIB Expression Editor changes the add (+) operator to the minus (-) operator. If you place your cursor at the divide (/) operator in the MIB Expression as shown in the example below, the MIB Expression Editor changes the operator to the add (+) operator. When using the <> (Change Operator) button, note the following: • You must select an operator in the MIB Expression before using the Change Operator (<>) button. • You can replace a MIB Variable with another MIB Variable or with a Constant. You can replace a Constant value with a MIB Variable or Constant.
Delete	Deletes the entry that is selected. If no entry is selected, NNMi deletes the last entry in the MIB Expression.
OK	Closes the MIB Expression Editor and saves your changes.
Clear	Removes any entries in the MIB Expression.
Cancel	Closes the MIB Expression Editor without saving your changes.

Configure Threshold Information for a Custom Poller Collection

Prerequisite: You must have specified the MIB Expression you want to poll. See "Specify the MIB Variable Information for a Custom Poller Collection" (on page 935) for more information.

Thresholds specify minimum and maximum values for the MIB Expression that is polled. These values are used to determine when to generate an incident as well as the State of the Polled Instance.

When configuring Threshold settings, note the following:

- If a polled value is between the high range and the low range, the Polled Instance state is Normal.
- You can configure Comparison Maps, which also contribute to State calculations. If you configure both Thresholds and Comparison Maps, NNMi first checks the Threshold settings to determine State values. If the threshold evaluates to non-Normal, NNMi uses the Threshold settings to determine State values. if the Threshold evaluates to Normal, NNMi checks for a non-Normal State using any Comparison Map

configuration. See "Configure Comparison Maps for a Custom Poller Collection" (on page 945) for more information about configuring Comparison Maps.

- The MIB Expression must evaluate to a numeric type. (OCTET STRING type is not supported.)
- When evaluating Threshold configurations with MIB Expressions that include one or more MIB Variables of type Counter or Counter64, NNMi evaluates the MIB Variable value using the difference in value between the most recent poll and the poll before it.

To configure thresholds for a MIB Variable:

1. Navigate to the Custom Poller Collection form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Custom Poller Configuration**.
   c. Navigate to the Custom Poller Collections tab.
   d. Do one of the following:
      - To create a collection, click the New icon, and continue.
      - To edit a collection, select a row, click the Open icon, and continue.
   a. Locate the **Thresholds** section of the form.
2. Make your configuration choices (see table).
3. Click **Save and Close** to close the **Custom Poller Collection** form.
4. Click **Save and Close** to close the **Custom Poller Configuration** form.

   To verify that Custom Poller is working as expected, see the report on the Custom Poller tab in **Help → System Information**.

Threshold Attributes for a Custom Polled Instance

Attribute	Description
High State	The Polled Instance State when NNMi returns a value that exceeds a specified High Value. Required only for thresholds with a High Value setting.
High Value	Value used to define the high threshold. Required only for thresholds with a High Value setting.
High Value Rearm	Value used to define when the Polled Instance is no longer in the High State. The default value is the High Value.
High Trigger Count	The number of consecutive times the returned value must exceed the specified High Value to transition to the High State. The default value is 1.
Low State	The Polled Instance State when NNMi returns a value below a specified Low Value. Required only for thresholds with a Low Value setting.
Low Value	Value used to define the low threshold. Required only for thresholds with a Low Value setting.
Low Value Rearm	The value used to define when the Polled Instance is no longer in the Low State. The default value is the Low Value.

Attribute	Description
Low Trigger Count	The number of consecutive times the returned value must exceed the specified Low Value to transition to the Low State. The default value is 1.

Configure Comparison Maps for a Custom Poller Collection

Prerequisite: You must know the valid values that might be returned when the MIB Expression is polled.

Custom Poller enables you to map the returned value of a MIB Expression to a Custom Poller Polled Instance State. These values are used to determine when to generate an incident, as well as the State of the Polled Instance. For example, you might want the hrDeviceStatus value of **5** (down) to be mapped to a **Critical** State. This means that NNMi changes the State of the Polled Collection Instance to **Critical** each time the hrDeviceStatus returns a value of **5** when polled.

When configuring Comparison Maps, note the following:

- NNMi applies the Comparison Maps according to the Ordering number defined. The first comparison criteria met defines the State for the Polled Instance.
- You can configure Thresholds, which also contribute to State calculations. If you configure both Thresholds and Comparison Maps, NNMi first checks the Threshold settings to determine State values. If the threshold evaluates to non-Normal, NNMi uses the Threshold settings to determine State values. if the Threshold evaluates to Normal, NNMi checks for a non-Normal State using any Comparison Map configuration. See "Configure Threshold Information for a Custom Poller Collection" (on page 943) for more information about configuring thresholds.
- When evaluating Threshold configurations with MIB Expressions that include one or more MIB Variables of type Counter or Counter64, NNMi evaluates the MIB Variable value using the difference in value between the most recent poll and the poll before it.

To configure Comparison Maps for a MIB Expression:

1. Navigate to the **Custom Poller Collection** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Custom Poller Configuration**.
    c. Navigate to the **Custom Poller Collections** tab.
    d. Do one of the following:
        - To create a collection, click the New icon, and continue.
        - To edit a collection, select a row, click the Open icon, and continue.
2. Locate the **Comparison Maps** tab.
3. Do one of the following:
    - To create a Comparison Map, click the New icon, and continue.
    - To edit a Comparison Map, select a row, click the Open icon, and continue.
4. Make your configuration choices (see table).
5. Click **Save and Close** to close the **Custom Poller Collection** form.
6. Click **Save and Close** to close the **Custom Poller Configuration** form.

Note: Each time you save a Comparison Maps configuration, NNMi suspends Custom Polling for the Custom Poller Collection. When you finish making your Comparison Mapping changes, set the Active State to **Active** for each of the policies in the Custom Poller Collection that you want to be in use. See "Create a Policy" (on page 947) for more information.

To verify that Custom Poller is working as expected, see the report on the Custom Poller tab in **Help → System Information**.

State Mapping Attributes

Attribute	Description
Ordering	The order in which the State mapping (Comparison Maps) operations should be performed. **Note**: NNMi uses the Ordering value to determine which State mapping to use. The lower the number, the higher the priority. For example, 1 is the highest priority.
Comparison Operator	Operator used to evaluate the Comparison Value and subsequently determine its State. For example, the < (less than) Comparison Operator indicates the polled value must be less than the Comparison Value specified to change the Custom Poller Polled Instance to the specified State value. Possible Comparison Operator values are: • **<** (Less than) • **<=** (Less than or equal to) • **=** (Equal to) • **!=** (Not equal to) • **>** (Greater than) • **>=** (Greater than or equal to) • **is null** (Null or unavailable) • **is not null** (Contains a value) • **default** (Sets the State when no matches are found using the other Comparison Operators) **Note**: Ordering for the **default** Comparison Operator must be the last.
Comparison Value	The value returned when the MIB Expression is evaluated when polled.
State Mapping	The State to assign to the Custom Poller Polled Instance when the polled value is returned. For example, each time the value **3** (warning) is returned when NNMi polls hrDeviceStatus, you can specify that you want NNMi to change the State of the Polled Instance to **Warning**. Possible State values for a *Polled Instance* (Threshold = High State/Low State; or Comparison Map = State Mapping) are: Normal Warning Minor

Attribute	Description
	Major Critical

Create a Policy

Prerequisite: Make sure that the Node Group has been created to which you want to apply the Custom Polling Policy. See Define Node Groups for more information about creating Node Groups.

You can create one or more policies for a Custom Poller Collection. When configuring a Custom Poller Policy, you define which MIB variable or variables NNMi gathers from members of a specific Node Group.

Note: If you configure more than one Policy per Collection, each Policy must be for a different Node Group.

To configure a Custom Poller Policy:

1. Navigate to the Custom Poller Policies form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Custom Poller Configuration**.
   c. Locate the **Policies** tab.
   d. Do one of the following:
      - To create a policy, click the New icon, and continue.
      - To edit a policy, select a row, click the Open icon, and continue.
2. Make your configuration choices (see table).
3. Click **Save and Close** to return to the **Custom Poller Configuration** form.

   To verify that Custom Poller is working as expected, see the report on the Custom Poller tab in **Help → System Information**.

Custom Poller Policy Attributes

Attribute	Description
Name	The Name of the Policy configuration. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+) are allowed. No spaces are allowed. **Note**: The Policy name appears in any incidents generated as a result of the Collection. Specify a name that will help you to indicate the types of nodes that are polled with this policy.
Ordering	The order in which the Policy should be considered for nodes that appear in multiple Node Groups and therefore might have conflicting Policies. For example, Ordering is used in the following scenario: • Two Policies associated with the same Custom Poller Collection specify `ifOperStatus` as the MIB Expression. • One Policy uses the Routers Node Group and the second Policy uses the Switches Node Group. • Each Policy has a different Polling Interval.

Attribute	Description
	In the example scenario above, if a device was in both the Routers Node Group and the Switches Node Group, NNMi would poll the device only one time according to the Policy with the lowest Ordering number.
Collection	Click the Lookup icon and select Quick View or Open to display more information about the Custom Poller Collection.
Active State	Use the Active State setting to specify which Custom Poller Policies you want to enable or temporarily disable. The Active State for the associated Custom Collect Policy. Possible values are described below: **Active** - Indicates the Custom Poller Policy is in use. **Inactive** - Indicates the Custom Poller Policy is not in use. NNMi removes all Polled Instances associated with the Policy. **Suspended** - Indicates someone on your team changed this Custom Poller Policy's *Active State* to `Suspended`, or the NNMi administrator disabled Custom Poller in the *Global Control* settings of **Configuration** workspace, **Custom Poller Configuration** form. NNMi suspends polling and retains the most recent State value from before the Policy was suspended.
Node Group	The Node Group to which the Custom Poller Policy applies.

Attribute	Description
MIB Filter	The MIB Filter value to be used as the filter for determining the Polling Instances. When using a MIB Filter, note the following: • The MIB Filter value must match the return type of your filter variable. For example, because `hrDeviceDescr` is of type String, to poll only those MIBs associated with each node that includes the description for a COM (communication) port, **COM*** would be the MIB Filter for the example MIB Filter Variable `hrDeviceDescr`. • If your MIB Expression includes a MIB Variable that has multiple instances, you MUST specify a MIB Filter Variable and MIB Filter. For example, because a node can have multiple interfaces, MIB variables containing interface information have repeating instances and require you to use a MIB Filter to specify which interfaces you want NNMi to poll. • If your MIB Expression contains more than one MIB Variable with multiple instances, the MIB Filter must apply to each of these MIB Variables. • Valid types for MIB Filter Variables include the following: ▪ INTEGER ▪ UNSIGNED INTEGER ▪ Gauge ▪ OCTET STRING ▪ IpAddress (IPv4 only) Click here for information about valid values for the MIB Filter Expression. Valid values for MIB Filter include the following: • For numeric values only, you can specify a range using a dash (-). For example 1-6. • For string values only, you can use the wildcard character (*) at either the beginning or end of a string value. For example: *vlan, vlan*, and *vlan*. To match all instances, specify *. • For either numeric or sting values, you can use the Not operator (!) at the beginning of the MIB Filter expression. For example: !1-3, !*vlan, and !vlan. When using MIB Filters, note the following: • NNMi uses exact matches for string comparisons. • String comparisons are case insensitive. • NNMi ignores leading and trailing white spaces • You can specify multiple MIB Filter expressions by separating each MIB Filter using a comma (,) • When you enter multiple MIB Filter expressions, NNMi um combines them using the OR operator. • To include the dash (-), asterisk (*), or exclamation (!), or comma (,) in your search, use a leading backslash (\) before the special character.
Polling Interval	The interval in which to perform the Custom Poll.

Create a Report Group (HP Network Node Manager iSPI Performance for Metrics Software)

Report Groups enable you to define which Custom Poller Collections are reported to HP Network Node Manager iSPI Performance for Metrics Software. Each Report Group you configure represents a tab in the HP Network Node Manager iSPI Performance for Metrics Software Report Menu.

Caution: If you delete a Report Group, HP Network Node Manager iSPI Performance for Metrics Software removes all historical reporting data associated with that Report Group. To retain the historical reporting data, change the Active State of the associated Custom Poller policy to **Suspend**. See "Create a Policy" (on page 947) for more information.

To configure a Report Group:

1. Navigate to the **Report Group** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Custom Poller Configuration**.
   c. Locate the **Report Groups** tab.
   d. Do one of the following:
      - To create a Report Group, click the New icon, and continue.
      - To edit a Report Group, select a row, click the Open icon, and continue.
      - To delete a Report Group, select a row, and click the Delete icon.
2. Make your configuration choices (see table).
3. Click **Save and Close** to return to the **Custom Poller Configuration** form.
4. Create a Report Collection to associate one or more Custom Poller Collections with this Report Group. See "Create a Report Collection (HP Network Node Manager iSPI Performance for Metrics Software)" (on page 950) for more information.

Custom Poller Report Group Attributes

Attribute	Description
Name	Enter the name that you want to appear in the tab in the HP Network Node Manager iSPI Performance for Metrics Software Report Menu for this Report Group. The name can be up to 255 alphanumeric characters. Spaces are allowed. The following special characters (<, >, ", ', &, \, \, #) are not allowed.

Create a Report Collection (HP Network Node Manager iSPI Performance for Metrics Software)

Report Collections enable you to specify a Custom Poller Collection to be associated with a Report Group as well as the type of data that is being collected. You can create one or more Report Collections for a Report Group.

Caution: If you delete a Report Collection, HP Network Node Manager iSPI Performance for Metrics Software removes all historical reporting data for that Report Collection.

To configure a Report Collection:

1. Navigate to the **Report Collection** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select **Custom Poller Configuration**.
    c. Locate the **Report Groups** tab.
    d. Do one of the following:
        - To create a Report Group, click the New icon, and continue.
        - To edit a Report Group, select a row, click the Open icon, and continue.
2. Select the **Report Collections** tab.
3. Do one of the following:
    - To create a Report Collection, click the New icon, and continue.
    - To edit a Report Collection, select a row, click the Open icon, and continue.
    - Make your configuration choices (see table).
4. Click **Save and Close** to return to the **Custom Poller Configuration** form.

    To verify that Custom Poller is working as expected, see the report on the Custom Poller tab in **Help → System Information**.

Custom PollerReport Collection Attributes

Attribute	Description
Custom Poller Collection	Specifies a Custom Poller Collection that should be associated with the Report Group you are configuring. Click the Lookup icon, and do one of the following: • To specify a Custom Poller Collection, select Quick Find . In the Quick Find dialog, select the Custom PollerCollection of interest. • To create a Custom Poller Collection, click the New icon, and continue. • To edit a Custom Poller Collection, select a row, click the Open icon, and continue. When specifying a Custom PollerCollection, note the following: • A Custom Poller Collection can be associated with only one Report Group.

Attribute	Description

	• If you associate more than one Custom Poller Collection with the same Report Group, make sure the combination of Collections will generate a meaningful report. Use the following general guidelines:
	  ▪ Select Collections with MIB Variables that are indexed in the same MIB table. For example, you might group a collection that includes power supply information, such as UPS line voltage (upsInputVoltage and upsOuputVoltage), UPS line current (upsInputCurrent and upsOutputCurrent) and UPS line power (upsInputPower and upsOutputPower).
	  ▪ Select Collections with MIB Variables that are stored in different MIBs, but that would be useful to visualize together at an aggregate level. For example, you might choose to group power supply line load (upsOutputPercentLoad) and power supply battery temperature (upsBatteryTemperature).
	  ▪ Select Collections representing the same index value or similar data across Custom Poller Collections. For example, you might want to examine environment sensor information (such as temperature, humidity, dew point, airflow, and audible sounds (e.g. alarms)), in the same report even though this information comes from different MIBs.
	• As soon as the Report Collection is saved, NNMi updates the information in the HP Network Node Manager iSPI Performance for Metrics Software Report Menu.

Attribute	Description
Report Data Type	Determines how HP Network Node Manager iSPI Performance for Metrics Software interprets the metrics to be displayed. Possible values include: • **Gauge** – Represents single non-cumulative values. Examples of Gauge data types include Response Time, Bit Rate, and Temperature. When Gauge data types are aggregated, HP Network Node Manager iSPI Performance for Metrics Software calculates the minimum, maximum, and average values. • **Percent** – Represents single non-cumulative values that are formatted with a percent sign (%) and two decimal places. Examples of Percent data types include Utilization and Discard Rate. When Percent data types are aggregated, HP Network Node Manager iSPI Performance for Metrics Software calculates the minimum, maximum, and average values.. • **Counter** – Represents incremental values. Examples of Counter data types include byte counts, packet counts, and flow counts. When Counter data types are aggregated, HP Network Node Manager iSPI Performance for Metrics Software calculates the sum.

Purchase an HP Network Node Manager i Smart Plug-in

HP Network Node Manager i Software Smart Plug-ins (iSPIs) extend NNMi capabilities.For more information about purchasing, contact your HP sales representative. For example, each NNM iSPI may or may not:

- Enhance the data that is available.
- Add new workspaces, views, and forms.
- Add tabs to existing NNMi forms.
- Change the features of the NNMi user interface.

Multiple NNM iSPIs are available, enabling you to manage your network in a way that makes sense in your organization:

- HP Network Node Manager iSPI Network Engineering Toolset Software (NNM iSPI NET)

  Tip: See the NNMi Release Notes for a description, **Help → Documentation Library → Release Notes**.
- HP Network Node Manager iSPI for IP Multicast Software
- HP Network Node Manager iSPI for MPLS Software
- HP Network Node Manager iSPI Performance for Metrics Software
- HP Network Node Manager iSPI Performance for Quality Assurance Software
- HP Network Node Manager iSPI Performance for Traffic Software
- HP Network Node Manager iSPI for IP Telephony Software

See also the documentation available for each NNM iSPI, available at:
`http://h20230.www2.hp.com/selfsolve/manuals`.

Related Topics:

"Track Your NNMi Licenses" (on page 1019)

"Extend a Licensed Capacity" (on page 1020)

"Integrations with Other HP Products" (on page 954)

Integrations with Other HP Products

Multiple HP Software products can be configured to share data with NNMi and receive data from NNMi. See the “Integrations” section of the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

Each integration adds some or all of the following functionality (depending on the integration):

- NNMi incidents are available in the integrated product’s events viewer.
- NNMi receives and monitors traps related to the integrated product.
- NNMi operators can open some of the integrated product’s views from within the NNMi console. Those views are in context of the object selected in the NNMi console (for example, node or interface).
- Operators of the integrated product can open some NNMi console views from within the integrated product. Those views are in context of the object selected in the integrated product.
- Network topology (inventory) information is shared between NNMi and the integrated product.

For information about the available integrations, see **Help → Documentation Library → Release Notes** and contact your HP sales representative.

Related Topics:

"Track Your NNMi Licenses" (on page 1019)

"Extend a Licensed Capacity" (on page 1020)

"Purchase an HP Network Node Manager i Smart Plug-in" (on page 953)

Integration Configuration Form

A variety of HP and third-party software products (that run independently of NNMi) can be integrated with NNMi. See **Help → Documentation Library → Release Notes**, and locate the **Support Matrix** for a complete list of supported products.

Each integration adds some or all of the following functionality (depending on the integration):

- NNMi incidents are available in the integrated product’s events viewer.
- NNMi receives and monitors traps related to the integrated product.
- NNMi operators can open some of the integrated product’s views from within the NNMi console. Those views are in context of the object selected in the NNMi console (for example, node or interface).
- Operators of the integrated product can open some NNMi console views from within the integrated product. Those views are in context of the object selected in the integrated product.
- Network topology (inventory) information is shared between NNMi and the integrated product.

Some of these products require that you provide information in the NNMi **Integration Module Configuration** workspace. Use the appropriate integration configuration form to provide the information required for enabling an integration between NNMi and the associated product. For the latest information about an integration and the fields on its integration configuration form, see the appropriate chapter in the

“Integrations” section of the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

Note: HP Network Node Manager i Software Smart Plug-ins (iSPIs) do not use the Integration Module Configuration workspace. NNM iSPIs have an entirely different configuration strategy. See "Purchase an HP Network Node Manager i Smart Plug-in" (on page 953).

Integrating NNMi Elsewhere with URLs

Use URLs to provide access to the console or certain NNMi features. For example:

- Embed views within your company Web portal.
- Launch a map from within other applications, such as from an email.
- Launch a filtered view from a browser window to quickly find the information you need.
- Run a tool without opening the console.

The URLs you write must conform to "W3C Rules for URLs" (on page 954).

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Prerequisite: NNMi requires authentication for access through URLs. See "Authentication Requirements for URLs Access".

Related Topics:

"Launch the Console (showMain)" (on page 957)

"Launch a View (showView)" (on page 957)

"Launch a Form (showForm)" (on page 990)

"Launch Menu Items (runTool)" (on page 1001)

"Confirm that NNMi Is Running (cmd=isRunning)" (on page 1017)

W3C Rules for URLs

The World Wide Web Consortium (W3C) allows only ASCII characters in URLs.

When configuring URLs, the following characters are always allowed:

- Alpha-numeric (A-Z a-z 0-9)
- - (hyphen)
- . (period)
- _ (underline)
- ~ (tilde)

Depending on the browser and the context, some characters require special formatting with Percent Encoding. A small number of possible values are shown in the quick reference table below.

You can designate the space character several ways:

- `+` (works in all browsers, recommended because it is easiest to read)
- `%20` (Percent Encoded value, works in all browsers)
- space character (works in the browsers supported by NNMi, but is not guaranteed to work in all browsers)

RFC 3986 Characters Reserved as Delimiters
(If not specifying a delimiter, use Percent-Encoding value)

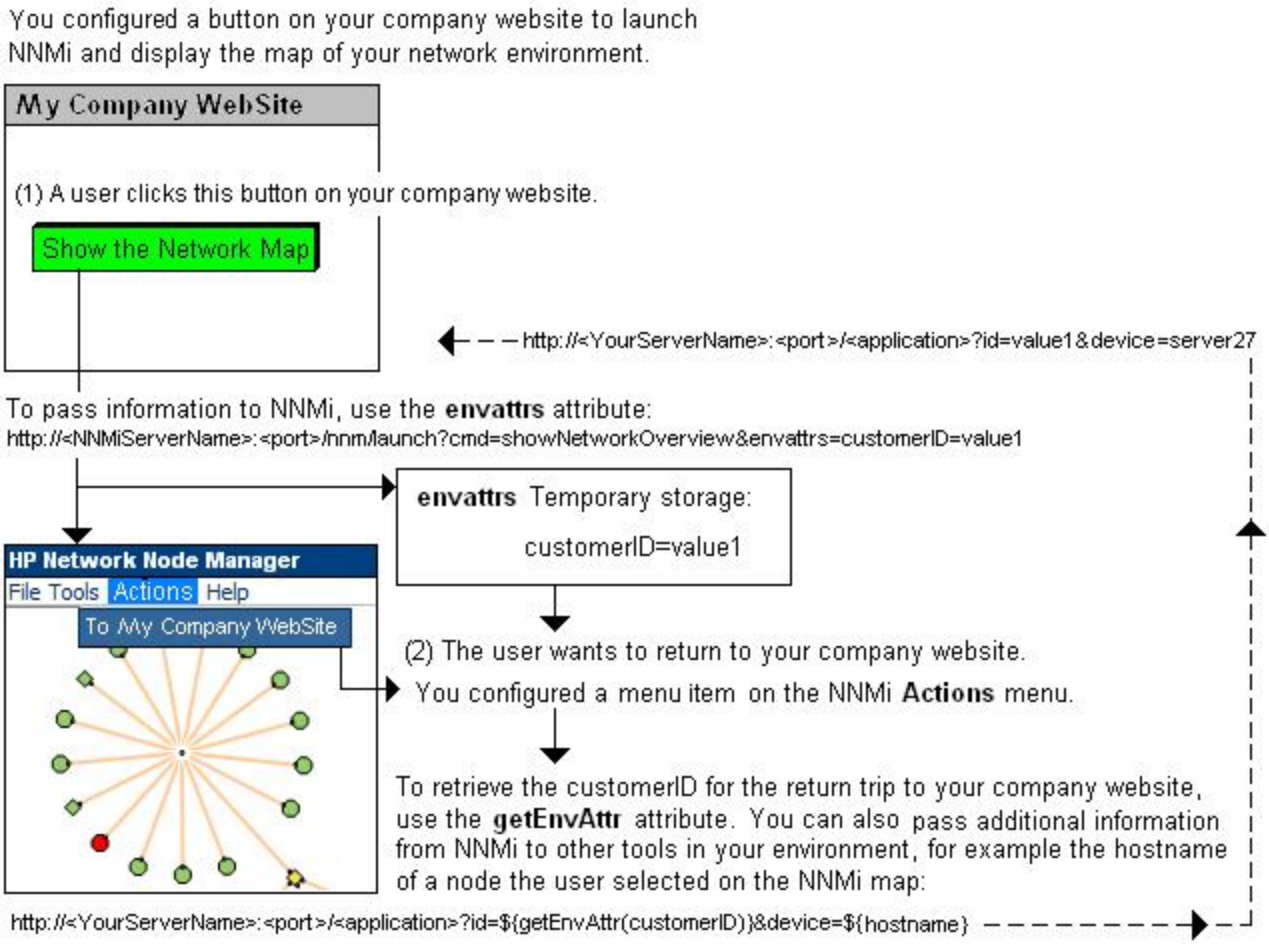
Character	:	/	?	#	[	]	@	!	$
Percent Encoded	%3A	%2F	%3F	%23	%5B	%5D	%40	%21	%24
Character	&	'	(	)	*	+	,	;	=
Percent Encoded	%26	%27	%28	%29	%2A	%2B	%2C	%3B	%3D

Additional Commonly Used Characters and Their Percent Encoding

Character	space	%	<	>
Percent Encoded	%20 (or + allowed)	%25	%3C	%3E

Pass Environment Attributes

Environment Attributes (`envattrs`) are received from another application when NNMi is launched from that external application, see "Launch a View (showView)" (on page 957) or "Launch a Form (showForm)" (on page 990) for more information. These `envattrs` attributes are session-specific and not stored in the NNMi database. NNMi temporarily retains the `envattrs` name-value pairs. You can use `getEnvAttr` to retrieve a current `envattrs` value pair and pass it back to that application. Click here for an illustrated example.



Note: See "Configure Launch Actions" (on page 878) for information about adding menu items to the NNMi Actions menu.

You can send any number of Environment Attributes (`envattrs`) when launching NNMi from another website or program. You can use `getEnvAttr` to retrieve the current `envattrs` name=value pairs and pass them back:

`${getEnvAttr(<applicationAttrName>)}`

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<yourServerName>:<portNumber>/<application>?<yourURLparameter1>=`
`${getEnvAttr(<applicationAttrName1>)}&<yourURLparameter2>=`
`${getEnvAttr(<applicationAttrName2>)}`

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For example, the following Full URL provides an Action within the NNMi console that returns the user to exactly the same place within your company website where the user was prior to launching NNMi:

```
http://<myHost>/<myApplication>?com.my.sessionId=
${getEnvAttr(com.my.sessionId)}&com.my.objectName= $getEnvAttr(com.my.objectName)}
```

The Full URL entry could result in the following URL:

```
http://<myHost>/<myApplication>com.my.sessionId= 123&com.my.objectName= node25
```

Note: If the Environment Attribute that you request in your Action does not exist for the selected view or form, the resulting URL passes an empty string.

Launch the Console (showMain)

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

To launch the entire console, use the following URL:

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`showMain`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

To launch the console and bypass login, use the following URL:

`http://<serverName>:<portNumber>/nnm/launch?cmd=showMain`**`&j_username=<accountName>&j_password=<accountPassword>`**

Caution: Review in the information in "Authentication Requirements for URLs Access" before bypassing the login.

Launch a View (showView)

Tip: A view session launched with a URL never times out. (If you are using Mozilla Firefox, see also Configure Mozilla Firefox Timeout Interval.) To continuously display up-to-date information in your network operation center (NOC), launch an Integration URL view .

To launch a default table view that displays all instances of a specified object type, use the following URL:

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showView&objtype= <x>`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

Default View for Each Object Type and Available Filters

x = objtype Value	Default View	Node Filter	Interface Filter
`Incident`	Incidents workspace, All Incidents table view	Yes	No
`Node`	Inventory workspace, Nodes table view	Yes	No
`Interface`	Inventory workspace, Interfaces table view	Yes	Yes
`IPAddress`	Inventory workspace, IP Addresses table view	Yes	Yes
`IPSubnet`	Inventory workspace, IP Subnets table view	No	No
`NodeGroup`	Inventory workspace, Node Groups table view	No	No
`InterfaceGroup`	Inventory workspace, Interface Groups table view	No	No

The following are optional filter parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showView&objtype= <x>`**`&nodegroup= <Name>`**

`http://<serverName>:<portNumber>/nnm/launch?cmd= showView&objtype= <x>`**`&ifgroup= <Name>`**

Filter by Node Group (launched Incident, Node, Interface, and IP Address views)

Attribute	Values
nodegroup	The *case-sensitive* Name attribute value of the Node Group to use as a filter for this view. **Note**: The Node Group name is translated. If your team shares NNMi within multiple locales, use nodegroupid or nodegroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" (on page 954)).
nodegroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use

Attribute	Values
	the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
nodegroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

Filter by Interface Group (launched Interface and IP Address views)

Attribute	Values
ifgroup	The *case-sensitive* Name attribute value of the Interface Group to use as a filter for this view. **Note**: The Interface Group name is translated. If your team shares NNMi within multiple locales, use ifgroupid or ifgroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" (on page 954)).
ifgroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Interface Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
ifgroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Interface Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

The following are optional parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showView&objtype= <x>`**`&menus= <true|false>&newWindow= <true|false>&readonly= <true|false>&readonlynodegroupselector= <true|false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested

Attribute	Values
	view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
readonlynodegroupselector	true = Prevents the user from selecting a Node Group. false = Enables the user to select a Node Group.
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

If you want to launch some other view, specify the view rather than the object type:

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showView&view= <x>`**

For more information, see:

"Launch an Incident View" (on page 961)

"Launch a Topology Maps Workspace View" (on page 965)

"Launch a Monitoring Workspace View" (on page 972)

"Launch a Troubleshooting Workspace View" (on page 975)

"Launch an Inventory Workspace View" (on page 982)

"Launch a Management Mode Workspace Views" (on page 985)

"Launch a Configuration Workspace View" (on page 988)

Launch an Incident View

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showView&view= <x>`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

Potential Incident Workspace Views and Available Filters

View Name	*x* = View ID	Node Filter	Interface Filter
All Incidents	`allIncidentsTableView` **Tip**: To display the All Incidents view filtered by a specified node, see "Launch the All Incidents View Filtered by Node" (on page 963)	Yes	No
Closed Key Incidents	`closedKeyIncidentsTableView`	Yes	No
Custom Incidents	`customIncidentTableView`	Yes	No
Custom Open Incidents	`customOpenIncidentTableView`	Yes	No
My Open Incidents	`myIncidentTableView`	Yes	No
NNM 6.x / 7.x Events	`nnm6x7xIncidentTableView`	Yes	No
Open Key Incidents	`openKeyIncidentsTableView`	Yes	No
Open Root Cause Incidents	`openRCIncidentTableView`	Yes	No

View Name	*x* = View ID	Node Filter	Interface Filter
Service Impact Incidents	`serviceImpactIncidentTableView`	Yes	No
SNMP Traps	`snmpTrapsIncidentTableView`	Yes	No
Unassigned Open Key Incidents	`unassignedKeyIncidentsTableView`	Yes	No

The following are optional filter parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view= <x>`**`&nodegroup= <Name>`**

Filter by Node Group (launched Incident, Node, Interface, and IP Address views)

Attribute	Values
nodegroup	The *case-sensitive* Name attribute value of the Node Group to use as a filter for this view. **Note**: The Node Group name is translated. If your team shares NNMi within multiple locales, use nodegroupid or nodegroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" (on page 954)).
nodegroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
nodegroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

The following are optional parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view= <x>`**`&menus= <true|false>&newWindow= <true|false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the

Attribute	Values
	browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch the All Incidents View Filtered by Node

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showIncidents&object-identity=<Name>`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

The following are optional filter parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showIncidents`**`&object-identity= <Name>`**

Filter the All Incidents View by Node Name

Attribute	Values
Name	The *case-sensitive* Name attribute value of the Node to use as a filter for this view. **Note**: The Node Name is translated. If your team shares NNMi within multiple locales, use the `showView` command with nodegroupid or nodegroupuuid. See "Launch an Incident View" (on page 961) for more information. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" (on page 954)).

The following are optional parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showIncidents&object-identity= <Name>`**`&menus= <true|false>&newWindow= <true|false>&readonly= <true|false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external*

Attribute	Values
	application to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch a Topology Maps Workspace View

The URL required for each one is unique.

Tip: A map session launched with a URL never times out. The view automatically updates every 30 seconds. This is useful if your network operation center (NOC) continuously displays a map of the most important nodes. See "Configure Maps" (on page 279). (If you use the Mozilla Firefox browser and are prompted to click Continue before the map appears, see Configure Mozilla Firefox Timeout Interval.)

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Click here to show the example of a URL that opens the **Node Group Overview** map (cmd=showNodeGroupOverview).

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http:/<serverName>:<portNumber>/nnm/launch?`**`cmd= showNodeGroupOverview`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

The following are optional parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showNodeGroupOverview`**`&menus= <true/false>&newWindow= <true/false>&readonly= <true|false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Click here to show the example of a URL that opens the **Network Overview** map (cmd=showNetworkOverview).

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http:/<serverName>:<portNumber>/nnm/launch?`**`cmd= showNetworkOverview`**

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

<*portNumber*> = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

The following are optional parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showNetworkOverview`**`&menus= <true/false>&newWindow= <true/false>&readonly= <true|false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass <*name=value*> pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to

Attribute	Values
	preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Click here to show the example of a URL that opens the **Networking Infrastructure Devices** node group map (cmd=showView).

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

See quick reference "W3C Rules for URLs" (on page 954).

`http:/<serverName>:<portNumber>/nnm/launch?`**`cmd= showView&objtype= Node&nodegroup= Networking+Infrastructure+Devices`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

The following are optional parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showView&objtype= Node&nodegroup= Networking+Infrastructure+Devices`**`&menus= <true|false>&newWindow= <true|false>&readonly= <true|false>&readonlynodegroupselector= <true|false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu

Attribute	Values
	bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
readonlynodegroupselector	true = Prevents the user from selecting a Node Group. false = Enables the user to select a Node Group.
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Click here to show the example of a URL that opens the **Routers** node group map (cmd=showNodeGroup&name=Routers).

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http:/<serverName>:<portNumber>/nnm/launch?`**`cmd= showNodeGroup&name= Routers`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

The following are optional parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showNodeGroup&name= Routers`**`&menus= <true|false>&newWindow= <true|false>&readonly= <true|false>&readonlynodegroupselector= <true|false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
readonlynodegroupselector	true = Prevents the user from selecting a Node Group. false = Enables the user to select a Node Group.
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows:

Attribute	Values
	`http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Click here to show the example of a URL that opens the **Switches** node group map (cmd=sho-wNodeGroup&name=Switches).

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http:/<serverName>:<portNumber>/nnm/launch?`**`cmd= showNodeGroup&name= Switches`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

The following are optional parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showNodeGroup&name= Switches`**`&menus= <true|false>&newWindow= <true/false>&readonly= <true|false>&readonlynodegroupselector= <true|false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view.

Attribute	Values
	false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
readonlynodegroupselector	true = Prevents the user from selecting a Node Group. false = Enables the user to select a Node Group.
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch a Monitoring Workspace View

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showView&view= <x>`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

Monitoring Workspace Views and Available Filters

View Name	*x* = View ID	Node Filter	Interface Filter
Non-Normal Node Components	`nonNormalNodeComponentTableView`	No	No
Non-Normal Cards	`nonNormalCardTableView`	No	No
+Non-Normal Interfaces	`nonNormalInterfaceTableView`	Yes	Yes
+Non-Normal Nodes	`nonNormalNodeTableView`	Yes	No
+Not Responding Addresses	`notRespondingIPAddressTableView`	Yes	Yes
Interface Performance	`interfacePerformanceTableView`	Yes	Yes
Card Redundancy Groups	`cardRedundancyGroupsTableView`	No	No
Router Redundancy Groups	`routerRedundancyGroupsStatusTableView`	No	No
Node Groups	`nodeGroupsStatusTableView`	No	No
Custom Node Collections	`customPollerNodeCollectionsTableView`	No	No
Custom Polled Instances	`customPollerPolledInstancesTableView`	No	No

The following are optional filter parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view= <x>&nodegroup= <Name>`

`http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view= <x>&ifgroup= <Name>`

Filter by Node Group (launched Incident, Node, Interface, and IP Address views)

Attribute	Values
nodegroup	The *case-sensitive* Name attribute value of the Node Group to use as a filter for this view. **Note**: The Node Group name is translated. If your team shares NNMi within multiple locales, use nodegroupid or nodegroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" (on page 954)).
nodegroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use

Attribute	Values
	the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
nodegroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

Filter by Interface Group (launched Interface and IP Address views)

Attribute	Values
ifgroup	The *case-sensitive* Name attribute value of the Interface Group to use as a filter for this view. **Note**: The Interface Group name is translated. If your team shares NNMi within multiple locales, use ifgroupid or ifgroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" (on page 954)).
ifgroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Interface Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
ifgroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Interface Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

The following are optional parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view= <x>`**`&menus= <true|false>&newWindow= <true|false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).

Attribute	Values
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch a Troubleshooting Workspace View

There are four types of views in the Troubleshooting workspace. The URL syntax required for each one is unique.

Tip: A map session launched with a URL never times out. The view automatically updates every 30 seconds. This is useful if your network operation center (NOC) continuously displays a map of the most important nodes. See "Configure Maps" (on page 279). (If you use the Mozilla Firefox browser and are prompted to click Continue before the map appears, see Configure Mozilla Firefox Timeout Interval.)

Click here to show examples of URLs that open a **Layer 2 Neighbor View** (cmd=showLayer2Neighbors).

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showLayer2Neighbors`**

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showLayer2Neighbors&nodename= <x>&hops= <#>`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

Layer 2 Neighbor View Attributes

Attribute	Value
nodename	The source node's DNS hostname (full or short) or IP address. If you use this attribute, NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.
hops	1 - 9

The following are optional parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showLayer2Neighbors`**`&menus= <true|false>&newWindow= <true|false>&readonly= <true|false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)

Attribute	Values
	false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Click here to show examples of URLs that open a **Layer 3 Neighbor View** (cmd=showLayer3Neighbors).

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showLayer3Neighbors`**

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showLayer3Neighbors&nodename= <x>&hops= <#>`**

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

Layer 3 Neighbor View Attributes

Attribute	Value
nodename	The source node's DNS hostname (full or short) or IP address. If you use this attribute, NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.
hops	1 - 9
menus	true = Show the menus and window toolbar in the form. If not specified, the default is true. false = Hide the menus and window toolbar in the view.

The following are optional parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showLayer3Neighbors`**`&menus= <true|false>&newWindow= <true|false>&readonly= <true|false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program).

Attribute	Values
	For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Click here to show examples of URLs that open a **Path View** (cmd=showPath).

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showPath`**

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showPath&src= <x>&dest= <y>`**

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

Note: *NNMi Advanced*. Path View works only with IPv4 addresses. The NNMi Advanced IPv6 address values are not valid choices for Path View. Any devices in your network that are configured with IPv6 addresses cannot be displayed on Path View maps.

Path View Attributes

Attribute	Value
src	The source node's DNS hostname (full or short) or IP address. NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.
dest	The destination node's DNS hostname (full or short) or IP address. NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab.

Attribute	Value
	• Check the value in the Name field on the Node form.

The following are optional parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showPath&menus= <true|false>&newWindow= <true|false>&readonly= <true|false>&envattrs= <name1= value>;<name2= value>`

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Click here to show examples of URLs that open a **Node Group Map View** (cmd=showNodeGroup).

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is

available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showNodeGroup&name= <x>`**

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

Node Group Map View Attributes

Attribute	Value
name	The *case-sensitive* Name attribute value from the Node Group form. **Note**: The Node Group name is translated. If your team shares NNMi within multiple locales, use nodegroupid or nodegroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" (on page 954)).
objid	The `id` is the Unique Object Identifier (unique per object type in the NNMi database). Provide the `id` of the Node Group to use as a filter for this view. NNMi displays the `id` attribute value on the object form's Registration tab.
objuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Node Group to use as a filter for this view. NNMi displays the `uuid` attribute value on the object form's Registration tab.

The following are optional parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showNodeGroup&name= <x>`**`&menus= <true|false>&newWindow= <true|false>&readonly= <true|false>&readonlynodegroupselector= <true|false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).

Attribute	Values
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
readonlynodegroupselector	true = Prevents the user from selecting a Node Group. false = Enables the user to select a Node Group.
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch an Inventory Workspace View

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showView&view= <x>`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following

limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

Inventory Workspace Views and Available Filters

View Name	*x* = View ID	Node Filter	Interface Filter
Nodes	`allNodesTableView`	Yes	No
Interfaces	`allInterfacesTableView`	Yes	Yes
IP Addresses	`allIPAddressTableView`	Yes	Yes
IP Subnets	`allIPSubnetsTableView`	No	No
VLANs	`allVlansTableView`	No	No
Cards	`allCardsTableView`	No	No
Ports	`allPortsTableView`	No	No
Nodes by Management Server	`nodesByNNMiManagementServerTableView`	No	No
Custom Nodes	`customNodeTableView`	Yes	No
Custom Interfaces	`customInterfaceTableView`	Yes	Yes
Custom IP Addresses	`customIPAddressTableView`	Yes	Yes
MIB Variables	`mibVariablesTableView`	No	No
Card Redundancy Groups	`allCardRedundancyGroupsTableView`	No	No
Router Redundancy Groups	`routerRedundancyGroupsTableView`	No	No
Node Groups	`nodeGroupsTableView`	No	No
Interface Groups	`interfaceGroupsTableView`	No	No
Management Stations (6.x/7.x)	`allManagementStationsTableView`	No	No

The following are optional filter parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view= <x>&nodegroup= <Name>`

`http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view= <x>&interfacegroup= <Name>`

Filter by Node Group (launched Incident, Node, Interface, and IP Address views)

Attribute	Values
nodegroup	The *case-sensitive* Name attribute value of the Node Group to use as a filter for this view.

Attribute	Values
	Note: The Node Group name is translated. If your team shares NNMi within multiple locales, use nodegroupid or nodegroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" (on page 954)).
nodegroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
nodegroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

Filter by Interface Group (launched Interface and IP Address views)

Attribute	Values
ifgroup	The *case-sensitive* Name attribute value of the Interface Group to use as a filter for this view. **Note**: The Interface Group name is translated. If your team shares NNMi within multiple locales, use ifgroupid or ifgroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" (on page 954)).
ifgroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Interface Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
ifgroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Interface Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

The following are optional parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view= <x>`**`&menus= <true/false>&newWindow= = <true/false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true.

Attribute	Values
	false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch a Management Mode Workspace Views

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showView&view= <x>`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following

limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

Management Mode Workspace Views

View Name	*x* = View ID	Node Filter	Interface Filter
Unmanaged[1] Nodes	`unManagedNodeTableView`	Yes	No
Unmanaged[2] Interfaces	`unManagedInterfaceTableView`	Yes	Yes
Unmanaged[3] IP Addresses	`unManagedIPAddressTableView`	Yes	Yes
Unmanaged[4] Cards	`unManagedCardTableView`	Yes	No
Unmanaged[5] Node Components	`unManagedNodeComponentTableView`	Yes	No

The following are optional filter parameters:The following are optional parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view= <x>`**`&nodegroup= <Name>`**

`http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view= <x>`**`&ifgroup= <Name>`**

Filter by Node Group (launched Incident, Node, Interface, and IP Address views)

Attribute	Values
nodegroup	The *case-sensitive* Name attribute value of the Node Group to use as a filter for this view. **Note**: The Node Group name is translated. If your team shares NNMi within multiple locales, use nodegroupid or nodegroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" (on page 954)).
nodegroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.

[1]Indicates the Management Mode is "Not Managed" or "Out of Service".
[2]Indicates the Management Mode is "Not Managed" or "Out of Service".
[3]Indicates the Management Mode is "Not Managed" or "Out of Service".
[4]Indicates the Management Mode is "Not Managed" or "Out of Service".
[5]Indicates the Management Mode is "Not Managed" or "Out of Service".

Attribute	Values
nodegroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

Filter by Interface Group (launched Interface and IP Address views)

Attribute	Values
ifgroup	The *case-sensitive* Name attribute value of the Interface Group to use as a filter for this view. **Note**: The Interface Group name is translated. If your team shares NNMi within multiple locales, use ifgroupid or ifgroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" (on page 954)).
ifgroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Interface Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
ifgroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Interface Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

`http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view= <x>`**`&menus= <true/false>&newWindow= = <true/false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)

Attribute	Values
	false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch a Configuration Workspace View

Configuration workspaces require that the user be assigned to the **Administrative** role.

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showView&view= <x>`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

Configuration Workspace Views

View Name	*x* = View ID
Node Groups	`nodeGroupsTableView`
Interface Groups	`interfaceGroupsTableView`
IfTypes	`allIfTypesTableView`
Device Profiles	`allDeviceProfilesTableView`
Loaded MIBs	`loadedMibsTableView`
MIB Expressions	`mibExpressionsTableView`
RAMS Servers	`ramsServerTableView`
Management Stations (6.x/7.x)	`allManagementStationsTableView`

The following are optional parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view= <x>`**`&menus= <true/false>&newWindow= = <true/false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs=`

Attribute	Values
	`com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch a Form (showForm)

To launch a particular form, use the following URL:

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?`**`cmd=showForm...`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Launch a form to see information about a particular node, interface, address, subnet, or incident. In the URL string, you must include one or more attributes that enable NNMi to find a specific object. If more than one object meets the criteria, NNMi opens the first one found. When designating more than one attribute, separate each with a semicolon character.

"Launch a Node Form" (on page 990)

"Launch an Interface Form" (on page 993)

"Launch an IP Address Form" (on page 994)

"Launch a Subnet Form" (on page 995)

"Launch an Incident Form" (on page 997)

"Launch a Node Group Form" (on page 998)

"Launch a Configuration Form" (on page 1000)

Launch a Node Form

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showForm&objtype= Node&nodename= <x>`**

`http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype= Node&`**`objattrs= name= <x>`**

`http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype= Node&`**`objattrs= hostname= <x>`**

`http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype= Node&`**`objattrs= snmpAgent.agentSettings.managementAddress= <x>`**

`http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype= Node&`**`objattrs= systemName= <x>`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi form with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.
- The Save and Close button saves data, but does not close the window.
- The File → Save and Close menu item saves data, but does not close the window.

Use the browser buttons to close the form.

Node Form Attributes

Attribute	Values
nodename	Provide the node's DNS hostname (full or short) or IP address. If you use this attribute, NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.
name	The Name attribute value from the Node form.
hostname	The *case-sensitive* Hostname attribute value from the Node form of the discovered node must match what is entered here. NNMi follows a set of rules to dynamically generate the value stored in the NNMi database for each Node's Hostname. Click here for details. **Note**: The actual Hostname *might be converted* to all uppercase or all lowercase before it is added to the NNMi database (depending on

Attribute	Values
	how the NNMi administrator configured settings in the `nms-topology.properties` file). See the information about the `nms-topology.properties` file in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`. • If the Node supports SNMP, NNMi requests the Hostname using the IP Address of the associated SNMP agent (the Management Address attribute value on the Node form). If the NNMi administrator chooses ☑ **Enable SNMP Address Rediscovery** in the Communication Configuration: ▪ If the SNMP Agent does not respond, NNMi checks for another Management Address to request the Hostname, and the Hostname could change. ▪ If the SNMP Agent associated with the node changes, the Management Address and Hostname could change. If the NNMi administrator disables ☐ **Enable SNMP Address Rediscovery** in the Communication Configuration: ▪ If the SNMP Agent does not respond, NNMi uses the previously gathered Management Address attribute value to request the Hostname. ▪ If the SNMP Agent associated with the node changes, NNMi uses the previously gathered Management Address attribute value to request the Hostname. • If the Node does not support SNMP, no Management Address is available. NNMi requests a Hostname starting with the lowest IP Address associated with the node (a Discovery Seed value or an IP address value gathered from a neighboring device). NNMi uses the first Hostname provided. The Hostname might change during a future discovery cycle.
snmpAgent.agentSettings.managementAddress	The Management Address attribute value from the SNMP Agent form of the agent assigned to the specified node. The value is an IP address.
systemName	System Name attribute value from the Node form, General tab.

The following are optional filter parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype= Node&nodename=<x>&menus= <true/false>&envattrs= <name1= value>;<name2= value>`

Attributes for Launched Forms

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the

Attribute	Values
	external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch an Interface Form

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showForm&objtype= Interface&objattrs= hostedOn.hostname= <x>;name= <y>`**

`http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype= Interface&objattrs= hostedOn.hostname= <x>;`**`ifName= <y>`**

`http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype= Interface&objattrs= hostedOn.hostname= <x>;`**`ifAlias= <y>`**

`http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype= Interface&objattrs= hostedOn.hostname= <x>;`**`ifIndex= <y>`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi form with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.
- The Save and Close button saves data, but does not close the window.
- The File → Save and Close menu item saves data, but does not close the window.

Use the browser buttons to close the form.

Interface Form Attributes

Attribute	Values
hostedOn.hostname	The *case-sensitive* Hostname of the Node in which the interface resides. This is the Hostname attribute value from the associated Node form.
name	The Name attribute value from the Interface form.
ifName	The IfName attribute value from the Interface form.
ifAlias	The IfAlias attribute value from the Interface form.
ifIndex	The IfIndex attribute value from the Interface form.

The following are optional filter parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype= Interface&objattrs= hostedOn.hostname= <x>;name= <y>`**`&menus= <true/false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Forms

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch an IP Address Form

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showForm&objtype= IPAddress&objattrs= value= <y>`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi form with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.
- The Save and Close button saves data, but does not close the window.
- The File → Save and Close menu item saves data, but does not close the window.

Use the browser buttons to close the form.

IP Address Form Attributes

Attribute	Values
value	The Address attribute value from the IP Address form.

The following are optional filter parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype= IPAddress&objattrs= value= <y>`**`&menus= <true/false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Forms

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch a Subnet Form

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is

available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showForm&objtype= IPSubnet&objattrs= name= <x>`**

`http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype= IPSubnet&`**`objattrs= prefix= <x>`**

`http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype= IPSubnet&`**`objattrs= prefix= <x>;prefixLength= <y>`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi form with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.
- The Save and Close button saves data, but does not close the window.
- The File → Save and Close menu item saves data, but does not close the window.

Use the browser buttons to close the form.

IP Subnet Form Attributes

Attribute	Values
name	The *case-sensitive* Name attribute value from the IP Subnet form.
prefix	The Prefix attribute value from the IP Subnet form.
prefixLength	The Prefix Length attribute value from the IP Subnet form.

The following are optional filter parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype= IPSubnet&objattrs= name= <x>`**`&menus= <true/false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Forms

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external*

Attribute	Values
	application to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch an Incident Form

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showForm&objtype= Incident&objid= <x>`**

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showForm&objtype= Incident&objuuid= <x>`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi form with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.
- The Save and Close button saves data, but does not close the window.
- The File → Save and Close menu item saves data, but does not close the window.

Use the browser buttons to close the form.

Individual incident objects must be identified by their *database unique identifiers*.

Incident Attributes

Attribute	Values
objid	The Unique Object Identifier (unique per object type in the NNMi database). NNMi displays the `id` attribute value on the object form's Registration tab.

Attribute	Values
objuuid	The Universally Unique Object Identifier (unique across all databases). NNMi displays the `uuid` attribute value on the object form's Registration tab.

The following are optional filter parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&showForm&objtype= Incident&objid= <x>`**`&menus= <true/false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Forms

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch a Node Group Form

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showForm&objtype= NodeGroup&name= <y>`**

`http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype= NodeGroup`**`&nodegroupid= <y>`**

`http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype= NodeGroup`**`&nodegroupuuid= <y>`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi form with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.
- The Save and Close button saves data, but does not close the window.
- The File → Save and Close menu item saves data, but does not close the window.

Use the browser buttons to close the form.

Node Group Form Attributes

Attribute	Values
name	The *case-sensitive* Name attribute value from the Node Group form. **Note**: The Node Group name is translated. If your team shares NNMi within multiple locales, use nodegroupid or nodegroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" (on page 954)).
nodegroupid	The `id` is the Unique Object Identifier (unique per object type in the NNMi database). Provide the `id` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
nodegroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

The following are optional filter parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype= NodeGroup&name=<y>`**`&menus= <true/false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Forms

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows:

Attribute	Values
	`http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch a Configuration Form

Configuration forms require that the user be assigned to the **Administrative** role.

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showConfigForm&name= <y>`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi form with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.
- The Save and Close button saves data, but does not close the window.
- The File → Save and Close menu item saves data, but does not close the window.

Use the browser buttons to close the form.

Configuration Form Attributes

Attribute	Values
name	The name attribute value specifies which form: • **customcorrelation** = the Custom Correlation Configuration • **communication** = the Communication Configuration form • **custompoller** = the Custom Poller Configuration form • **discovery** = the Discovery Configuration form • **globalnetworkmanagement** = the Global Network Management form • **monitoring** = the Monitoring Configuration form

Attribute	Values
	• **incident** = the Incident Configuration form • **status** = the Status Configuration form • **trap** = the Trap Forward Configuration form • **ui** = the User Interface Configuration form

The following are optional filter parameters:

`http://<serverName>:<portNumber>/nnm/launch?cmd= showConfigForm&name= <x>`**`&menus= <true/false>&envattrs= <name1= value>;<name2= value>`**

Attributes for Launched Forms

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note**: See "Pass Environment Attributes" (on page 955) for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch Menu Items (runTool)

To launch a menu item, use the following URL:

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`runTool&tool=<x>`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

Provide quick access to NNMi menu items wherever your team needs them:

"Launch the Actions: Communication Configuration Command" (on page 1002)

"Launch the Actions: Configuration Poll Command" (on page 1003)

"Launch the Actions: Line Graph (showLineGraph)" (on page 1004)

"Launch the Actions: Monitoring Settings Command" (on page 1006)

"Launch the Actions: Ping Command" (on page 1009)

"Launch the Actions: Status Details Command (for Node Groups)" (on page 1010)

"Launch the Actions: Status Poll Command" (on page 1011)

"Launch the Actions: Trace Route Command" (on page 1012)

"Actions: Execute a Launch Action" (on page 1013)

"Launch the Tools: MIB Browser (showMibBrowser)" (on page 1014)

"Launch the Tools: NNMi Status Command" (on page 1015)

"Launch the File: Sign-Out Command" (on page 1016)

"Launch the Tools: Sign-In/Out Audit Log Command" (on page 1016)

Launch the Actions: Communication Configuration Command

This URL is equivalent to the **Actions → Communication Settings** command in the console.

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

To launch a window that reports the current ICMP and SNMP configuration for a node, use the following URL:

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`runTool&tool=commconf`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

After you specify a node, the real-time results of the ICMP and SNMP configuration report appear.

To launch the real-time results of the ICMP and SNMP configuration report, use the following URL:

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`runTool&tool=commconf&nodename=<x>`**

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`runTool&tool=commconf&IPAddress=<x>`**

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`runTool&tool=commconf&Interface=<x>`**

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`runTool&tool=commconf&snmpAgent=<x>`**

Communication Configuration Command Attributes

Attribute	Values
nodename	The node's DNS hostname (full or short) or IP address. If you use this attribute, NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.

Related Topics:

"Troubleshooting Communication Settings" (on page 108)

Launch the Actions: Configuration Poll Command

This URL is equivalent to the **Actions** → **Configuration Poll** command in the console.

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

To launch a window that reports the current configuration for a node, use the following URL:

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`runTool&tool=configurationpoll`**

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

<portNumber> = the port that the jboss application server uses for communicating with the NNMi console

After you specify a node, the real-time results of the node's configuration appear.

To launch the real-time results of a node's configuration, use the following URL:

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`runTool&tool=configurationpoll&nodename=<x>`**

Configuration Poll Command Attributes

Attribute	Values
nodename	The node's DNS hostname (full or short) or IP address.

Attribute	Values
	If you use this attribute, NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.

Related Topics:

Verify Device Configuration Details

Launch the Actions: Line Graph (showLineGraph)

Use the showLineGraph URL to launch a Line Graph that displays real-time SNMP data about a selected object. See "Configure SNMP Line Graph Actions" (on page 890).

Note: If you are displaying graphs for NNMi objects, the node or interface for which you want to graph information must support SNMPv1, SNMPv2C, or SNMPv3.

Use the showLineGraph syntax in a URL when you want to do any of the following:

- Display a Line Graph in an application other than NNMi.
- Display a Line Graph outside of an application and add it to your Favorites browser list.

To launch a Line Graph with the showLineGraph syntax, use the following URL:

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`showLineGraph`** `[parameter list]`

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`showLineGraph`**
`&init=<x>&objtype=<x> &maxlines=<x> &maxtimerange=<x> &defaultsecs=`
`<x>&faststart=<true/false>`

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Line Graph Parameters

Parameter	Description
`&init`	Use to define the lines you want displayed in the graph: • `instancelist` - Use to specify which instances of the SNMP MIB object to display. • One of the following for each line:

Parameter	Description
	▪ `oid` - Use to specify the SNMP MIB object identifier value of each instance. ▪ `expr` - Use to specify the name of a MIB Expression that will be used for gathering the values on the Line Graph. • `label` - Use to specify the label to be used in the legend that describes each line on the graph.
`&objtype`	Use to specify the Object Type. For a Node Object Type, this value must be `${snmpAgent.id}`. For an Interface Object Type, this value must be `${hostedOn.snmpAgent.id}` **Note**: If you want to provide a Line Graph for a specified node, use the ID value for the node's SNMP Agent. NNMi displays the ID attribute value on the SNMP Agent form's Registration tab.
`&maxlines`	Use to specify the number of lines that NNMi should initially display on the Line Graph. To use the default value specified in the User Interface Configuration, omit this parameter.
`&maxtimerange`	Use to specify the number of hours for the Maximum Time Range in which the data in the Line Graph should be retained. After the Maximum Time Range number is reached, NNMi discards the oldest data point sets so that it can display the most recent data for the time range specified.
`&defaultsecs`	Use to specify the Polling Interval in which the graph data should be collected. To use the default value specified in the User Interface Configuration, omit this parameter.
`&faststart`	Use to specify whether to increase the initial Polling Interval so that the initial data appears more quickly on the graph. Possible values are `true` or `false`. When you specify `true` for this option, NNMi increases the initial Polling Interval and then gradually decreases the Polling Interval until it reaches the Polling Interval configured for the graph. When you specify `false`, NNMi uses the Polling Interval set for the graph.
`&defaultfixedvertical`	Used to specify whether to lock the Y-axis. Possible values are `true` or `false`. When you specify `true`, the Y-axis remains fixed at the minimum and maximum values for the current set of data regardless of the time segment selected. This means NNMi does not automatically re-adjust the Y-axis to match the data values for the selected time segment. When you specify `false`, NNMi automatically adjusts the Y-axis to match the data values for the selected time segment.
`&ylabel`	Use to specify the label to be used for the Y-axis of the Line Graph.

Parameter	Description
more...	HP Network Node Manager i Software Smart Plug-ins (iSPIs) might provide more attributes to customize the line graph. See the documentation for the NNM iSPIs installed in your network environment.

Launch the Actions: Monitoring Settings Command

This URL is equivalent to the **Actions → Monitoring Settings** command in the console.

Launch the real-time results of the Monitoring configuration report. You must specify the target object.

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

<*portNumber*> = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

(*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager:

- Node managed by the Global Manager = Actions → Monitoring Settings displays a report, provided by the Global Manager (NNMi management server).
- Node managed by a Regional Manager = Actions → Monitoring Settings accesses that Regional Manager (NNMi management server) and requests the report.

  Note: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.

To launch a window that displays a current Monitoring configuration report about a Node, use the following URL:

`http://<`
`serverName`
`>:<`
`portNumber`
`>/nnm/launch?cmd=`**`runTool&tool=monitoringconf&objtype=SnmpAgent&nodename=<x>`**

Monitoring Configuration Command Node Report Attributes

Attribute	Values
nodename	The node's DNS hostname (full or short) or IP address. If you use this attribute, NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Node form, Addresses tab,

Attribute	Values
	• Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.

To launch a window that displays a current Monitoring configuration report about an Interface, use one of the following URLs:

NNMi displays the report for the first matching Interface found. Provide one or more attributes to ensure a unique match. See "Launch an Interface Form" (on page 993) for more information about each available attribute.

```
http://<
serverName
>:<
portNumber
>/nnm/launch?cmd=
runTool&tool=monitoringconf&objtype=Interface&objattrs=hostedOn.hostname=<x>;name=<x>
```

```
http://<
serverName
>:<
portNumber
>/nnm/launch?cmd=runTool&tool=monitoringconf&objtype=Interface&objattrs=
hostedOn.hostname=<x>;ifName=<x>
```

```
http://<
serverName
>:<
portNumber
>/nnm/launch?cmd=runTool&tool=monitoringconf&objtype=Interface&objattrs=
hostedOn.hostname=<x>;ifAlias=<x>
```

```
http://<
serverName
>:<
portNumber
>/nnm/launch?cmd=runTool&tool=monitoringconf&objtype=Interface&objattrs=
hostedOn.hostname=<x>;ifIndex=<x>
```

Interface Form Attributes

Attribute	Values
hostedOn.hostname	The Hostname of the Node in which the interface resides. This is the Hostname attribute value from the associated Node form.
name	The Name attribute value from the Interface form.
ifName	The ifName attribute value from the Interface form.
ifAlias	The ifAlias attribute value from the Interface form.
ifIndex	The ifIndex attribute value from the Interface form.

To launch a window that displays a current Monitoring configuration report about an IP Address, use the following URL:

```
http://<
```

serverName
>:<
portNumber
>/nnm/launch?cmd=**runTool&tool=monitoringconf&objtype=IPAddress&objattrs=value=<x>**

IP Address Form Attributes

Attribute	Values
value	The Address attribute value from the IP Address form.

To launch a window that displays a current Monitoring configuration report about a Router Redundancy Member (Instance), use the following URL:

http://<
serverName
>:<
portNumber
>/nnm/launch?
cmd=runTool&tool=monitoringconf&objtype=RouterRedundancyInstance&objid=<x>

Monitoring Configuration Command Router Redundancy Member Report Attributes

Attribute	Values
objid	The Unique Object Identifier (unique per object type in the NNMi database). This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
objuuid	The Universally Unique Object Identifier (unique across all databases). This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

To launch a window that displays a current Monitoring configuration report about a Tracked Object, use the following URL:

http://<
serverName
>:<
portNumber
>/nnm/launch?**cmd=runTool&tool=monitoringconf&objtype=TrackedObject&objid=<x>**

Monitoring Configuration Command Tracked Object Report Attributes

Attribute	Values
objid	The Unique Object Identifier (unique per object type in the NNMi database). This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
objuuid	The Universally Unique Object Identifier (unique across all databases). This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

Attribute	Values

To launch a window that displays a current Monitoring configuration report about a Node Component, use the following URL:

`http://<`*serverName*`>:<`*portNumber*`>/nnm/launch?`**`cmd=runTool&tool=monitoringconf&objtype=ComponentHealth&objid=<x>`**

Monitoring Configuration Command Node Component Report Attributes

Attribute	Values
objid	The Unique Object Identifier (unique per object type in the NNMi database). NNMi displays the `id` attribute value on the Node form's Registration tab.
objuuid	The Universally Unique Object Identifier (unique across all databases). NNMi displays the `uuid` attribute value on the Node form's Registration tab.

Related Topics:

"Verify Monitoring Configuration Settings" (on page 246)

Launch the Actions: Ping Command

This URL is equivalent to the **Actions → Ping (from server)** command in the console.

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

To launch a window that requests you to enter a node name, use the following URL:

`http://<`*serverName*`>:<`*portNumber*`>/nnm/launch?cmd=`**`runTool&tool=ping`**

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

<*portNumber*> = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

After you specify a node, the real-time results of the ping command appear.

To launch the real-time results of the ping command, use the following URL:

`http://<`*serverName*`>:<`

portNumber
>/nnm/launch?cmd=**runTool&tool=ping&timeoutSecs=<x>&numPings=<x>&nodename=<x>**

(*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager:

- Node managed by the Global Manager = Actions → Ping issues an ICMP request from the Global Manager (NNMi management server).
- Node managed by a Regional Manager = Actions → Ping accesses that Regional Manager (NNMi management server) and issues the ICMP request.

  Note: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager.

Ping Command Attributes

Attribute	Values
nodename	A DNS-resolvable hostname or IP address. The nodename value is passed literally to the underlying command you specify. The nodename value (in this case) is not required to correlate with anything in the NNM database.
timeoutSecs	Amount of time NNMi waits before abandoning a ping request.
numPings	Maximum number of retries.

Related Topics:

Test Node Access (Ping)

Launch the Actions: Status Details Command (for Node Groups)

This URL is equivalent to the **Actions** → **Status Details** command in the console.

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

To launch a real-time calculation of current status for a specified Node Group, use the following URL:

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`runTool&tool=nodegroupstatus`**

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

<*portNumber*> = the port that the jboss application server uses for communicating with the NNMi console

After you specify a node group, the real-time results of the node group's status calculation appear.

To launch a real-time calculation of current status for a specified Node Group and display a report of the information gathered, use the following URL:

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`runTool&tool=nodegroupstatus&nodegroup=<x>`**

Filter by Node Group (launched Incident, Node, Interface, and IP Address views)

Attribute	Values
nodegroup	The *case-sensitive* Name attribute value of the Node Group to use as a filter for this view. **Note**: The Node Group name is translated. If your team shares NNMi within multiple locales, use nodegroupid or nodegroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" (on page 954)).
nodegroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
nodegroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

Related Topics:

Check Status Details for a Node Group

Launch the Actions: Status Poll Command

This URL is equivalent to the **Actions** → **Status Poll** command in the console.

NNMi calculates the status of devices each time additional information is gathered from various sources. You can instruct NNMi to gather real-time data for all the information that NNMi uses to calculate Status for the specified Node. A window displays with a report about which information was gathered. The NNMi administrator determines the list of information gathered by establishing Monitoring configuration settings. See "Monitoring Network Health" (on page 215) for more information.

Note: To see the resulting Node status, see Verify Current Status of a Device.

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

To launch a window that reports the current status for a node, use the following URL:

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`runTool&tool=statuspoll`**

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

<*portNumber*> = the port that the jboss application server uses for communicating with the NNMi console

After you specify a node, the real-time results of the node's status appear.

To launch the real-time results of a node's status, use the following URL:

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`runTool&tool=statuspoll&nodename=<x>`**

Status Poll Command Attributes

Attribute	Values
nodename	The node's DNS hostname (full or short) or IP address. If you use this attribute, NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.

Related Topics:

Verify Current Status of a Device

Launch the Actions: Trace Route Command

This URL is equivalent to the **Actions → Trace Route (from server)** command in the console.

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

To launch a window that requests you to enter a node name, use the following URL:

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`runTool&tool=traceroute`**

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

<*portNumber*> = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

After you specify a node, the real-time results of the trace route command appear.

To launch the real-time results of the trace route command, use the following URL:

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`runTool&tool=traceroute&nodename=<x>`**

Trace Route Command Attributes

Attribute	Values
nodename	A DNS-resolvable hostname or IP address. The nodename value is passed literally to the underlying command you specify. The nodename value (in this case) is not required to correlate with anything in the NNM database.

Related topics:

Find the Route (traceroute)

Actions: Execute a Launch Action

The showMenuItem command launches a Menu Item that has been configured as a Launch Action in NNMi.

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

To execute a Launch Action configured in NNMi:

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`showMenuItem&key=<MenuItemKey>[&nodename=<hostname or IP_address>`**

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

<portNumber> = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

After you specify a *case-sensitive* Hostname, the real-time results of the Launch Action appear.

Trace Route Command Attributes

Attribute	Values
MenuItemKey	The Unique Key used for the Menu Item configuration. See "Configure Menu Item Basic Details" (on page 874) for more information.
nodename	*Optional*. A DNS-resolvable hostname or IP address indicating the node on which the action should be executed.

See "Configure Launch Actions" (on page 878) for information about creating a Launch Action.

Launch the Tools: MIB Browser (showMibBrowser)

Use the showMibBrowser URL to display the MIB Browser window and the SNMP getNext responses from one Node.

You must provide the Node name/IP-address and one MIB variable OID (Object Identifier) value to determine the starting point. This starting point can be any OID from any MIB file that is currently loaded onto the NNMi management server and supported by the SNMP agent for the specified node.

NNMi automatically gathers responses to all MIB objects from the designated OID down though the Internet MIB tree.

NNMi enables you to launch the SNMP MIB Browser in any of the following ways:

- Use the showMibBrowser syntax in a URL as described in this help topic.
- Click **Tools → SNMP MIB Browser** ("Determine the MIB Variables Supported for a Node (for Administrators)" (on page 913)
- Click **Actions → Browse MIB** ("Determine the MIB Variables Supported for a Node (for Administrators)" (on page 913)

To launch an SNMP MIB Browser, use the following URL:

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<`*`serverName`*`>:<`*`portNumber`*`>/nnm/launch?cmd=`**`showMibBrowser`**`&node=<`*`name|address`*`>&oid=<`*`name|number`*`>`

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

<*portNumber*> = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

SNMP MIB Browser Parameters

Parameter	Description
node	The node's DNS hostname (full or short) or IP address. If you use this attribute, NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Addresses tab in the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.
oid	Enter one of the following as defined in the associated MIB file. NNMi uses this OID as the starting displayed value and the starting point for the SNMP Walk feature in the SNMP MIB Browser window: • Textual name of the object, for example `.iso.org.dod.internet.mgmt.mib-2.ifMIB.ifMIBObjects.ifTestTable.ifTestEntry.ifTestResult` or `.1.3.6.1.2.1.31.1.3.1.ifTestResult`. • Numeric representation of the object as defined in the MIB file, for example `.1.3.6.1.2.1.31.1.3.1.4` (the numeric value must always begin with a period character). **Tip**: You can view the list of MIB variables provided by each available MIB file using the **Loaded MIBs** view. See "Loaded MIBs View" (on page 912) for more information.
Community String	*Optional*. In the **Community String** attribute, enter a valid SNMPv1 *read community string* for the Node. • If you provide a *read community string*, NNMi uses SNMPv1 communication protocol. • If none is provided, NNMi checks the NNMi database for any configured SNMP read community string for that Node.
Walk	After the MIB Browser window displays, click the Walk button to issue an SNMP getNext for all MIB objects from the designated OID down though the Internet MIB tree.

Launch the Tools: NNMi Status Command

This URL is equivalent to the **Tools → NNMi Status** command in the console.

To launch a report of the current status of all NNMi processes and services, use the following URL:

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`runTool&tool=nnmstatus`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Related Topics:

"Verify that NNMi Processes Are Running" (on page 45)

Check the Status of NNMi

"NNMi Processes and Services" (on page 44)

Launch the File: Sign-Out Command

This URL is equivalent to the **File → Sign Out** command in the console.

To provide a link that issues a sign-out command, use the following URL:

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`signOut`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

This closes the user session and frees up any memory associated with the session.

Related Topics:

"Sign Out from the Console" (on page 278)

Launch the Tools: Sign-In/Out Audit Log Command

This URL is equivalent to the **Tools → Sign In/Out Audit Log** command in the console.

To launch a window that reports the current configuration for a node, use the following URL:

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`runTool&tool=signinaudit`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

NNMi logs the history of sign-in and sign-out activity for each user since the NNMi management server was last restarted.

Related Topics:

"Audit NNMi User Activity" (on page 274)

Confirm that NNMi Is Running (cmd=isRunning)

To launch a message reporting whether NNMi is currently running, use the following URL:

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. (See the"Enabling https for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.)

`http:/<serverName>:<portNumber>/nnm/launch?cmd=`**`isRunning`**

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" (on page 259))

`<portNumber>` = the port that the jboss application server uses for communicating with the NNMi console

One of the following messages appears:

- NNMi is running.
- A browser error message that the URL is unreachable.

Maintaining NNMi

As an NNMi administrator, you will want to perform the following tasks when maintaining NNMi configurations and data.

"Track Your NNMi Licenses" (on page 1019)

"Extend a Licensed Capacity" (on page 1020)

"Export and Import Configuration Settings" (on page 1022)

"Back Up and Restore NNMi" (on page 1036)

"Archive and Delete Incidents" (on page 1037)

Check NNMi Health

As an NNMi administrator, you can check the status and overall health of NNMi using any of the following:

- Use the **Tools** → **NNMi System Health Report** to access detailed information regarding NNMi health, including NNMi's overall health status as well as information and any issues related to the following:
  - Memory
  - The NNMi database
  - System resources
  - Disk usage
  - SNMP requests and queues
  - Global Network Management (NNMi Advanced)

  Click here for more information about NNMi's overall health status.

  NNMi uses the following statuses when monitoring its health:

  NNMi Overall Health Status

Status	Description
Warning	Indicates performance issues that are not significantly affecting NNMi.
Minor	Indicates problems that might result in out of date data. For example, a component, such as State Poller might be out of synch because it is operating outside of expected ranges.
Major	Indicates problems that are significantly affecting the NNMi management server's operations, but are not yet critical. Major Status usually indicates that some action is required. For example, a trap threshold is reached.
Critcal	Indicates NNMi is not functioning. For example, NNMi is out of memory, all database connections are lost, or a major component has failed.

- Use **Help** → **System Information** to view NNMi health and component information details. See Displaying NNMi System Information for more information.
- Use the **Tools** → **NNMi Self-Monitoring Graphs** to view information about NNMi components and their

usage. NNMi Self-Monitoring Graphs include:

- SNMP Trap Pipeline Rate
- SNMP Trap Forwarding Rate
- Discovery Progress
- SNMP Requests

Tip: Use the SNMP Requests Graph to tune Communication Configuration settings.

- NNMi administrators can use the command line on any NNMi management server to generate a report about NNMi health. See the nnmhealth.ovpl Reference Page for more information.

Track Your NNMi Licenses

To assist you in tracking your NNMi licenses, NNMi displays a status message at the bottom of the main console whenever the number of nodes in the database reaches your licensed capacity limit (compared to the number of nodes discovered). Install additional licenses (for 50 node increments or more) to extend the limit.

To see a report of the current number of discovered nodes and the current NNMi licensed capacity limit, access **View Licensing Information** from either of the following locations:

- **Help → About HP Network Node Manager i software**
- The Console Sign-In window

There are four categories of NNMi Software Licenses. Within each category, there are three types (instant-on, temporary, or permanent):

- Licenses for NNMi or NNMi Advanced:
  - Base (NNMi:Runtime)
  - iAdvanced
- Integration Enablement licenses. For example, required when connecting to HP Network Node Manager i Software Smart Plug-ins (iSPIs) on a remote server or extending the functionality of NNMi in other ways.
- Licenses for developers (SDK licenses).

When tracking license information, note the following:

- NNMi discovers and manages nodes up to the NNMi licensed capacity limit.
- If the number of discovered nodes reaches or exceeds the licensed capacity limit, NNMi randomly "Unmanages" nodes until the number of "Managed" nodes matches the licensed capacity limit. For example: this situation might occur when an Instant-On or Temporary license expires or when an incremental license is intentionally uninstalled from a particular server. No new nodes are discovered unless one of the following occurs:
  - Install a license extension, see "Extend a Licensed Capacity" (on page 1020). (Any seeds that were "**Unmanaged**[1]" because of license issues, must be manually changed back to "Managed" after the license extension is installed.)

[1]Indicates the Management Mode is "Not Managed" or "Out of Service".

- Review your configuration settings and limit NNMi discovery to only the important nodes in your network environment (see "Discovering Your Network" (on page 113)). Then, delete nodes and let NNMi rediscovery reset the managed inventory of nodes (see "Delete Nodes" (on page 172)).

- NNMi generates Incidents under the following circumstances:
  - The number of discovered nodes exceeds the current licensed capacity limit.
  - An Instant-On or Temporary license expires.
  - HP Network Node Manager i Software Smart Plug-ins (iSPIs) are purchased and installed on the NNMi management server. However, the NNMi licensed capacity limit does not match the NNM iSPI licensed capacity limit. See "Purchase an HP Network Node Manager i Smart Plug-in" (on page 953) for more information about the NNM iSPIs.
  - The NNMi licensed capacity limit does not match the required Integration Enablement licensed capacity limit. For example, when connecting to HP Network Node Manager i Software Smart Plug-ins (iSPIs) on a remote server or when extending the functionality of NNMi in other ways by using the NNMi SDK.

Related Topics:

"Extend a Licensed Capacity" (on page 1020)

"Purchase an HP Network Node Manager i Smart Plug-in" (on page 953)

"Integrations with Other HP Products" (on page 954)

Extend a Licensed Capacity

To extend the licensed capacity, purchase and install an additional NNMi or NNMi Advanced Software License.

Contact your HP Sales Representative or your Authorized Hewlett-Packard 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 HP License Key Delivery Service: https://webware.hp.com/welcome.asp

For more information, see the *HP Network Node Manager i Software Installation Guide* and nnmlicense.ovpl for more information.

Note: The licensed capacity count is cumulative for each licensed product (across all installed license keys for that licensed product).

After you purchase a software license, install the NNMi Software License key using one of the following methods:

- **From the command line**:
  a. At the command prompt for the NNMi management server, type the following (see the nnmlicense.ovpl Reference Page for more information):

     For <*product*>, use one of the following: NNM, iSPI-NET, iSPI-Points, or PerfSPI

     - **Windows**:
       `%NnmInstallDir%\bin\nnmlicense.ovpl <product> -f <license_file>`
     - **UNIX**:
       `opt/OV/bin/nnmlicense.ovpl <product> -f <license_file>`

  b. NNMi automatically completes the installation.

- **Using Autopass and your HP Order Number (not possible behind a firewall)**:

  a. Open the Autopass user interface. At the command line for the NNMi management server, type the following (see the nnmlicense.ovpl Reference Page for more information):

     For <*product*>, use one of the following: NNM, iSPI-NET, iSPI-Points, or PerfSPI

     - **Windows**:
       `%NnmInstallDir%\bin\nnmlicense.ovpl <product> -gui`
     - **UNIX**:
       `opt/OV/bin/nnmlicense.ovpl <product> -gui`

  b. On the left side of the Autopass window, click **License Management**.

  c. Click **Install License Key**.

  d. Click **Retrieve/Install License Key**.

  e. Enter your HP Order Number and follow the Autopass prompts to complete the License key retrieval process.

  f. Autopass automatically completes the installation.

Related Topics:

"Track Your NNMi Licenses" (on page 1019)

"Purchase an HP Network Node Manager i Smart Plug-in" (on page 953)

"Integrations with Other HP Products" (on page 954)

About Environment Variables

These are the default values for NNMi environment variables. Actual values depend on the selections made during NNMi installation. See the nnm.envvars Reference Page for more information.

Operating System	Environment Variable Values

Windows 2008	`%NnmInstallDir% = <drive>\Program Files(x86)\HP\HP BTO Software\`
	`%NnmDataDir% = <drive>\ProgramData\HP\HP BTO Software\`
	<*drive*> is the location where NNMi was installed.
	Note: On Windows systems, the NNMi installation process creates these environment variables so they are always available.

Windows 2003:	`%NnmInstallDir% = <drive>\Program Files(x86)\HP\HP BTO Software\`
	`%NnmDataDir% = <drive>\Documents and Settings\All Users\Application Data\HP\HP BTO Software\`
	<*drive*> is the location where NNMi was installed.
	Note: On Windows systems, the NNMi installation process creates these environment variables so they are always available.

Operating System	Environment Variable Values

UNIX	`$NnmInstallDir = /opt/OV/`
	`$NnmDataDir = /var/opt/OV/`
	Note: On UNIX systems, you must manually create these environment variables if you want to use them. For more information, see the HP Network Node Manager i Software Deployment Reference, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`, and the nnm.envvars Reference Page.

Export and Import Configuration Settings

See the nnmconfigexport.ovpl and nnmconfigimport.ovpl Reference Pages for more information, including the complete list of the command line arguments for each command.

The choices that you make when exporting NNMi configuration settings determine how that configuration information can be used. For example:

- Export a copy of the existing NNMi configuration settings before you try experimenting with a new idea. You can use that exported file to restore your configuration settings if your experiment doesn't work the way you thought it would work.
- Export the NNMi configuration settings from a server in your test environment. Import those configuration settings onto the NNMi management server that your team will use to manage your network environment.
- (*NNMi Advanced - Global Network Management feature*) Export configuration settings to share configuration settings among the Regional Managers in your network environment (for example, Node Group definitions, Trap Forward to Global Managers settings, and NNM 6.x/7.x Event Incidents Forward to Global Managers settings).

Carefully review the following topics to make an informed choice:

"Export/Import Behavior and Dependencies" (on page 1022)

"Export a Snapshot of Your Configuration Settings" (on page 1027)

"Import Configuration Files to Restore Previous Settings" (on page 1028)

"Transfer Configuration Settings to Another NNMi Management Server" (on page 1030)

"Troubleshooting Imports of Configuration Files" (on page 1032)

Export/Import Behavior and Dependencies

Your configuration settings can be exported to make a copy, and then imported onto the same NNMi management server or another NNMi management server. The locale setting on the NNMi management server at the time of Export must match the locale setting on the NNMi management server at the time of Import.

You need to understand the behavior and dependencies (see the table below). The choices that you make when exporting NNMi configuration settings determine how that configuration information can be used:

Replaces all. Export files with this behavior make changes to the NNMi database when Imported (click here for more information).

- NNMi replaces all object instances with matching key identifiers (see "Troubleshooting Imports of Configuration Files" (on page 1032) for information about key identifiers).
- NNMi adds all object instances with key identifiers that do not exist in the NNMi database
- ***NNMi deletes all existing object instances with key identifiers that do not match any in the exported file***.

Incremental. Export files with this behavior make changes to the NNMi database when Imported (click here for more information).

- NNMi updates all object instances with matching key identifiers (see "Troubleshooting Imports of Configuration Files" (on page 1032) for information about key identifiers).

  Caution: NNMi also overwrites the values of any codes associated with these object instances (for example, incident family).
- NNMi adds all object instances with key identifiers that do not exist in the NNMi database.
- NNMi does not touch existing object instances with key identifiers that do not match any in the exported file.

Incremental (subset). Export files with this behavior include configuration changes that were made by one Author. Export files with this behavior make changes to the NNMi database when Imported (click here for more information).

- NNMi updates all object instances with matching key identifiers (see "Troubleshooting Imports of Configuration Files" (on page 1032) for information about key identifiers).

  Caution: NNMi also overwrites the values of any codes associated with these object instances (for example, incident family).
- NNMi adds all object instances with key identifiers that do not exist in the NNMi database.
- NNMi does not touch existing object instances with key identifiers that do not match any in the exported file.

Export/Import Behavior and Dependencies Among Configuration Areas

Configuration Workspace's View Name	Export Option	Import Behavior	Dependencies
Author *	-c author	Incremental	No dependencies. Import requires one Export file (author.xml). * Not a workspace, but an important data object.
	-c author -a <*authorName*>	Incremental (subset)	No dependencies. Import requires one Export file (author.xml).
Communication	-c comm	Replaces all	No dependencies. Import requires one Export file (comm.xml). **Caution**: SNMPv3 configuration settings cannot be exported because SNMPv3 data is encrypted based on the NNMi encryption key (generated during NNMi installation). Therefore, the SNMPv3 encrypted data

Configuration Workspace's View Name	Export Option	Import Behavior	Dependencies
			cannot be imported into another installed version of NNMi because the encryption key is different.
Custom Correlations	-c customCorrelation	Incremental	
	-c device -a <*authorName*>	Incremental (subset)	The required Author information is embedded in the Export file.
Custom Poller	-c custpoll	Incremental	Import requires four Export files, and they must be imported in this order: (1) author.xml, (2) device.xml, (3) nodegroup.xml, and (4) custpoll.xml **Note**: When importing Custom Poller configurations, NNMi automatically sets the Active State attribute value for all imported Policies to *Suspended*.
Device Profiles	-c device	Incremental	Import requires two Export files, and they must be imported in this order: (1) author.xml and (2) device.xml
	-c device -a <*authorName*>	Incremental (subset)	Import requires one Export file (device.xml). The required Author information is embedded in the Export file.
Discovery	-c disco	Replaces all	Import requires seven Export files, and they must be imported in this order: (1) comm.xml, (2) discoseed.xml, (3) iftype.xml, (4) author.xml, (5) device.xml, (6) ifgroup.xmland, and (7) disco.xml
Discovery Seeds	-c discoseed	Incremental	Import requires two Export files, and they must be imported in this order: (1) comm.xml and (2) discoseed.xml
Global Network Management			No export/import allowed at this time.
Incident	-c incident	Replaces all	Import requires two Export files, and they must be imported in this order: (1) author.xml and (2) incident.xml

Configuration Workspace's View Name	Export Option	Import Behavior	Dependencies
-c incident -a <*authorName*>	Incremental (subset)	Import requires one Export file (incident.xml). The required Author information is embedded in the Export file.	
Interface Groups	-c ifgroup	Incremental	Import requires five Export files, and they must be imported in this order: (1) iftype.xml, (2) author.xml, (3) device.xml, (4) nodegroup.xml, and (5) ifgroup.xml
IfTypes	-c iftype	Incremental	Interface Types. No dependencies. Import requires one Export file (iftype.xml).
Management Stations (6.x/7.x)	-c station	Incremental	NNM 6.x or 7.x Management Stations. No dependencies. Import requires one Export file (station.xml).
Menus	-c menu	Incremental	Import requires two Export files, and they must be imported in this order: (1) author.xml and (2) menu.xml
	-c menu -a <*authorName*>	Incremental (subset)	Import requires one Export file (menu.xml). The required Author information is embedded in the Export file.
Menu items (formally URL Actions)	-c menuitem (formally -c urlaction)	Incremental	Import requires two Export files, and they must be imported in this order: (1) author.xml and (2) menuitem.xml

Configuration Workspace's View Name	Export Option	Import Behavior	Dependencies
-c menuitem -a <*authorName*> (formally -c urlaction -a <*authorName*>)	Incremental (subset)	Import requires one Export file (menuitem.xml). The required Author information is embedded in the Export file.	
MIB Expressions	-c mibexpr	Incremental	Import requires two Export files, and they must be imported in this order: (1) author.xml and (2) mibexpr.xml
	-c mibexpr -a <*authorName*>	Incremental (subset)	Import requires one Export file (mibexpr.xml). The required Author information is embedded in the Export file.
Monitoring	-c monitoring	Replaces all	Import requires six Export files, and they must be imported in this order: (1) author.xml, (2) device.xml, (3) nodegroup.xml, (4) iftype.xml, (5) ifgroup.xml, and (6) monitoring.xml
Node Groups	-c nodegroup	Incremental	Import requires three Export files, and they must be imported in this order: (1) author.xml, (2) device.xml, and (3) nodegroup.xml **Caution**: Island Node Groups are never exported. See "Island Node Groups" (on page 211).
Node Group Map Settings	-c ngmap	Incremental	Import requires four Export files, and they must be imported in this order: (1) author.xml, (2) device.xml, (3) nodegroup.xml, and (4) ngmap.xml **Note**: Any time you save a map layout, NNMi deletes any previous node locations. Therefore, each export contains only the node locations that were last saved.

Configuration Workspace's View Name	Export Option	Import Behavior	Dependencies
RAMS Servers	-c rams	Incremental	HP Router Analytics Management Systems data from the RAMS Servers view (does not include data from the Integration Module Configuration *HP RAMS MPLS WAN*). No dependencies. Import requires one Export file (rams.xml).
Status	-c status	Replaces all	No dependencies. Import requires one Export file (status.xml). The imported status applies to all Node Groups in the database.
User Accounts and Roles User Principals	-c account	Incremental	This command gathers data from two Configuration workspace views. Import requires one Export file (account.xml). The data from both User Accounts and Roles User Principals is embedded in the Export file.
User Interface	-c ui	Incremental	No dependencies. Import requires one Export file (ui.xml).

Export a Snapshot of Your Configuration Settings

If you export your configuration settings before you begin making changes, you can easily "undo" your changes if you decide that you do not like the results.

To export a snapshot of your configuration settings:

1. Import behavior is determined when you generate the Export files. Make sure you understand your Import behavior choices before you begin. See "Export/Import Behavior and Dependencies" (on page 1022) (consider printing that topic for reference).

   Caution: The locale setting on the NNMi management server at the time of Export must match the locale setting on the NNMi management server at the time of Import. Otherwise, you risk losing data. Also, make sure that both NNMi management servers have the same NNMi version/patch number.

2. A user name and password are required with the export command:

   If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access" (on page 275) for more information.

   `-u <NNMiadminUserName> -p <NNMiadminPassword>`

3. Check whether the configuration settings you want to export have dependencies, see "Export/Import Behavior and Dependencies" (on page 1022).

- If no dependencies, export only the configuration settings you are planning to change.
- If yes, decide if you need a copy of the dependencies (only if you plan to make changes to those configuration settings, as well). Then export all the required files.

4. At the command line of the NNMi management server, type the command to generate the required export files.
   - To export all configuration settings, use the following command:

     `nnmconfigexport.ovpl -c -all -f <directory>`

     You can use `-x <file_prefix>` to provide a unique prefix to the set of exported files. For example, use today's date or a clue about the reason you needed the files:

     `nnmconfigexport.ovpl -c -all -f <directory> -x <file_prefix>`

   - To export specific configuration settings <*X*> from multiple configuration workspace views, separate each with a comma (see "Export/Import Behavior and Dependencies" (on page 1022) or the nnmconfigexport.ovpl Reference Pages for the list of choices):

     `nnmconfigexport.ovpl -c <X>, <X>, <X> -f <directory>`

     You can use `-x <file_prefix>` to provide a unique prefix to the set of exported files. For example, use today's date or a clue about the reason you needed the files:

     `nnmconfigexport.ovpl -c <X>, <X>, <X> -f <directory> -x <file_prefix>`

   - To export configuration settings that were created by a particular author (for Author, Device Profiles, Incident, or URL Actions), add the `-a <authorName>` attribute to the command and provide the Unique Key.

     Note: Only one author per `-a <authorName>` export command is allowed.

     Find the Unique Keys for all authors by exporting an author.xml file, then open the file in a text editor and locate the Key attribute values.

     Find the Unique Key for a particular Author, in the NNMi console:

     i. Open one of these Configuration workspaces in the NNMi console: Device Profiles Configuration, Incident Configuration, or URL Actions.

     ii. Select an object created by the Author of interest.

     iii. Display the Author form, and copy the value of the Unique Key attribute.

5. Verify that the required xml files are in the specified directory.

   Caution: Do not edit the exported file before importing.

You are now ready to make configuration changes.

If you need to undo your configuration setting changes, see "Import Configuration Files to Restore Previous Settings" (on page 1028).

Import Configuration Files to Restore Previous Settings

If you have a set of export files, you can change the Configuration settings on your NNMi management server to match the settings in the exported files.

Note: You can change the names of the exported files before importing. The import still works the same.

Caution: Do not edit the exported file before importing.

To import a previous snapshot of your configuration settings:

1. Import behavior is determined when you generate the Export files. Make sure your exported files were generated in a manner that meets your current needs. See "Export/Import Behavior and Dependencies" (on page 1022) (consider printing that topic for reference).

   Caution: The locale setting on the NNMi management server at the time of Export must match the locale setting on the NNMi management server at the time of Import. Otherwise, you risk losing data. Also, make sure that both NNMi management servers have the same NNMi version/patch number.

2. A user name and password are required with the import command:

   If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access" (on page 275) for more information.

   `-u <NNMiadminUserName> -p <NNMiadminPassword>`

3. Check whether the configuration settings you want to import have dependencies, see "Export/Import Behavior and Dependencies" (on page 1022).

   - If no dependencies, import only the configuration settings you are planning to change.
   - If yes, decide if you need a copy of the dependencies (only if you made changes to those configuration settings, as well). Then import all the required files.

4. At the command line of the NNMi management server, type the command to import a file:

   `nnmconfigimport.ovpl -f <filename>`

   When importing multiple XML files at once using `-f <directory>`, the NNMi `nnmconfigimport.ovpl` command takes care of ordering issues.

   - To import all configuration settings, use the following command:

     `nnmconfigimport.ovpl -f <directory>`

     You can use `-x <file_prefix>` to specify a unique prefix for a set of files:

     `nnmconfigimport.ovpl -f <directory> -x <file_prefix>`

   - To import specific configuration settings from multiple configuration areas, create a directory that contains the set of files you want to import.

     At the command line of the NNMi management server, type the appropriate command to import files:

     `nnmconfigimport.ovpl -f <directory>`

     You can use `-x <file_prefix>` to specify a unique prefix for a set of files:

     `nnmconfigimport.ovpl -f <directory> -x <filePrefix>`

   - To import configuration settings that were created by specific authors (for Author, Device Profiles, Incident, or URL Actions), create a directory that contains the set of files you want to import.

     At the command line of the NNMi management server, type the appropriate command to import the files:

     `nnmconfigimport.ovpl -f <file>`

```
nnmconfigimport.ovpl -f <directory>
```

You can use `-x <file_prefix>` to provide a unique prefix for a set of exported files:

```
nnmconfigimport.ovpl -f <directory> -x <filePrefix>
```

5. If you encounter problems, see "Troubleshooting Imports of Configuration Files" (on page 1032).

   Additional import options for timeout or memory issues:

   You can append the following options to any import command if you encounter problems:

Option	Description	Default Setting
`-timeout <seconds>`	For larger data imports, you may encounter timeout issues. To increase the number of seconds that NNMi waits (per file) during an import, append the `-timeout` option to the end of your command line.	1800 seconds (minimum)
`-memory <megabytes>`	For larger data imports, you may encounter memory issues. To increase the number of megabytes allotted to memory during an import, append the `-memory` option to the end of your command line.	512 megabytes

6. After completing the import, open NNMi and verify your configuration settings.

Transfer Configuration Settings to Another NNMi Management Server

You can export configuration settings and import them onto another NNMi management server to save time.

Note: You can change the names of the exported files before importing. The import still works the same.

Caution: Do not edit the exported file before importing.

To move configuration settings to another NNMi management server:

1. Import behavior is determined when you generate the Export files. Make sure you understand your Import behavior choices before you begin. See "Export/Import Behavior and Dependencies" (on page 1022) (consider printing that topic for reference).

   Caution: The locale setting on the NNMi management server at the time of Export must match the locale setting on the NNMi management server at the time of Import. Otherwise, you risk losing data. Also, make sure that both NNMi management servers have the same NNMi version/patch number.

2. A user name and password are required with the export command:

   If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access" (on page 275) for more information.

   ```
   -u <NNMiadminUserName> -p <NNMiadminPassword>
   ```

3. Check whether the configuration settings you want to export have dependencies, see "Export/Import

Behavior and Dependencies" (on page 1022).

- If no dependencies, export only the configuration settings you are planning to change.
- If yes, decide if you need a copy of the dependencies (only if you plan to make changes to those configuration settings, as well). Then export all the required files.

4. At the command line of the NNMi management server export all configuration settings, type the appropriate command:

   `nnmconfigexport.ovpl -c -all -f <directory>`

   You can use `-x <file_prefix>` to provide a unique prefix to the set of exported files. For example, use today's date or a clue about the reason you needed the files:

   `nnmconfigexport.ovpl -c -all -f <directory> -x <file_prefix>`

   Note: You can change the names of the exported files before importing. The import still works the same.

5. Delete any files that you do not need.
6. To export configuration settings that were created by a particular author (for Author, Device Profiles, Incident, or URL Actions), repeat the export command for each configuration item modified by that author. Add the `-a <authorName>` attribute to the command and provide the Unique Key.

   Note: Only one author per `-a <authorName>` export command is allowed.

   `nnmconfigimport.ovpl -a <authorName> -f <directory>`

   You can use `-x <file_prefix>` to provide a unique prefix to the set of exported files. For example, use today's date or a clue about the reason you needed the files:

   `nnmconfigimport.ovpl -a <authorName> -f <directory> -x <file_prefix>`

   Find the Unique Keys for all authors by exporting an author.xml file, then open the file in a text editor and locate the Key attribute values.

   Find the Unique Key for a particular Author, in the NNMi console:

   a. Open one of these Configuration workspaces in the NNMi console: Device Profiles Configuration, Incident Configuration, or URL Actions.
   b. Select an object created by the Author of interest.
   c. Display the Author form, and copy the value of the Unique Key attribute.

7. Verify that all required xml files are in the specified directory.

To import the configuration settings onto the other NNMi management server:

1. Import behavior is determined when you generate the Export files. Make sure you understand your Import behavior choices before you begin. See "Export/Import Behavior and Dependencies" (on page 1022) (consider printing that topic for reference).

   Caution: The locale setting on the NNMi management server at the time of Export must match the locale setting on the NNMi management server at the time of Import. Otherwise, you risk losing data. Also, make sure that both NNMi management servers have the same NNMi version/patch number.

2. A user name and password are required with the export command:

If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access" (on page 275) for more information.

`-u <NNMiadminUserName> -p <NNMiadminPassword>`

3. Verify that all required xml files are in the specified directory.
4. When importing multiple XML files at once using `-f <directory>`, the NNMi `nnmconfigimport.ovpl` command takes care of ordering issues.

   At the command line of the NNMi management server, type the appropriate command to import the configuration files that you gathered for transfer:

   `nnmconfigimport.ovpl -f <filename>`

   `nnmconfigimport.ovpl -f <directory>`

   You can use `-x <file_prefix>` to specify a unique prefix for a set of files:

   `nnmconfigimport.ovpl -f <directory> -x <file_prefix>`
5. If you encounter problems, see "Troubleshooting Imports of Configuration Files" (on page 1032).

   Additional import options for timeout or memory issues:

   You can append the following options to any import command if you encounter problems:

Option	Description	Default Setting
`-timeout <seconds>`	For larger data imports, you may encounter timeout issues. To increase the number of seconds that NNMi waits (per file) during an import, append the `-timeout` option to the end of your command line.	1800 seconds (minimum)
`-memory <megabytes>`	For larger data imports, you may encounter memory issues. To increase the number of megabytes allotted to memory during an import, append the `-memory` option to the end of your command line.	512 megabytes

6. After completing the import, open NNMi and verify your configuration settings.

Troubleshooting Imports of Configuration Files

When importing incremental sets of configuration files, NNMi abandons the import if mis-matched configuration objects are encountered. Each configuration object has a set of Unique Identifier values that must match for incremental updates, or must not match any existing data before NNMi adds the configuration object to the database, see tables below.

If you receive an error message while trying to import configuration information, use the information below to figure out how to use the error message to determine what you need to change before creating another export file (thus, solving the problem).

Note: You can change the names of the exported files before importing. The import still works the same.

Caution: Do not edit the exported file before importing.

Author Configuration Unique Identifiers

Configuration Item Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
Author = Author form	author = Unique Key attribute value	Yes	

Communication Configuration Unique Identifiers

Attribute Name = NNMi Console Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
CommunicationRegion = Communication Region form	uuid	No	name = Name value ordering = Ordering value **Caution**: SNMPv3 configuration settings cannot be exported because SNMPv3 data is encrypted based on the NNMi encryption key (generated during NNMi installation). Therefore, the SNMPv3 encrypted data cannot be imported into another installed version of NNMi because the encryption key is different.

Custom Poller Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
ComparisonMap = Comparison Map form	uuid	No	ordering = Ordering value
Policy = Custom Poller Policy form	uuid	No	The combination of these two: collection = Collection value ordering = Ordering value

Device Profile Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
DeviceCategory = Device Category form	key = Unique Key attribute	Yes	
DeviceFamily = Device Family form	key = Unique Key attribute	Yes	
DeviceProfile = Device Profile form	snmpObjectId = SNMP Object ID value	Yes	
DeviceVendor = Device Vendor form	key = Unique Key attribute	Yes	

Discovery Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
AutoDiscoveryRegion = Auto-Discovery Rule form	uuid	No	name = Name value ordering = Ordering value

Discovery Seed Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
DiscoverySeed = Discovery Seed form	host = Hostname (*not case-sensitive*) / IP Address value	Yes	

Incident Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
MgmtEventConfig = Management Event Configuration form	uuid	No	name = Name value
SnmpTrapConfig = SNMP Trap Configuration (by OID) form	uuid	No	iod = SNMP Object ID value name = Name value
RemoteNnmEventConfig	uuid	No	iod name = Name value
PairwiseConfig = Pairwise Configuration form	uuid	No	name = Name value The combination of these two: firstIncidentName = First Incident Configuration value secondIncidentName = Second Incident Configuration value

Interface Groups Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
InterfaceGroup = Interface Group form	uuid	No	name = Name value

Interface Type Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
IfType	IfType attribute	Yes	

Menus Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
Menu = UI Configuration > Menus form	key = Unique Key attribute	Yes	

Menu Items Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
MenuItem = Menu Item form	key = Unique Key	Yes	

Monitoring Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
InterfaceSettings = Interface Settings form	uuid	No	ordering = Ordering value
NodeSettings = Node Settings form	uuid	No	ordering = Ordering value

Node Group Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
NodeGroup = Node Group form	uuid	No	name = Name value

User Accounts and Roles Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
Account = User Account form	uuid	No	name = Name value
Principal = User Principal form	uuid	No	Name value
RoleEntry = Account Mapping form	uuid	No	The combination of these two:

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
			principal = Account value role = Role value

Back Up and Restore NNMi

As an NNMi administrator, develop a plan for NNMi backups.

For the most complete information, see the "NNMi Back Up and Restore" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`. See also nnmbackup.ovpl and nnmrestore.ovpl (**Help** → **Documentation Library** → **Reference Pages**, in the Administrator Commands category).

Use the `nnmbackup.ovpl` and `nnmrestore.ovpl` command line tools to do any of the following:

- Back up the NNMi management server and restore data to the same machine.
- Back up the NNMi management server and use the nnmrestore.ovpl command to place the backed up configuration records and database records onto another NNMi management server. For example, moving NNMi to another NNMi management server due to a hardware failure on the original server.

  Note: Both machines must have the same type of operating system and NNMi version and patch level. To move NNMi configuration settings from one computer to another computer that is running a different type of operating system, see "Export and Import Configuration Settings".

  After you restore NNMi on the second NNMi management server, uninstall NNMi from the original NNMi management server. For more information, see the HP Network Node Manager i Software Deployment Reference, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.
- Back up the NNMi management server as a safeguard before upgrading the operating system on the server.
- Back up the NNMi management server as a safeguard before updating to a newer version of NNMi.

Note: The back up and restore data may or may not include data from any HP Network Node Manager i Software Smart Plug-ins (iSPIs) installed in your network environment. Check the documentation that came with each NNM iSPI for details.

Before you begin a backup, ensure you have adequate storage space for the backup copy. Verify that you have enough space to store the contents of the directories listed in the following table.

Note: You may compress the files after backup.

See also "About Environment Variables" (on page 1021).

NNMi Directories

Operating System	Data	Default Location
Windows 2008	Configuration Files	`<drive>:\Program Files (x86)\HP\HP BTO Software` `<drive>` is the drive on which NNMi is installed
	Configuration Data	`<drive>:\ProgramData\HP\HP BTO Software`
	Embedded NNMi Database Storage	`<drive>:\ProgramData\HP\HP BTO Software\shared\nnm\databases\Postgres` If you chose the Oracle database instead of the embedded NNMi database at install time, you must use that database's tools for backup in addition to `nnmbackup.ovpl`.
Windows 2003	Configuration Files	`<drive>:\Program Files (x86)\HP\HP BTO Software` `<drive>` is the drive on which NNMi is installed
	Configuration Data	`<drive>:\Documents and Settings\All Users\Application Data\HP\HP BTO Software`
	Embedded NNMi Database Storage	`<drive>:\Documents and Settings\All Users\Application Data\HP\HP BTO Software\shared\nnm\databases\Postgres` If you chose the Oracle database instead of the embedded NNMi database at install time, you must use the Oracle tools for backup in addition to `nnmbackup.ovpl`.
HP-UX, Linux, Solaris	Configuration Files	`/opt/OV`
	Configuration Data	`/var/opt/OV`
	Embedded NNMi Database Storage	`/var/opt/OV/shared/nnm/databases/Postgres` If you chose the Oracle database instead of the embedded NNMi database at install time, you must use the Oracle tools for backup in addition to `nnmbackup.ovpl`.

Related Topics

"Export and Import Configuration Settings" (on page 1022)

"Archive and Delete Incidents" (on page 1037)

Archive and Delete Incidents

NNMi enables you to archive and remove incidents that you no longer want to track. For example, this feature is useful if you want to purge the database of incidents that are older than a specified time period or date. Use the `nnmtrimincidents.ovpl` command to create a comma-separated-values (CSV) file containing the history of incidents, and then trim the volume of incidents to manage the size of your database.

To archive and then delete incidents in NNMi, use the `nnmtrimincidents.ovpl` command. You can choose to only archive or only delete your incidents as described in the arguments table that follows.

When archiving and deleting incidents, for the best performance results, archive and delete your incidents frequently to keep the size of the NNMi database as small as possible.

SNMP traps are a subset of NNMi incidents (see `-origin` in the arguments table that follows). NNMi monitors the volume of SNMP traps that are stored in the NNMi database. The maximum allowed number of SNMP traps is 100,000. Note the following:

- After 90 percent of the maximum limit for SNMP traps is reached or exceeded, NNMi generates an incident with Severity set to Warning to notify you that NNMi is approaching the maximum limit.
- After 95 percent of the maximum limit for SNMP traps is reached or exceeded, NNMi generates an incident with Severity set to Major to notify you that NNMi is approaching the maximum limit. In addition, NNMi only accepts traps required for Causal Engine analysis until the number of SNMP traps within the database has been reduced using the `nnmtrimincidents.ovpl` command.
- After the maximum SNMP trap limit is reached or exceeded, NNMi generates an incident with Severity set to Critical. NNMi no longer accepts any SNMP traps until the number of SNMP traps within the database has been reduced using the `nnmtrimincidents.ovpl` command.

Use the `nnmtrimincidents.ovpl` command to archive and delete your incidents based on any of the attributes described in the following table. See the nnmtrimincidents.ovpl command for more information, including a complete list of arguments for this command.

Note: The archive's comma-separated-values (CSV) file cannot be used to import the incidents back into NNMi.

Archive Incident Arguments

Incident Attribute	Description
-archiveOnly	Used to specify that you want to only archive incidents rather than archive and then delete them.
-trimOnly	Used to specify that you want to only delete incidents rather than archive and then delete them.
-date	The date must be entered in the following ISO 8601 format: <*yyyy-mm-dd*>T<*hh*>:<*mm*>:<*ss*>[Z,-<*hh*>:<*mm*>,+<*hh*>:<*mm*>] ISO Date Format: • *yyyy* — Four-digit year • *mm* — Two-digit month • *dd* — Two-digit day • *hh* — Two digits representing the hour (00 through 23) • *mm* — Two digits representing the minutes (00 through 59) • *ss* — Two digits representing the seconds (00 through 59) • +<*hh*>:<*mm*> — Local time zone which is the hours (<*hh*>) and minutes (<*mm*>) ahead of Coordinated Universal Time • -<*hh*>:<*mm*> — Local time zone which is the hours (<*hh*>) and minutes (<*mm*>) behind Coordinated Universal Time

Incident Attribute	Description
	For example: `2007-11-05T08:15:30-5:00` corresponds to November 5, 2007, 8:15:30 am, Eastern Standard Time. **Note**: You must specify either a -age or a -date value.
-age	The age of the incident specified in number of days, weeks, or months. **Note**: You must specify either an -age or a -date value.
-family	The incident Family. See Incident Form: General Tab for a list of possible Family values.
-incr	The increment value that helps determine the -age value. Supported increments include **days**, **weeks**, and **months**. The default increment value is **days**.

-path	Specifies the archive file name, including the complete path. The default archive file name is:
	
	<*date*> is the date in *yyyy-mm-dd* format
	
	<*ms*> is milliseconds
	
	Windows:
	
	`%NnmDataDir%\tmp\incidentArchive.<date>.<ms>.txt.gz`
	
	UNIX:
	
	`/var/opt/OV/tmp/incidentArchive.<date>.<ms>.txt.gz`
	
	Note: Each time you generate an archive, NNMi overwrites any existing file with the same name. Therefore, to ensure that all archive files are preserved, provide a unique archive file name each time you want to archive incidents.

-lifecycle	Identifies where the incident is in the incident lifecycle. Possible values are **Registered**, **In Progress**, **Completed**, and **Closed**. See About the Incident Lifecycle for more information about **Lifecycle State**. **Note**: This argument is optional.
-name	Identifies the name of the incident configuration.

Incident Attribute	Description
-nature	Identifies the nature of the incident. Possible values are: **Info, None, Root Cause, Secondary Root Cause, Service Impact,** and **Symptom**. See Using the Incident Form for more information. **Note**: This argument is optional.
-origin	Identifies the Origin of the incident configuration. Possible values are: **Management Software**, **Manually Created**, **Remotely Generated**, and **SNMP Trap**. See Incident Form: General Tab for more information.
-u	The user name required to run this command. This user name must be a valid NNMi user name with a role of either Administrator or System. **Note**: The user name might be a Principal object stored in the NNMi database or might be from your environment's directory service database (depending on NNMi configuration). See "Configure Sign-In Access" (on page 262) for more information.
-p	The associated password for the user name specified by the -u attribute value. If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access" (on page 275) for more information.
-quiet	Use this argument when you want to trim incidents without requiring user prompts and responses. (Status information appears.)
-sysobjectid	The system object identification (system OID) assigned to the incident configuration. For SNMP Trap incidents, this value is obtained from the incoming SNMP trap. For Remote NNM 6.x/7.x Event incidents, this value is obtained from the Remote NNM 6.x/7.x event. For Management Event incidents generated by NNMi, the system OID is assigned by NNMi.

For example, archive and delete all incidents with lifecycle equal to Closed and age equal to or greater than 1 month.

```
nnmtrimincidents.ovpl -age 1 -incr months -lifecycle Closed -u <NNMiadminUsername>
-p <NNMiadminPassword>
```

You can also specify a batch size when archiving or deleting incidents. Specify the maximum number of incidents to delete at one time within a single database transaction. This number then determines how often you see a status message that the deletions are complete. Using the default value of 1,000 as an example, NNMi displays a status message after successfully deleting each 1,000 incidents.

Note: The default value of 1,000 was selected to maintain a balance between performance and the frequency of progress messages for the archive and delete operation. This default determines the maximum number of incidents archived and deleted at one time within a single database transaction.

Related Topics

"Back Up and Restore NNMi" (on page 1036)

Appendix A: Glossary Terms

A

AES
Advanced Encryption Standard

Anycast Rendezvous Point IP Address
Rendezvous Point addresses are loopback addresses used for routers in multi-cast network configurations.

B

BGP
Border Gateway Protocol

C

Causal Engine
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 problems and determines the root cause for you, whenever possible, sending incidents to notify you of problems. Any incident generated from a Causal Engine management event has an Origin of NNMi in your incident views.

CBC
Cipher Block Chaining

D

DES
Data Encryption Standard

E

EIGRP
Enhanced Interior Gateway Routing Protocol

G

global unicast address
(2000:: to 3fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff) A publically routable IPv6 unicast address, used for communication between nodes anywhere on the internet. The first part of the address is a global routing prefix in the 2000::/3 address space for your organization (assigned by the Internet Service Providers). The complete host address may either be manually configured or automatically assigned using IPv6 auto-configuration and neighbor discovery.

H

HMAC
Hash-based Message Authentication Code

HSRP
Hot Standby Router Protocol

I

ISIS

Intermediate System to Intermediate System Protocol

K

Key Incident

Incidents with both: (1) Severity = other than Normal. (2) Correlation Nature = equal to Root Cause, Service Impact, Stream Correlation, Rate Stream Correlation, Info, or None.

L

Layer 2

Refers to the Data Link layer of the multilayered communication model, Open Systems Interconnection (OSI). The Data Link layer moves data across the physical links in the network. The switches and bridges are devices that redirect data messages at the layer 2 level, using the destination Media Access Control (MAC) address to determine where to direct the message.

Layer 3

Refers to the Network layer of the multilayered communication model, Open Systems Interconnection (OSI). The Network layer is concerned with knowing the address of the neighboring nodes in the network, selecting routes and quality of service, and recognizing and forwarding incoming messages to local host domains. The router and switch-router are the devices that redirect data messages at the Layer 3 level. Everything in a subnet is connected at the Layer 3 (IP) level.

Link Aggregation

A Link Aggregation is comprised of an Aggregator Link, Aggregator Interface, and the physical interfaces and connections that they represent. An Aggregator Link object represents many-to-many physical connections. For example, two nodes might be connected with four physical connections. These four physical connections are depicted as a single Aggregator Link object using a thick line on the Layer 2 Neighbor View map. The interface depicted at each end of the Aggregator Link object is an Aggregator Interface object. An Aggregator Interface object represents the collection of physical interfaces for one end of an Aggregator Link.

link-local address

A non-routable IPv6 unicast address only used for communication with other nodes on

the same link (LAN or VLAN). Link local addresses cannot be used for communication that must be forwarded through a router. IPv6 auto-configuration automatically assigns a unique link local address in the fe80::/10 address space to each IPv6-enabled interface on a system.

loopback address

The address associated with the loopback interface. The loopback interface is a virtual interface on a device that provides a route for internal communication. Many vendors provide a specially configured loopback for management purposes. Exact details of how loopbacks are configured varies by vendor and model. See each device's documentation for details. NNMi identifies these loopback addresses by using IfType 24, softwareloopback from the IANA ifType-MIB.

M

MD5

Message-Digest algorithm 5

MPLS

Multiprotocol Label Switching

multicast address

Used to identify a group of hosts joined into a group, IPv4 multicast addresses are in the range 224.0.0.0 to 239.255.255.255 and IPv6 multicast addresses have the prefix ff00::/8

multiconnection

A set of multiple connections between Nodes, Node Groups, or Nodes and Node Groups that is indicated on a map view using a thick line. The number of connections that must exist before NNMi replaces the multiple connections (and any associated interfaces) with a single thick line on the map is set using the Multiconnection Threshold attribute when configuring the User Interface.

O

OSPF

Open Shortest Path First Protocol

P

private IP addresses

These are IPv4 addresses that can be reused in home and office local area networks (LANs). Following the standards set by RFC 1918 and RFC 4193 (10.*.*.*, 169.254.*.*, 172.16-31.*.*, and 192.168.*.*)

R

RAMS

HP Router Analytics Management System

S

SHA

Secure Hash Algorithm

U

unique local address

(fd00:: to fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff) A privately routable IPv6 unicast address used only for communication between nodes within your organization. The unique local addresses cannot be routed to the public internet. Comprised of a routing prefix in the fd00:/8 address spaces, assigned locally by your organization. And the full host address may be manually configured or automatically assigned using IPv6 auto-configuration and neighbor discovery.

Unmanaged

Indicates the Management Mode is "Not Managed" or "Out of Service".

V

VRRP

Virtual Router Redundancy Protocol

Appendix B: Index

A

Abstract Syntax Notation 452, 597, 741
access
  command line 275
  configuring sign-in 262-263, 266, 268, 271
  controlling 262, 264
  denying 272
  disabling user 272
  Tools menu 264
  troubleshooting 273, 275
  URL 264
accessing
  console 263
  device details 135
  directory service 262
  help 51
  NNM iSPI Performance Headline report 231
  Trap Analytics reports 433
Account Mapping form 266-267, 270
accounts, user
  auditing activity 274
  changing 262-263, 266, 268, 271
  determining roles 263
  disabling 272
  editing 267, 270-271
Action Configuration tab 415, 572, 717, 851
action server properties 571, 716, 850
actions
  accessing 263
  arguments 419
  configuring
    incident 414-415, 482, 521, 571-572, 628-629, 667, 716-717, 772, 850-851
    Java 897
    JavaScript 896
    node groups 807
    payload filters 574, 668, 718, 773, 808
  executing launch items 1013
  incident
    configuration 864
    parameters 415, 446, 578, 592, 723, 736, 859
  Java 897
  JavaScript 896
  launching commands
    Communication Configuration 1002
    Configuration Poll 1003
    Monitoring Settings 1006
    ping 1009
    Sign In/Out Audit Log 1016
    Sign Out 1016
    Status Details 1010
    Status Poll 1011
    traceroute 1012-1013
  line graphs 872, 890, 893
  out-of-box 30
Actions menu
  commands
    Communication Configuration 1002
    Configuration Poll 1003
    Monitor Settings 1006
    Ping 1009
    Status Details 1010
    Status Poll 1011
    Trace Route 1012-1013

configuring 873-874, 877
controlling 264, 872
Actions stage 309
Actions tab 716
actvity, auditing account 274
adding
Boolean operators 193
connections 176
custom attributes 213
discovery
seeds 159
discovery seeds 162, 164
IfTypes 208
interface
group filters 207
interfaces for new hardware 174
node
group filters 183, 190-191
optional filters 898
SNMP trap configuration 437
Additional Filters Editor
adding Boolean operators 193
buttons 183, 190-191, 207
components 183, 190-191, 198, 206-207
guidelines 191, 207
Additional Filters tab 183, 190-191, 198, 206-207
address gathering phase
node name decision tree 118
Spiral Discovery 114
addresses
assigning management mode 254, 256
configuring
IP 86
monitoring 246
direct management mode 257
discovery ranges 146
excluding from discovery 121, 133-134, 155
IPv4 50, 122, 126
IPv6 50, 126
monitoring configuration 246
out-of-box actions 30
parameters 883
restarting management 250
stopping management 250
unmanaged 253
adjusting
discovery intervals 138
administrator
backing up NNMi 1036
changing management mode 256
configuring
menu items 296
menus 296
Custom Poller 929
introduction 20
MIB
files 915
supported 911
variables 913
overriding default node group status 243
restoring NNMi 1036
role permissions 254, 262-264, 266, 268, 271, 273, 275
tools 22
Advanced, NNMi
changing management mode 256
global monitoring 217
Global Network Management 30, 52, 64, 181, 554, 696, 1022

IPv6 addresses 126
Management Stations view 583
monitoring router redundancy groups 246
Path View maps 291
port aggregation 121
RAMS 868
agents, SNMP
configuring 428
discovering 129
monitoring 215
allowedOids.conf file 431
ampersand (&)
character 419
AND variable 193
applications
fail-over health issues 64
launching NNMi from 954-955
archiving incidents 1037
arguments
action 419
nnmincidentcfg.ovpl command 429
nnmtrimincidents.ovpl command 1037
ARP cache 136
ASN 452, 597, 741
assigning
hostname wildcards 89
management mode 254, 256
node group status 243
assignments, importing community string 106
attribute values
Actions menu 873
causal rules 385, 387
communication
MIB Expressions form 936
protocol settings for node 96
regions 80
Communication Configuration form 1000, 1002
community strings
defaults 75
regions 90, 103
Configuration Poll command 1003
configuration suppression 618, 675
Configuration Workspace view 988
connection 176
correlation
filters 375
rules 358
custom correlation registration 356
Custom Incident Attributes 504, 650, 755, 790
Custom Poller 932, 935, 945, 947, 950
dampening 687, 762, 797, 829
deduplication 562, 707
default
line graph settings 295
monitoring 219
Default Device Credentials form 78
Default Map Settings 279
Description 598
device
credentials 94, 105
profile 135
Diagnostic Selection form 422
Discovery Configuration form 1000
Discovery Seeds form 159
Discovery State 166
enrichment 542, 684, 826
excluded
addresses 155

interfaces 157
Family 443, 445, 590, 733-734
Filter Editor 367, 397, 404
Full URLs 890
Global Manager 56, 554, 696
Hostname Filter 89
Incident
form 997
view 961, 963
Incident Configuration form 1000
incidents
archive arguments 1037
category codes 443-444, 589, 733
family 443, 445, 590, 733-734
severity 446, 591, 735
interface
dampening settings 472
enrichment settings 462, 465, 608, 752
suppression settings 454, 600, 744
Interface form
launching 993
monitoring settings 1006
Interface Group form 198, 206-207
interface groups 599, 743
Interface Settings form 223, 453
Inventory Workspace view 982
IP Address form
launching 994
monitoring settings 1006
IP address ranges 86
Java Action form 897
JavaScript Actions form 896
Launch Actions form 878
launched
forms 990, 993-995, 997-998, 1000
views 957, 961, 963, 965, 972, 975, 982, 985, 988
Layer 2 connection 176
Layer 2 Neighbors view 975
Lifecycle Transition Action form 415, 572, 717, 851
Management Mode Workspace view 985
Menu Item Contexts form 877
Menu Items form 874
menu nesting 873
MIB 935
MIB Expression Editor 925
MIB Expression form 920
MIB form 913
MIB Specification form 893
Monitor Settings command 1006
Monitoring Configuration form 1000
Monitoring Workspace view 972
Network Overview view 965
Networking Infrastructure Devices view 965
Node form view 990
Node Group form 183, 190-191, 998
Node Group Map Settings form 281-282, 286-287
Node Group Map view 975
Node Group Overview view 965
node groups
configuring SNMP trap settings 488
dampening settings 797
enrichment settings 501, 787
management events 778
remote NNM 6.x/7.x events 634
suppression settings 489, 779
node names 116, 118, 140
Node Settings form 234

nodes
dampening settings 657
diagnostic settings 673
enrichment settings 647
suppression settings 636
organizing incidents 443, 733
Pair Item Configuration form 353
Pairwise Configuration form 351
pairwise configurations 353
Path View 890, 975
Payload Filter Editor 792
ping command 1009
RAMS Configuration form 868
Rate Configuration 347, 712
Rediscovery Interval 138
Regional Manager 55
Regional Manager Connection form 56
Routers view 965
SNMP
default settings 67
trap definitions 429, 586, 730
SNMP Line Graph form 890
SNMPv3 297, 438
State Mapping 945
Status Configuration form 1000
Status Details command 1010
Status Poll command 1011
Subnet form 995
suppression 529, 815
Switches view 965
system object ID ranges 150
Thresholds Settings form 228, 231, 240
traceroute command 1012-1013
Trap Forwarding Destination form 302
Trap Forwarding Filter Expressions 300
Trap Forwarding Filters form 300, 304
Troubleshooting Workspace view 975
Unresponsive Interval (Days) 138
User Interface Configuration form 259, 1000

attributes
capability 886
custom 214, 887
envattrs 955
getEnvAttr 955
Attributes, Custom Incident
configuring enrichment 504, 611
Deduplication Comparison Parameters form 346, 566, 711, 845
Full URLs 888
message formats 446, 452, 592, 597, 736, 741
out-of-box 312
parameters 415, 446, 592, 736, 859
Rate Comparison Parameters form 347, 569, 714, 848
special characters 419
audit log files 274
auditing account activity 274
author configuration 1032
Auto-Discovery Rule tab 146
auto-discovery rules 120
configuring 141
basic settings 143
Ping Sweep 139
custom 128
discovery seeds 119
everything 130
interfaces 134
IP addresses 133-134, 146

- object IDs 134
- overview 127
- routers 128
- SNMP devices 129
- switches 128
- system object IDs 150
- vendors 132

Auto-Discovery Rules form 141
autocomplete 28
automated commands 422
automatic actions, configuring
- incidents 414
- interface groups 482, 772
- management events 850
- node groups 521, 667, 807
- remote NNM 6.x/7.x events 716
- SNMP trap incidents 571

automation script, discovery 179
Autopass interface 1020

B

background
- images
  - configuring 287
  - out-of-box 290
  - scaling 291
  - specifying 291

backing up NNMi 1036
Base licenses 1019
basic settings
- auto-discovery 143
- management event incidents 730
- SNMP trap configuration 438

behavior
- Actions menu 872
- auto-discovery 127
- configuring monitoring 216
- exporting configuration settings 1022
- importing configuration settings 1022
- threshold monitoring 240
- timeout/retry 72-73

BGP 120
BGP4+ 50
Blackboard stage 308
Boolean operators, adding 193
Border Gateway Protocol 50, 120
browser
- launching
  - All Incidents view 963
  - Configuration form 1000
  - Configuration Workspace view 988
  - Incident form 997
  - Incident view 961
  - Interface form 993
  - Inventory Workspace view 982
  - IP Address form 994
  - line graphs 1004
  - Management Mode Workspace view 985
  - Monitoring Workspace view 972
  - Node form 990
  - Node Group form 998
  - Subnet form 995
  - Topology Maps Workspace view 965
  - Troubleshooting Workspace view 975
  - views 957
- MIB 911, 1014
- sessions 277

buttons
- Additional Filters Editor 183, 190-191, 198, 206-207

child filters 367
Custom Filter Editor 898
Filter Editor 360, 375, 389, 397, 404
Payload Filter Editor 467, 472, 483, 489, 501, 506, 522, 529, 554, 574, 629, 636, 652, 657, 668, 687, 696, 744, 757, 762, 773, 792, 797, 808, 815, 829, 852
buying HP iSPIs 953-954
bypassing logins 957

C

capabilities
extending NNMi 872
card health
causal rules 410
cards
direct management mode 254, 257
disabled 426
restarting management 250
stopping management 250
unmanaged 253
categories
incidents 443-444, 589, 733
Causal Engine 113, 215-216, 307-308, 342, 428, 728
causal rules example 410
causal rules, configuring
child
incident filters 389
incidents 387
rules 385
source
node filters 404
object filters 397
causes, SNMP traps 443, 588
CDP 120
changes, detecting interface 174
changing
configuration settings 1022
event configurations 328
names 267, 270-271
passwords 267, 270-271, 273, 275
roles 267, 270-271
characters
action argument 419
incident messages 446, 736
reserved 882, 954
URLs 882, 954
checking
address connectivity 246
discovery
inventory 169
Layer 2 results 170
Layer 3 results 170
node state 166
progress 166
seed status 167
seed success 167
Global Manager health 63
NNMi
health 1018
Regional Manager
connection state 62
health 63
security 264
child incident filters 367, 382, 387, 389, 410
CIAs
configuring
enrichment 465, 504, 611, 650
Parent Incident filter 360

Deduplication Comparison Parameters form 34(56(71 84

Full URLs 888

message formats 446, 452, 592, 597, 736, 741

out-of-box 312

parameters 415, 446, 592, 736, 859

Rate Comparison Parameters form 347, 569, 714, 848

special characters 419

Cisco

correlation rules 382

protocols 120

routers 422

switches 422

CiscoLinkDown incident 350, 711, 714, 845, 848, 888

CiscoLinkUp incident 347

clock synchronization 61

closing incidents 310

collecting incidents 307

collections, Custom Poller 930

colors

status 216

comma (,) character 419

comma separated values 196

command line

adding discovery seeds 164

deleting discovery seeds 173

extending licenses 1020

granting user access to commands 275

loading

SNMP trap incident configurations 429

loading MIBs 919

setting up access 275

starting services 49

stopping services 49

verifying

services running 47

verifying SNMP/ICMP configuration for IP addresses 106

commands

automated 422

Communication Configuration 1002

Configuration Poll 1003

get 66

isRunning 1017

Monitoring Settings 1006

nnmbackup.ovpl 1036

nnmcommconf.ovpl 106

nnmcommload.ovpl 106

nnmconfigexport.ovpl 1022, 1027, 1030

nnmconfigimport.ovpl 1022, 1027-1028, 1030

nnmconnedit.ovpl 176

nnmhaclusterinfo.ovpl 47

NNMi Status 1015

nnmincidentcfg.ovpl 299-300, 302, 304, 429

nnmlicense.ovpl 1020

nnmloadattributes.ovpl 214

nnmloadmib.ovpl 912, 919

nnmloadnodegroups.ovpl 182, 196

nnmloadseeds.ovpl 162, 164

nnmnodedelete.ovpl 172

nnmrestore.ovpl 1036

nnmsetcmduserpw.ovpl 59, 275, 1027-1028, 1030

nnmtrimincidents.ovpl 1037

ovstart 45, 49

ovstatus 45, 47

ovstop 45, 49

ping 66, 215, 422, 1009
showMenuItem 1013
Sign In/Out Audit Log 1016
Sign Out 1016
Status Details 1010
Status Poll 1011
traceroute 422, 1012-1013
communication
configuring
protocols 66
Regional Manager 56
disconnecting Regional Manager 59
regions 89
team 277
troubleshooting configuration file imports 1032
verifiying settings 108-111
Communication Community String form 90
Communication Configuration
command 1002
form 66-67, 73, 78-79, 95, 106, 1000
Communication Default Community String form 75
communication protocols, configuring 95-96
Communication Region form 80, 89, 92-94
Communication Region SNMPv3 Settings form 93
Communication Settings menu item 264
community strings
configuring default 73
loading from file 106
setting 90, 103
default 75
nodes 96
regions 80
SNMPv2c 72
Community Strings tab 90
comparison criteria, deduplication 562, 707, 841
comparison maps, configuring 945
Comparison Parameters form 569, 848
comparison parameters, deduplications 346, 566, 711, 845
Comparison Params form 714
components
Additional Filters Editor 183, 190-191, 198, 206-207
Causal Engine 216, 307-308, 342, 728
Custom Filter Editor 898
event pipeline 309
nodes 246
Payload Filter Editor 467, 472, 483, 506, 522, 529, 554, 574, 600, 618, 629, 636, 652, 657, 668, 687, 696, 744, 757, 762, 773, 797, 808, 815, 829, 852
conclusions, down incident 310
Condition Listener stage 308
configuration
auto-discovery 128-130, 132-134, 139, 141, 143
creating objects 29
dampening incident 356, 657
deduplication 341-342, 346, 566, 711, 845
describing incident configurations 453, 598, 742
disabling 26
enabling 26, 425
enrich incident 355
event 328
exporting settings 1022, 1027-1028
importing settings 1022, 1030, 1032
incident 425, 730
management event 330

maps 279, 945
menu nesting 873
monitoring
reports 1006
settings 246
moving settings 1030
node group map settings 279, 281
pairwise 339, 342, 347-348, 350
rate 342, 347, 569, 714, 848
Regional Manager 56
reports 864
suppressing incident 355, 600
verifying
DNS 125
wizard 23
workspaces 23
Configuration Item menu item 264
Configuration Poll
command 1003
Configuration workspace 263
Configuration Workspace view 988
configure
unresponsive nodes 138
configuring
access 262
actions
interface groups 628, 772
Lifecycle Transition Action form 415, 572, 717, 851
management events 850
node groups 521, 667, 807
payload filters 668, 773, 808
remote events 716
SNMP trap incidents 571
Actions menu 874, 877
audit log files 274
auto-discovery rules 141, 143
automatic actions 414
background images 287
communication
management server 56
protocols 66
communication protocols
devices 95
nodes 96
communication regions 80
community strings
nodes 103
community strings for regions 90
console 20
credential settings
nodes 105
regions 94
Custom Incident Attributes 465, 504, 611, 650, 790
Custom Poller 929, 932, 935, 939, 943, 945, 950
default
community strings 73
device credentials 78
line graph settings 295
protocol settings 67
SNMPv3 settings 76
device profiles 135
discovery
process 136
seeds 158-159
DNS 125
excluded
addresses 155

interfaces 157
Global Manager forwarding 554, 696
Global Network Management 52
hostname filters for regions 89
IANA ifType-MIB definitions 208
incidents 306-307, 312, 315, 328, 330, 339, 348, 351, 353, 437, 441, 446, 588, 591-592, 728, 732, 735-736
actions 414, 482
basic settings 438, 730
card health causal rules 410
causal rules 385, 387, 389
child 387
child filters 367, 382, 389, 410
Cisco correlation rules 382
correlation filters 375
correlation rules 358
custom correlations 356
Custom Incident Attributes 755
dampening settings 472, 511, 545, 618, 657, 673, 762, 797, 829
deduplication 562, 707, 841
diagnostics 421-422, 673, 813
enrichment settings 462, 467, 501, 506, 542, 608, 647, 684, 752, 757, 787, 826
interface settings 483, 599
Lifecycle Transition Action form 415, 572, 717, 851
management events 743, 850
node groups 667
node settings 634, 778
pairwise 347
parent filters 360, 382, 410
payload filters 522, 613, 629, 668
rates 567
remote events 583, 586
SNMP traps 431, 436
source node filters 404
source object filters 397
suppression settings 454, 489, 529, 600, 636, 744, 779, 815
interface
groups 772
monitoring 223
settings 453, 743
IP address ranges 86
Java actions 897
JavaScript actions 896
Launch Actions 872, 878
line graph settings 295
management
event display 728
stations 583
maps 279, 286
menus 296
MIB
exceptions 919
variables 925
monitoring
behavior 216
reports 1006
settings 246
network
devices 124, 428
region protocols 79
node
connectivity 286
monitoring 234
name resolution 140

node group
actions 521
diagnostics selections 527
map settings 279, 281
status calculations 243, 282
Path View maps 291
payload filters 574, 652, 668, 718, 757, 773, 792, 852
RAMS 868
rates
management events 846
remove events 712
remote events 584, 586
security settings for SNMPv3 297
sign-in access 262-263, 266, 268, 271
SNMP Line Graphs 872, 890
SNMP traps 297, 299-301, 304, 431, 436, 438, 453, 488
subnet connection rules 152
target status 244
threshold monitoring 228, 231, 240
user interface 259
USM 76-77, 92-93, 104
connecting management servers 52, 55
connection
editor file 176
phase 114
connection state, verifying 62
connections
adding 176
deleting 176
subnet rules 122, 152, 154
connectivity
checking addresses 246
configuring node group maps 286
console
accessing
actions 263
forms 263
workspaces 263
adding discovery seeds 159, 162
administrator tools 22
configuring 20, 259
defining node groups 182
disabling user access 272
launching from URLs 957
loading
MIBs 916
SNMP trap incident configurations 430
opening 277
running outside
help 51
tools 954
signing
in 278
out 278
unloading MIBs 915
contacting
devices 72-73
controlling
access 262, 264
Actions menu 872
address discovery 155
node names 140
SNMP traps 431-432
correlating incidents
Custom Correlation Configuration form 356
duplicates 346
filters 375
pairs 347, 351

rules 358
correlation rates
remote NNM 6.x/7.x events 712
SNMP trap incidents 567
correlation rules example 382
counts
incident frequency 347
management events 846
remote NNM 6.x/7.x events 712
SNMP trap incidents 567
creating
a Custom Poller Policy 950
additional group filters
interface 198, 206
interfaces 207
nodes 183, 190-191
auto-discovery rules 130
configuration objects 29
Custom Poller
collections 930
policies 947
report collections 950
report groups 950
discovery seed file 162
filter expressions 898
incident
categories 444, 589, 733
configurations 429
families 445, 590, 734
interface groups 181, 197
IP ranges 130
management
event configuration 728
station configuration 583
menu nesting 873
MIB expressions 920, 936
node group map settings 282
node groups 181-182
payload filter expressions 792
credential settings
configuring 94, 105
default 78
criteria
deduplication 562, 707, 841
CSV 196
custom attributes
Full URL 887
interface 214
interface objects 213
node objects 213
nodes 214
Custom Attributes tab 213, 887
Custom Correlation Configuration form 356, 1000
Custom Filter Editor
buttons 898
components 898
Custom Incident Attributes
configuring
enrichment 465, 504, 611, 650, 790
parent incident filters 360
Deduplication Comparison Parameters form 346, 566, 711, 845
Full URLs 888
message formats 446, 452, 592, 597, 736, 741
out-of-box 312
parameters 415, 446, 592, 736, 859
Rate Comparison Parameters form 347, 569, 714, 848

special characters 419
Custom Poller
configuration unique identifiers 1032
configuring 929
collection settings 932
comparison maps 945
MIB expressions 936, 939
thresholds 943
creating
collections 930
policies 947
report collections 950
report groups 950
description
administrators 929
disabling 930
enabling 930
loading MIBs 916, 919
specifying MIB variables 935
Custom Poller Collection form 930
Custom Poller Collection tab 932
Custom Poller Configuration form 1000
cycle times, discovery 138

D

dampening, configuring incident 356, 472, 511, 545, 657, 687, 762, 797, 829
data
backing up 1036
RAMS 868
restoring 1036
database
auto-discovery 120
object
identifiers 889
passwords 262, 266-267, 272
role mapping 262, 266-267, 272
user names 262, 266-267, 272
decision tree, node name 118
Dedup stage 309
Deduplication Comparison Parameters form 346, 566, 711, 845
deduplication configuration 342, 346, 562, 566, 707, 711, 841, 845
Deduplication form 346
Default Community String tab 73
Default Device Credentials form 78
defaults
community strings
configuring 73
setting 75
credential settings 78
health monitoring 219
line graph settings 295
node group map settings 279
protocols
settings 67
traffic control 219
secure zone proxy settings 67
USM settings 76-77
views 957
defining
interface
attributes 214
groups 197
node groups 181, 196
target status 244
definitions
flow 422
SNMP traps 429

deleting
Boolean operators 193
connections 176
discovery seeds 173
incidents 1037
management
event configuration 728
station configuration 583
node group map settings 282
nodes 172
SNMP trap configuration 437
table rows 29
user accounts 272
denying access 272
dependencies, configuration area 1022
describing incidents 453, 598, 742
Description attribute 453, 598, 742
detecting
interface changes 174
determining
node
names 118
user account roles 263
developer licenses 1019
developing filters 181
Device Credentials tab
Communication Regions form 94
Specific Node Settings form 105
Device Profile form 135, 174
devices
adding group filters 183, 190-191, 207
communication protocols 95
configuring
communication protocol settings 96
community strings 103
credential settings for regions 94
default credentials 78
network 428
node names 140
diagnostics 422
discovering
SNMP 129
specific 128
vendors 132
excluding from discovery 134
profile configuration 1032
profiles 124, 135
retry behavior 72-73
timeout behavior 72-73
types 135
DHCPv6 50
Diagnostic Selection form 422
diagnostics, configuring selections for node groups 673
diagnostics, incident
configuring 421-422, 527, 813
out-of-box 422
direct management mode 251, 257
directories, NNMi 1036
directory service
configuration 262, 268, 271
database 262
disabled
cards 426
incidents 426
disabling
configurations 26
Custom Poller 930
global monitoring 217
SNMP trap configuration 431
user accounts 272

discard rate incidents 232
disconnecting communication with Regional Manag
discovering networks
approach 127
auto-discovery regions 120
description 113
device profiles 124
discovery seeds 119, 158
everything 130
excluding
addresses 133
devices 134
interfaces 121, 157
IP addresses 121
initiating discovery 179
intervals 116
node name choices 116
prerequisites 124-125
process 114
routers 128
SNMP devices 129
specific devices 128
switches 128
vender devices 132
discovery
adjusting intervals 138
checking
discovery seed status 167
node state 166
progress 166
configuration unique identifiers 1032
configure unresponsive nodes 138
configuring 136
excluding
interfaces 121
IP addresses 121
initiating 179
IP address ranges 146
node name choices 116
protocols 120
seeds 118-119, 128, 158-159, 162, 164, 167
SNMP system object ID ranges 150
verifying 166-167, 169-170
Discovery Configuration form 136, 138, 167, 1000
Discovery IP Range form 146
Discovery Seeds
deleting 173
form 159
tab 158
Discovery System Object ID Range form 150
Discovery, Spiral 113
adjusting discovery interval 138
configure unresponsive nodes 138
configuring 127, 135
discovering
everying 130
routers and swtiches 128
excluding
interfaces 134
IP addresses 134
SNMP object IDs 134
fine-tuning 124
object ID ranges 150
process 114
speeding up 125
displaying
background images 290-291
incidents 307

management
events 728
station configurations 583
map views 291
MIB
files 915
monitoring configuration reports 1006
SNMP traps as root causes 443, 588
distribution graphs, status 249
DNS
names 116
prerequisite 125
dollar sign ($) 419
Domain Name System 125
down
card 426
incidents 310
interface 426
drag and drop, Filter Editor 360, 375, 389
duplicate incidents, correlating 346
Dynamic Host Configuration Protocol 50

E

editing
management
event configuration 728
station configuration 583
node group map settings 282
SNMP trap configuration 437
user accounts 267, 270-271
editors
Additional Filters Editor 183, 190-191, 193, 198, 206-207
Child Incident Filter 367
Custom Filter Editor 898
Layer 2 Connection 176
MIB Expression Editor 925
parent incident filters 360
Payload Filter 613, 618, 652
EIGRP 120
email, launching maps from 954
embedding views within Web portals 954
Enable Node Component Monitoring 217
Enable Router Redundancy Group Monitoring 217
Enable State Polling 217
enabling
communication with directory service 268
configurations 26
Custom Poller 930
incident configurations 425
JavaScript 277
SNMP trap configuration 431
encoding, character 882, 954
encryption, SSL 262
ENDP 120
engine, SNMPv3 297
enrich incidents 355
enriching incident configuration 467, 506, 608, 650, 792
enrichment settings, configuring 462, 501, 542, 611, 684, 752, 755, 787, 790, 826
Enterasys Discovery Protocol 120
enterprise-specific traps, SNMP 441
envattrs attribute 955
environment variables 955, 1021
equals sign (=) 419
errors
monitoring interfaces 232
Regional Manager 64

establishing
criteria for incoming incidents 342
monitoring behavior 219, 223, 234
rate correlation
remote NNM 6.x/7.x events 712
SNMP trap incidents 567
ESX, VMware 209
events
configurations 328, 634, 636, 716, 728
duplications 346
management 330, 730, 846
pipeline 309
everything, discovering 130
examples
adding group filters
interface 193
nodes 193
causal rules 410
correlation rules 382
custom attributes in Full URLs 887
incident message formats 452, 597, 741
IPv4 addresses 146, 155
MIB expressions 925, 939
threshold monitoring 232
exceptions, MIB 919
Excluded Interfaces tab 157
Excluded IP Address filter 134, 155
excluding from discovery
addresses 133-134, 155
auto-discovery rules 128
interfaces 121, 134, 157
IP addresses 121
object IDs 134
problem devices 125
executing launch items 1013
exporting
configuration settings 1022, 1027
expressions
Filter Editor 367, 375, 389, 397, 404
MIB
configuring compariosn maps 945
creating 920, 936
editor 939
specifying line labels 893
testing 924, 938
thresholds 943
variables 911
viewing 920
payload filter 792
Payload Filter Editor 467, 483, 506, 574, 613, 629, 757, 773, 808, 852
extending capabilities
NNMi 872
polling 219, 223, 234
extending licenses 1020

F

fail-over health issues 64
families
creating incident 445, 590, 734
out-of-box 589
Family attribute 443, 445, 733-734
fault
monitoring 223, 234
FDP 120
fields, Lookup 27-28
files
allowedOids.conf 431
backing up 1036
creating discovery seed 162

hostnolookup.conf 125
ipnolookup.conf 125
launchsamples.jsp 954
loading node settings 106
log 47, 274
MIB 429, 915
nms-local.properties 277
nms-roles.properties 275
nms-users.properties 273, 275
nnmDocs_en.war 51
PathConnections.xml 291
restoring 1036
text 196
filter expressions, creating 898
filtering
node groups 209
filters
auto-discovery 143
configuring
causal rules 389, 410
child 367, 382, 389, 410
correlation 375, 382
parent 360, 382, 410
payload 613, 629, 652, 718, 792, 852
regions 89
source node 404
source object 397
creating
additional 183, 190-191, 207
forwarding 53
developing 181
excluding from discovery
interfaces 121, 157
IP addresses 121, 134, 155
interface
groups 197-198, 206, 957, 972, 982, 985
Inventory Workspace view 982
node
groups 982, 985
node groups 181, 957, 961, 972
out-of-box 209, 212
SNMP trap forwarding 299
Find Node menu item 264
Firefox
launching
All Incidents view 963
Configuration form 1000
Configuration Workspace view 988
default table views 957
Incident form 997
Incident view 961
Interface form 993
Inventory Workspace view 982
IP Address form 994
Management Mode Workspace view 985
MIB browser 1014
Monitoring Workspace view 972
Node form 990
Node Group forms 998
Subnet form 995
Topology Maps Workspace view 965
Troubleshooting Workspace view 975
opening console 277
flow
definitions 422
form
Report Collection 950
Report Groups 950

formats, incident messages 446, 452, 592, 597, 736, 741
forms
accessing
console 263
Account Mapping 266-267, 270
Auto-Discovery Rules 141
Communication Community String 90
Communication Configuration 66-67, 73, 78-79, 95, 106, 1000
Communication Default Community String 75
Communication Region 80, 89, 92-94
Communication Region SNMPv3 Settings 93
Comparison Parameters 569, 848
Custom Correlation Configuration 356, 1000
Custom Poller Collection 930
Custom Poller Configuration 1000
Deduplication 346
Deduplication Comparison Parameters 346, 566, 711, 845
Default Device Credentials 78
Device Profile 135, 174
Diagnostic Selection 422
Discovery Configuration 136, 138, 167, 1000
Discovery IP Range 146
Discovery Seeds 159
Discovery System Object ID Range 150
Excluded IP Address 155
Global Network Management 1000
Incident
Custom Attributes tab 887
launching 997
parameters 415, 446, 578, 723, 859
Incident Category 444, 589, 733
Incident Configuration 306, 346-347, 426, 437, 569, 636, 848, 863, 1000
Incident Family 445, 590, 734
Included Address Range 86
Integration Configuration 954
Interface
launching 993
parameters 415, 578, 723, 859
Interface Group 197-198, 206-207
Interface Settings 223
IP Address 994
Java Action 897
JavaScript actions 896
Launch Actions 878
launching outside NNMi 990
Layer 2 Connection 415, 578, 859
Lifecycle Transition Action 415, 572, 717, 851
Management Event 728
Management Event Configuration
Causal Engine 728
Custom Incident Attributes 755, 790, 815
dampening settings 762, 829
deduplication 841
enrichment settings 752, 787, 826
incident actions 772, 807, 850
interface settings 743
node settings 778
payload filters 757, 852
suppression settings 744, 779
Menu Item 874, 878, 890
Menu Item Contexts 877-878, 890
MIB 913
MIB Expression 919-920, 924
MIB Expressions 936, 938

MIB Specification 893
MIB Variable 915
Monitoring Configuration 216-217, 1000
Node
Custom Attributes tab 887
discovery state 166
launching 990
parameters 415, 578, 723, 859
Node Group 183, 190-191, 998
Node Group Map Settings 281-282, 287
Node Group Map Settings form 286
Node Group Status Settings 244
Node Settings 234
opening
Lookup fields 27
Pair Item Configuration 353
Pairwise 347
Pairwise Configuration 351
Principal 272
RAMS Configuration 868
Rate Comparison Parameters 347, 714
Region Device Credentials 94
Region Hostname Filter 89
Regional Manager Connection 56
Remote NNM 6.x/7.x Event Configuration 584, 586 599-600 608, 613 634, 647 668, 675 68
SNMP Line Graph 890
SNMP Trap Configuration 437, 453-454, 462, 465, 529, 542
Specific Node Device Credentials 105
Specific Node Settings 96, 103, 105
Status Configuration 243-244, 1000
Subnet 995
Subnet Connection Rule 152
Thresholds Settings 228, 231, 240
toolbar 29
Trap Forward Configuration 297, 299
Trap Forwarding Destination 302
Trap Forwarding Filter Expressions 300
Trap Forwarding Filters 300, 304
URL 898
User Account 267, 270
User Interface Configuration 259, 878, 890, 1000
VLAN 415, 578, 859
forwarding
Global Managers 53, 554, 696
SNMP trap 297, 301
Foundry Discovery Protocol 120
frequency, tracking incident 347
Full URL
capability attributes 886
custom attributes 887
Custom Incident Attributes 888
path view attributes 890
syntax 889
functions, incident message 736

G

gathering incidents 307
generating incidents
card disabled 426
configuration 307
interface disabled 426
performance threshold 426
values within messages 415, 859
generic traps, SNMP 441-442

- get command 66
- getEnvAttr attribute 955
- global control attributes for interface configuration25
- Global Manager
  - configuring 53, 554, 696
  - connecting 52, 55-56
  - verifying
    - health 63
- global monitoring, setting 217
- Global Network Management
  - actions 30
  - connecting
    - Global Manager to Regional Manager 55
    - multiple management servers 52
  - creating
    - forwarding filters 53
    - node groups 181
  - disconnecting communication 59
  - error messages 64
  - exporting configuration settings 1022
  - form 1000
  - troubleshooting 61
  - verifying
    - connection state 62
    - health 63
- global settings, Ping Sweep 139
- granting user access 266
- graphical user interface
  - configuring 259
- graphs
  - MIB expression results 938
- greater than sign (>) 419
- groups
  - defining
    - report 950
  - interface
    - actions 628
    - filters 198, 206
  - interface filters 207
  - node filters 181-183, 190-191, 193, 196, 209, 211, 279, 281
  - node groups 243-244, 282
  - out-of-box
    - interface 212
    - node groups 209
  - router redundancy 246
- Guest role 263-264
- GUI
  - configuring 259

H

- Headine report, accessing 231
- health
  - application fail-over issues 64
  - monitoring
    - network 215-216
    - State Poller 246
  - monitoring network 216
  - verifying
    - management server 63
    - NNMi 1018
- help
  - running outside console 51
- high availability
  - NNMi 125
- Hostname attributes 118
- hostname resolution phase 118
- hostnames
  - assigning wildcards 89
  - configuring filters 89

discovery seeds 119
hostnolookup.conf file 125
Hot Standby Router Protocol (HSRP) 246
HSRP
objects 246
HTTPS encryption 262
Hypertext Transfer Protocol Secure encryption 262
Hypothesis Engine stage 308

I

iAdvanced licenses 1019
IANA ifType-MIB definitions 208
ICMP
device settings 96
enabling nodes 96
node configuration 1002
ping command 119, 215
regional settings 80
retries
default 67
node 96
regions 80
timeouts
default 67
nodes 96
region 80
traffic control 66
icons
Causal Engine 216
Lookup fields 27
identifying
attribute pairs 353
author configuration 1032
communication configuration 1032
IDs, SNMP object 134, 150, 196, 441-442
IfTypes
view 208
IKEv2 50
images, background
configuring 287
out-of box 290
scaling 291
troubleshooting 291
Important Nodes group 209
importing
community string assignments 106
configuration settings 1022, 1028, 1030, 1032
SNMP assignments 106
Incident Category form 444, 589, 733
Incident Configuration form 306, 346-347, 426, 437, 569, 636, 848, 863, 1000
incident configurations
reports 864
troubleshooting 863
Incident Family form 445, 590, 734
Incident form
Custom Attributes tab 887
launching 997
opening Incident Configuration form 863
parameters 415, 446, 578, 723, 736, 859
Incident Receiver stage 309
Incident views 961, 963
incidents
accident parameters 415, 446, 578, 592, 723, 736, 859
archiving 1037
attributes 312, 330, 339, 348
card disabled 426
categories 589

Causal Engine 308
CiscoLinkDown 350, 711, 714, 845, 848, 888
CiscoLinkUp 347
closing 310
configurations 330, 425, 730
  reports 864
  troubleshooting 863
  unique identifiers 1032
configuring 306-307, 312, 315, 328, 351, 437, 441, 588, 728, 732
  actions 414, 482, 521, 571, 628, 772, 807
  causal rules 385, 387, 389
  child 387
  child filters 367, 389
  correlation filters 375
  correlation rules 358
  custom correlations 356
  Custom Incident Attributes 790
  deduplication 562, 707, 841
  diagnostics 421-422, 673, 813
  enrichment settings 467, 483, 501, 506, 542, 608, 684, 755, 787
  interface settings 599
  Lifecycle Transition Action form 415, 572, 717, 851
  management events 730
  node groups 521, 667
  node settings 634, 647, 650, 778
  pairwise 347
  parent filters 360
  payload filters 613, 629, 668, 773
  rates 567
  remote events 583, 586, 716
  SNMP traps 431, 436
  source node filters 404
  source object filters 397
  suppression settings 454, 489, 529, 600, 636, 779
controlling incoming trap visibility 431
correlating
  duplicate 346
  pairs 347, 351
  rates 712
creating
  categories 444, 589, 733
  families 445, 590, 734
dampening configurations 356, 472, 511, 545, 657, 762, 797, 829
deleting 1037
descriptions 453, 598, 742
discard rate 232
down 310
enrichment 355
families 589
filters 961
gathering 307
generating
  card disabled 426
  functions 736
  interface disabled 426
  performance threshold 426
message format 446, 452, 592, 597, 736, 741
message functions 736
organizing 443, 733
out-of-box
  actions 30
parameters 883
reducing incoming 341-342
root causes 443, 588

severity 446, 591, 735
suppressing 355, 675
tracking
frequency 347
viewing 426
views
workspace 961
Incidents workspace 263
Included Address Range form 86
including CIAs in message format 597
incoming
incidents 341
SNMP traps
controlling 431-432
managing 427
incremental configurtion changes 1022
initiating discovery 179
instant-on licenses 1019
integrating NNMi with other applications 954
Integration Configuration form 954
interface
adding
custom attributes 213
management mode 254, 256
new hardware 174
Autopass 1020
capability attributes 886
configuring
ifTypes 208
monitoring 223, 246
settings for incidents 453, 483, 599, 743
user 259
creating groups 181
custom attributes 214
defining groups 197
detecting changes 174
direct management mode 257
excluding from discovery 121, 134, 157
extending polling scope 219, 223, 234
groups
configuring actions 628
monitoring
configuration 246
out-of-box
actions 30
groups 212
parameters 883
removing 174
restarting management 250
reviewing group definitions 197
stopping management 250
suppressing incident configuration 600
unmanaged 253
utilization 232
viewing
disabled incidents 426
Interface form
launching 993
parameters 415, 578, 723, 859
Interface Group
form 197-198, 206
Interface Group form
adding filters 207
interface groups
actions 30
adding Boolean operators 193
configuration unique identifiers 1032
configuring
actions 772
dampening settings 467, 472, 506

enrichment 462, 744, 752
incident actions 482
suppression 465
creating
filters 198, 206
interface groups 181
defining 197
filters 197, 957, 972, 982, 985
out-of-box
actions 30
groups 212
parameters 883
Interface Settings form 223
Interface Settings tab 675
interface type configuration unique identifiers 1032
Interface Type Filters tab 197
interfaces
adding
filters 207
Internet Control Message Protocol 66
Internet Explorer
opening console 277
Internet Key Exchange 50
intervals, discovery 116, 138
inventory
verifying discovery 169-170
Inventory workspace 263
Inventory Workspace view 982
IP Address form 994
IP addresses
configuring ranges 86
direct management mode 257
discovering
discovery seeds 119
networks 116
discovery ranges 146
excluding from discovery 121, 134, 155
parameters 883
unmanaged 253
verifying
monitoring configuration 246
SNMP configuration 106
IP Security 50
ipnolookup.conf.conf file 125
IPsec 50
IPv4 addresses 50, 122, 126
IPv4 tunnel 152
IPv6 addresses 50, 126
ISDN Interfaces group 212
Island Node Groups 209, 211
iSPIs
NNM iSPI Network Engineering Toolset
actions 30
credentials 78, 94, 105
dampening 545, 687, 829
deduplication 562
diagnostics 421-422, 527
incidents 356, 421
interface groups 472, 618, 762
management events 330, 762, 797, 829
node groups 196, 209, 511, 527, 657, 673, 797, 813
nodes 214, 422
purchasing 953
remote NNM 6.x/7.x events 618, 657, 673, 687
roles 263
services 46
traps 433, 472, 511, 545, 562

NNM iSPI Performance for Metrics
CIAs 312
incidents 310, 426
interface groups 197
interfaces 223
management event configurations 330
node groups 181-182, 196
nodes 234
report collections 950
report groups 950
setting default monitoring 219
State Poller 215
threshold monitoring 228, 231-232, 240
NNM iSPI Performance for Traffic
interface groups 197
node groups 181-182
purchasing 953-954
isRunning command 1017

J

Java
configuring actions 897
regular expressions 360
Java Action form 897
JavaScript
configuring actions 896
enabling 277
JavaScript Actions form 896
Jython methods 415, 446, 578, 592, 723, 736, 859

L

Launch Actions
configuring 878
form 878
limiting 883
menu item 872
launching
Quick Start Configuration wizard 23
launching from URLs
commands 1002-1003, 1006, 1009-1013, 1015-1017
console 957
environment variables 955
forms 990, 993-995, 997-998, 1000
line graphs 1004
menu items 1001
MIB browser 1014
views 954, 957, 961, 963, 965, 972, 975, 982, 985, 988
launchsamples.jsp file 954
Layer 2 Connection
attributes 176
form
parameters 415, 578, 736, 859
parameters 883
Layer 2 Discovery, verifying 170
Layer 2 Neighbors view 975
Layer 3 Discovery, verifying 170
Layer 3 Neighbors view 975
LDAP configuration 262, 268, 271
less than sign (<) 419
Level 1 Operator role 263
Level 2 Operator role 263
library, documentation 1004
licenses
extending 1020
issues 64
tracking 1019
Lifecycle Transition Action form 415, 572, 717, 851

limiting
  Launch Actions 883
  neighbor information 133
limits
  incident message characters 446
  licenses 1019
  Tools menu access 264
  URL
    access 264
line graphs
  actions 893
  configuring 872
  default 1004
  launching outside NNMi 1004
  testing MIB expressions 924
line labels, SNMP line graph actions 893
Link Aggregation group 212
listing object instances 28
loading
  community strings from file 106
  custom attributes 214
  MIBs 912, 916, 919
  SNMP trap
    definitions 429
    incident configurations 429-430
locales
  settings 1022
log files 47, 274
logins, bypassing 957
Lookup fields 27-28

M

maintaining NNMi 1018
Management Address attributes 118
Management Event Configuration form 728
  configuring
    Custom Incident Attributes 790
    deduplication 841
    enrichment settings 787, 826
    incident actions 772, 807, 850
    node settings 778
    payload filters 852
    suppression settings 779, 815
  Custom Incident Attributes 755
  dampening settings 762, 829
  enrichment settings 752
  interface settings 743
  Payload Filter Editor 757
  suppression settings 744
management event configurations, out-of-box330
Management Event form 728
management event incidents, configuring 730, 743
management mode
  assigning to interface or address 254, 256
  not managed 257
  out of service 257
  stopping or starting 250
  views 251
Management Mode workspace 263
Management Mode Workspace view 985
management server
  connecting to another server 52, 55-56
  deleting discovery seeds 173
  exporting configuration settings 1022
  importing configuration settings 1022, 1030
  moving configuration settings 1030
  SNMPv3 settings 297

verifying
connection state 62
health 63
viewing
MIBs 912
viewing MIBs 912
Management Stations
configuring
remote 583
SNMPv3 297
view 583
managing
incidents 307
SNMP traps 427
map views
configuring Path View 291
Device Profiles 135
navigating 108-111
mapping roles 262
maps
configuring 279, 291, 945
modifying device symbols 124
node group map settings 279, 281, 287, 291
Path View 291
updating 171
memory options 1028
menu access, controlling 264
Menu Item Contexts form 877-878, 890
Menu Item form 874, 878, 890
menu items
Actions menu 872
configuration unique identifiers 1032
configuring 296
creating nesting 873
launching outside NNMi 1001
menu nesting, creating 873
menus, configuring 296, 1032
messages, error 64
messages, incidents
formats 446, 452, 592, 597, 736, 741
parameters 446, 592, 736
MiB
file 429
MIB
available 911
browser 911, 1014
configuring 919
exceptions 919
expressions 911, 920, 924, 938, 943, 945
file 429-430, 915
form 913
loading 430, 912, 916, 919
supported 911
unloading 915
variables 911, 913, 929, 935, 943
view 912
viewing loaded 912
web page 916
MIB Expression Editor 925, 939
MIB Expression form 919-920, 924
MIB Expressions form 936, 938
MIB Expressions view 920
MIB IANA ifType definitions 208
MIB II sysName Values 116
MIB Specification form 893
MIB Variable form 915
Microsoft Internet Explorer
opening console 277
Microsoft Windows Systems node group 209

MLDv2 50
modes, management 250
modifying
  configuration settings 1022
  incident configurations
    NNM 6.x/7.x traps 328
    pairwise 339, 348
monitoring
  configuration
    reports 1006
    settings 246
  faults 223, 234
  interface 223
  network
    devices 216
    health 215-216, 219
    object status distribution 249
  nodes 234
  performance 223, 234
  router redundance groups 246
  State Poller health 246
  suspending all 217
  thresholds 228, 231-232, 240
Monitoring Configuration form 216-217, 1000
Monitoring Settings command 1006
Monitoring workspace 263
Monitoring Workspace view 972
moving configuration settins 1030
Mozilla Firefox
  launching
    All Incidents view 963
    Configuration form 1000
    Configuration Workspace view 988
    default table views 957
    Incident form 997
    Incident view 961
    Interface form 993
    Inventory Workspace view 982
    IP Address form 994
    Management Mode Workspace view 985
    MIB browser 1014
    Monitoring Workspace view 972
    Node form 990
    Node Group form 998
    Subnet form 995
    Topology Maps Workspace view 965
    Troubleshooting Workspace view 975
  opening console 277
multicast requirements 50
multiple
  browser sessions 277
  management servers 52
  objects 29

N

Name attribute 732
names
  changing user 267, 270-271
  incident configurations 441, 588, 732
navigating
  map views 108-111
  table views 108-111
neighbor information, limiting 133
nesting menu items 296, 873
network
  configuring devices 428
  discovering
    approach 127
    auto-discovery regions 120
    description 113

device profiles 124
discovery seeds 119, 158
everything 130
excluding addresses 133
excluding devices 134
excluding interfaces 157
node name choices 116
prerequisites 124-125
process 114
routers 128
SNMP devices 129
specific devices 128
switches 128
vender devices 132
monitoring
health 215
object status distribution 249
naming nodes 140
regions 79
updating topology 113
viewing management modes for objects 251
Network Engineering Toolset
actions 30
credentials 78, 94, 105
dampening 545, 687, 829
deduplication 562
diagnostics 421-422, 527
incidents 356, 421
interface groups 472, 618, 762
management events 330, 762, 797, 829
node groups 209, 511, 527, 657, 673, 797, 813
nodes 214, 422
purchasing 953
remote NNM 6.x/7.x events 618, 657, 673, 687
roles 263
services 46
traps 433, 472, 511, 545, 562
Network Management Framework 442
Network Overview
view 965
Network Time Protocol Daemon 52, 61
Networking Infrastructure Devices
node group 209
view 965
nms-local.properties file 277
nms-roles.properties file 275
nms-users.properties file 273, 275
nmsdbmgr process 45
NNM iSPI Network Engineering Toolset
actions 30
credentials 78, 94, 105
dampening 545, 687, 829
deduplication 562
diagnostics 421-422, 527
incidents 356, 421
interface groups 472, 618, 762
management events 330, 762, 797, 829
node groups 196, 209, 511, 657, 673, 797, 813
nodes 214, 422
purchasing 953
remote NNM 6.x/7.x events 618, 657, 673, 687
roles 263
services 46
traps 433, 472, 511, 545, 562
NNM iSPI Performance for Metrics
CIAs 312

incidents 310, 426
interface groups 197
interfaces 223
management event configurations 330
node groups 181-182, 196
nodes 234
report
collections 950
groups 950
setting default monitoring 219
State Poller 215
threshold monitoring 228, 231-232, 240
NNM iSPI Performance for Traffic
interface groups 197
node groups 181-182
nnmbackup.ovpl command 1036
nnmcommconf.ovpl command 106
nnmcommload.ovpl command 106
nnmconfigexport.ovpl command 1022, 1027, 1030
nnmconfigimport.ovpl command 1022, 1027-1028, 1030
nnmconnedit.ovpl command 176
nnmDocs_en.war file 51
nnmhaclusterinfo.ovpl command 47
NNMi
6.x/7.x
configuring actions 667-668, 716
configuring dampening settings 687
configuring enrichment settings 684
configuring events 584, 586
configuring rates 712
configuring remote events 583, 634
displaying events as root cause incidents 443, 588
gathering incidents 307
incident configurations 306, 328, 588, 599, 636, 647, 675
viewing management stations 583
actions 30
Advanced
changing management mode 256
global monitoring 217
Global Network Management 30, 52, 64, 181, 554, 696, 1022
IPv6 addresses 126
Management Stations view 583
monitoring router redundancy groups 246
Path View maps 291
port aggregation 121
RAMS 868
backing up 1036
checking
health 1018
CIAs 312
components 308
controlling access 262
denying access 272
directories 1036
extending capabilities 872
incident configurations 312
integrating with other applications 954
launching from URLs 955, 957
maintaining 1018
password 262, 266-267, 272
processes 44-45
properties files 275
restoring 1036
role mapping 267, 272
services 44, 46

SNMP trap varbinds 305
tracking licenses 1019
troubleshooting access 273, 275
user name 262, 266-267, 272
verifying running 1017
NNMi Status command 264, 1015
nnmincidentcfg.ovpl command 299-300, 302, 304, 429
nnmlicense.ovpl command 1020
nnmloadattributes.ovpl command 214
nnmloadmib.ovpl command 912, 919
nnmloadnodegroups.ovpl command 182, 196
nnmloadseeds.ovpl command 162, 164
nnmnodedelete.ovpl command 172
nnmrestore.ovpl command 1036
nnmsetcdmuserpw.ovpl command 59
nnmsetcmduserpw.ovpl command 275, 1027-1028, 1030
nnmtrimincidents.ovpl command 1037
node Component
direct management mode 254
Node Component
direct management mode 257
monitoring 217
node components
restarting management 250
stopping management 250
Node form
Custom Attributes tab 887
discovery state 166
launching 990
parameters 415, 578, 723, 736, 859
Node Group form
adding filters 183, 190-191
launching 998
node group map
configuring
connectivity 286
settings 279, 281
views 279
Node Group Map
background images 291
menu item 264
Node Group Map Settings form 281-282, 286-287
Node Group Overview
view 965
Node Group Status Settings form 244
node groups
actions 30
configuration unique identifiers 1032
configuring
actions 807, 813
dampening settings 511, 797
diagnostics selections 527, 673
enrichment settings 501, 647, 826
map settings 281
maps 279, 281, 286
SNMP trap settings 488
status 243-244, 282
suppression settings 489
connectivity 286
creating 181-182
defining 181
filters 181, 957, 961, 972, 982, 985
getting status 1010
maps 286-287, 290
out-of-box 209
parameters 883
populate 181

protocol traffic control 228, 231, 234, 240
verifying 181
Node Groups
view 182
node names
decision tree 118
discovery choices 116
Node Settings
tab 675
Node Settings form 234
nodename attribute 1003
nodes
adding
custom attributes 213, 887, 955
filters 183, 190-191, 193
assigning hostname wildcards 89
capability attributes 886
components 246
configuration settings 1010
configuring
CIAs 504, 650
communication protocol settings 95-96
community strings 103
connectivity 286
credential settings 105
diagnostics 422
enrichment settings 787
group status calculations 243, 282
management event incidents 778
monitoring 234, 246
name resolution 140
SNMP traps 488
SNMPv3 settings 104
source node filters 404
suppression settings 779
threshold monitoring 240
creating groups 181
custom attributes 214
defining groups 182, 196
deleting 172
discovering
description 113
name choices 116
specific nodes 158
state check 166
filtering
all incidents view 963
groups 209
group maps 290
ICMP configuration 1002
identifying 114
incidents 415, 859
Island Node Groups 211
management mode 257
MIB
supported 911
variables 913
monitoring
configuration 246
not managed 252
out-of-box
actions 30
parameters 883
populating 140
rendering 291
reports 1006
restarting management 250
selecting 108-111
SNMP configuration 1002
status 1011

stopping management 250
threshold monitoring 240
unmanaged components 254
Non-SNMP Devices node group 209
Nortel
switches 422
Not Managed mode 252-254, 257
Notification stage 309
notifications, sending SNMP 428
nsloopup 125
NTPD 52, 61

O

objects
adding custom attributes 213, 887, 955
capability attributes 886
creating 27, 29
deleting 29
excluding IDs from discovery 134
Full URL syntax 883
HSRP 246
identifiers 889
listing 28
management modes 254, 256
network 249
reports 1006
SNMP system object ID ranges 150
source 397
specifying SNMP ID 196, 441-442
State Poller 215
status distribution 249
tracked 246
types 883
viewing
management modes 251
VRRP 246
Open Shortest Path First 50, 120
opening
console 277
forms 27
Incident Configuration form 863
operating systems
audit log files 274
command-line access 275
discovery seeds 164
environment variables 1021
help systems 51
port numbers 277
operators
Boolean 193
Filter Editor 367, 375, 389, 397, 404
roles
Level 1 263-264
Level 2 263-264
optional filters, adding 898
options
MIB Expression Editor 925
OR variable 193
organizing incidents 443, 733
OSPF 120
OSPFv3 50
out-of-box
actions 30
background images 290
CIAs 312
configurations 26
diagnostics 422
event configurations 328
incident
categories 443, 589, 733

configurations 312, 315, 425
families 445, 589-590, 734
interface
groups 212
management event configurations 330
node
groups 209
pairwise configurations 339, 348
SNMP trap varbinds 305
subnet connection rules 154
user roles 262-263, 266
view filters 209
Out of Service mode 252-254, 257
over-utilization, monitoring 232
ovjboss process 45-46, 49
ovjboss services 47
OVsPMD process 45
ovstart command 45, 49
ovstatus command 45, 47
ovstop command 45, 49

P

Pair Item Configuration form 353
Pair Items tab 353
Pairwise
form 347
stage 309
Pairwise Configuration form 339, 348, 351
pairwise configurations 339, 342, 347-348, 350-351, 353
parameters
backup 1036
deduplication 346, 566, 711, 845
incident
actions 415, 446, 578, 592, 723, 736, 859
CIAs 415, 859
interface 415, 859
Layer 2 415, 859
messages 446, 592, 736
nodes 415, 859
not visible from forms 415, 578, 859
interface groups 883
interfaces 883
IP addresses 883
Layer 2 Connection 883
line graphs 1004
node groups 883
nodes 883
rate 347, 569, 714, 848
restore 1036
SNMP MIB browser 1014
parent
incident filters 360, 382, 410
passing environment variables 955
passwords
changing 267, 270
configuring 262, 266, 271
Principal form 272
resetting 273, 275
Path View
attributes 890, 975
configuring maps 291
maps 279
RAMS 868
PathConnections.xml file 291
Payload Filter Editor
buttons 467, 472, 483, 489, 501, 506, 522, 529, 554, 574, 629, 636, 652, 657, 687, 696, 744, 757, 762, 773, 797, 808, 815, 829, 852

components 467, 472, 483, 506, 522, 529, 554, 574, 600, 629, 636, 652, 657, 687, 696, 744, 757, 762, 773, 797, 808, 815, 829, 852
configuring
enrichment 542, 684
filters 522, 574, 629, 652, 668, 718, 792
suppression 529
creating expressions 613
dampening configuration 618
editor 574
expressions 467, 483, 506, 629, 757, 773, 808, 852
PDFs, documentation library 51
percentage, target status values 244
performance
generating incidents 426
monitoring 223, 234
permanently disabling user accounts 272
permissions, role 263
phases, Sprial Discovery 114
ping command 66, 119, 215, 422, 1009
Ping menu item 264
Ping Sweep
configuring 139
description 119
pipe symbol (|)
character 419
pipeline, event 309
pmd process 45
Point to Point Interfaces group 212
Policies tab 947
policies, Custom Poller 947
Policy
create for Custom Poller 950
polling scope, extending 219, 223, 234
populating
Lookup fields 28
node
groups 181
names 140
portals, embedding views within 954
ports
numbers 277
prerequisites
discovery 124
DNS 125
IPv6 addresses 126
pairwise configurations 350
Principal form 272
problems
Causal Engine 308
processes
configuring discovery 136
description 45
NNMi 44
Spiral Discovery 114
starting 45
stopping 45
verifying running 45
profiles, device 124, 135
progress
discovery 166
properties
action server 571, 716, 850
files 275
protocols
auto-discovery rules 120
BGP 50
communication 66
defaults 67

devices 95
configuring network regions 79
DHCP 50
region 80
SNMP 72
proxy servers, SNMP 95
proxy settings, secure zone 67, 80, 96
purchasing HP iSPIs 953-954

Q

Quick Find 27-28
Quick Start Configuration wizard 23
Quick View 27

R

RAMS
Configuration form 868
Path View 868
ranges
IP addresses 86, 146
SNMP system object ID 150
system object ID 150
Rate Comparison Parameters form 347, 714
rate configuration 342, 347, 567, 569, 712, 714, 846, 848
Rate Configuration tab 347, 569, 848
rate correlation
management events 846
remote NNM 6.x/7.x events 712
SNMP trap incidents 567
Rate stage 309
rate, discard 232
real-time data 868, 1002-1003, 1010-1011
reasons for closing down incidents 310
rediscovery interval
adjusting 138
description 116
reducing incoming incidents 341-342
redundancy groups, monitoring router 246
refreshing
console window 277
Region Device Credentials form 94
Region Hostname Filter form 89
Regional Manager
connecting
Global Manager 55-56
multiple management servers 52
creating forwarding filter 53
disconnecting communication 59
error messages 64
verifying
connection state 62
health 63
Regional Manager Connection form 56
regions
assigning hostname wildcards 89
configuring
address ranges 86
communication 80, 90, 103
credential settings 94
hostname filters 89
network protocols 79
USM settings 77, 92-93, 104
Regions tab 79-80, 86, 89-90, 92-93
registration attributes for interface configuration 259
registration, custom correlation 356
Relate stage 309
relative URLs 291

remote
  event configuration 328, 583-584, 634, 636
  management stations 583
Remote NNM 6.x/7.x Event Configuration
  form 584, 586, 599-600, 608, 613, 634, 647, 668, 687
  tab 628, 667, 675
Remote NNM 6.x/7.x Events tab 707, 716
removing
  incidents 1037
  interfaces 174
rendering nodes 291
replacing configuration settings 1022
Report Collection form 950
report collections, configuring 950
Report Groups form 950
report groups, configuring 950
reports
  audit 274
  incident configuration 864
  NNMi running 1017
  Trap Analytics 433
Request for Comments 442
requirements
  multicast 50
  pairwise configurations 350
reserved characters 882, 954
resolution, node name 140
Resolver stage 309
restarting management 250
restoring
  configuration settings 1028
  NNMi 1036
  system role 275
results
  discovery 166, 170
  discovery seeds 167
retry behavior 72-73
reviewing node group definitions 182
RFC 442
RFC 3986 882, 954
roles
  accessing
    menus 264
  assigning 262
  configuration unique identifiers 1032
  Guest 263
  mapping 262, 266
  operator
    Level 1 263
    Level 2 263
  system 275
  user
    changing 267, 270-271
    deleting 272
    determining account 263
    out-of-box 262-263, 266
  Web Service Client 263
root causes
  SNMP traps 443, 588
round-robin DNS 125
Route Analytics Management System 868
Router Redundancy Member
  form 246
routers
  Cisco 422
  discovering networks 128, 158
  Island Node Group 211
  monitoring 246

Routers
  monitoring 246
  node group 209
  view 965
rows
  deleting 29
rules, auto-discovery 120
  configuring 141
    basic settings 143
    Ping Sweep 139
  custom 128
  discovery seeds 119
  everything 130
  IP addresses 133, 146
  object IDs 134, 150
  overview 127
  routers 128
  SNMP devices 129
  switches 128
  vendors 132
rules, causal 410
rules, configuring causal 385
rules, correlation 356, 358, 382
rules, subnet connections 122, 152, 154
rules, URLs 882, 954
rules, W3C 882, 954
running
  help outside console 51
  verifying
    processes 45
    services 47
runtime licenses 1019
runTool 1001

S

scaling map background images 291
scope, extending polling 219, 223, 234
SDK licenses 1019
secure
  zone proxy settings 67, 80, 96
Secure Sockets Layer encryption 262
security check 264
security, user-based model
  configuring 66, 77, 79, 92-93, 104
  default settings 76
  importing SNMPv3 settings 106
  Quick Start Configuration wizard 23
  SNMP prerequisite 124
  SNMPv3 settings 297
seeds, discovery 118-119, 128, 158-159, 162, 164, 167, 173
selecting
  nodes 108-111
semicolon (;) character 419
services
  NNMi 44, 46
  ovjboss 46-47
  starting 49
  State Poller 135, 215, 246
  stopping 49
  verifying running 47
setting up command-line access 275
settings
  auto-discovery
    basic 143
    Ping Sweep 139
  communication protocols 67
  community strings 75, 90, 103

configuring
  credential settings 94
  dampening 687
  enrichment 501, 542, 684, 752, 787, 790
  incidents 599-600, 608, 618, 675, 730
  Java actions 897
  JavaScript actions 896
  management events 730, 743
  menu nesting 873
  node settings 778
  nodes 488
  payload filters 483
  SNMPv3 regions 92-93
  suppression 489, 529, 744, 779, 815
dampening incidents 618, 797
default
  credentials 78
  ICMP 67
  line graph 295
  secure zone proxy 67
  SNMP 67
  USM 77
enrichment 826
exporting configuration 1022
importing configuration 1022, 1028, 1030, 1032
interface
  enrichment settings 462
  exclusion settings 157
  suppression settings 454
Java actions 897
JavaScript actions 896
line graphs 295
locales 1022
management event incidents 730
menu nesting 873
node
  group maps 279, 281, 291
  group status 244
  name resolution 140
regional 80
SNMPv3 297, 438
suppressing configuration 675
verifying communication 108-111
severity, incident 446, 591, 735
Show All Incidents menu item 264
Show All Open Incidents menu item 264
show commands 422
Show Members menu item 264
showForm 990
showLineGraph 1004
showMain 957
showMenuItem command 1013
showMibBrowser 1014
showView 957
sign-in access, configuring 262-263, 266, 268, 271, 275
Sign In/Out Audit Log command 1016
Sign In/Sign Out Audit Log menu item 264
Sign Out
  command 1016
signing in
  console 278
signing out from console 278
Simple Network Management Protocol 66
Single Sign-On 52, 61
Smart Plug-ins, purchasing 953-954
snapshot, configuration settings
  exporting 1027
  importing 1028

SNMP
- adding trap configuration 437
- analyzing traps 433
- configuring
  - actions 571
  - basic attributes 730
  - dampening 545
  - deduplication 562
  - default settings 76-77
  - device settings 96
  - discovery 136
  - enrichment settings 542
  - interface settings 453
  - node settings 488
  - nodes 104
  - rates 567
  - suppression settings 454, 462, 465, 529
  - traps 297, 299-301, 304, 437
- default community strings 73
- deleting trap configuration 437
- discovering
  - devices 129
  - networks 114
- displaying
  - traps 443, 588
- editing trap configuration 437
- enabling nodes 96
- importing assignments 106
- incidents 427
- line graph actions 893
- loading
  - incident configuration 430
  - trap definitions 429
- managing incoming traps 427
- MIB browser 1014
- node configuration 1002
- phase 118
- ports
  - default 67
  - nodes 96
  - regions 80
- prerequisites 124
- protocol version 72
- proxy servers 95
- regional settings 80
- retries
  - default 67
  - nodes 96
  - regions 80
- sending notifications 428
- specifying object ID 196, 441-442
- State Poller 215
- system object ID ranges 150
- timeouts
  - default 67
  - nodes 96
  - regions 80
- traffic control 66
- traps
  - analyzing 433
  - basic attributes 730
  - configuring actions 571
  - configuring dampening 545
  - configuring deduplication 562
  - configuring enrichment 542
  - configuring filters 300
  - configuring forwarding 297, 299, 301-302
  - configuring incidents 306, 315, 429-430, 436, 438
  - configuring interface settings 453

configuring node settings 488
configuring suppression settings 454, 462, 465, 529
controlling incoming 431-432
gathering incidents 307
including CIAs 452, 597, 741
rates 567
specifying object ID 441
varbinds 305
verifying configuration 106
SNMP Line Graph Actions menu item 872
SNMP Line Graph form 890
SNMP Trap Configuration
form 437-438, 453-454, 462, 465, 529, 542
tab 436
SNMP Trap Receiver stage 309
SNMPv1 23, 66, 73, 90, 103, 106, 124, 428, 441-442
SNMPv2c 23, 66, 72-73, 90, 103, 106, 124, 428, 441-442
SNMPv3 72, 76-77, 92-93, 104, 106, 124, 297
Software Loopback Interfaces group 212
source node filters, configuring 404
source node, configuring
diagnostic selections 673
incident configurations 488
source object filters, configuring 397
sources, background images 290
space character 419
special characters in action arguments 419
Specific Node Device Credentials form 105
Specific Node Settings
form 105
tab 95, 106
Specific Node Settings form 96, 103
Spiral Discovery 113
adjusting discovery interval 138
configure unresponsive nodes 138
configuring 127, 135
discovering
everything 130
routers and switches 128
excluding
interfaces 134
IP addresses 134
SNMP object IDs 134
fine-tuning 124
process 114
SNMP prerequisites 124
speeding up 125
system object ID ranges 150
SSL encryption 262
SSO 52, 61
stages
Causal Engine 308
event pipeline 309
stand-alone help 51
starting
discovery 179
management 250
processes 45
services 49
State Mapping tab 945
State Poller
default monitoring 219
enabling 217
service 135, 215, 246
states
verifying discovery 166
stations, remote management 583

status
  colors 216
  configuring
    node groups 243
    targets 244
  Island Node Group 211
  network object distribution 249
  unmanaged
    addresses 253
    cards 253
    interfaces 253
    node components 254
    nodes 252
  verifying
    address connectivity 246
    health 1018
    NNMi 1015
    processes 45
Status Configuration form 243-244, 1000
Status Details
  command 1010
  menu item 264
Status Poll
  command 1011
  menu item 264
stopping
  management 250
  processes 45
  services 49
Store Bulk stage 309
strings, community
  configuring default 73
  setting 75, 90, 103
subnet
  connection rules 122, 152, 154
Subnet Connection Rule form 152
Subnet form 995
success, discovery seed 167
suppressing incidents 355, 675
suppression settings, incident 454, 489, 529, 600, 744, 779, 815
suspending all monitoring 217
Sweep, Ping 119, 139
switch
  discovering networks 128
switches
  Cisco 422
  Island Node Group 211
  Nortel 422
Switches
  view 965
synchronization, clock 61
Synoptics Network Management Protocol 120
system account password, changing 273, 275
System Name attributes 118
system object ID ranges 150
System role 275

T

table views
  default 957
  deleting rows 29
  navigating 108-111
tabls
  Hostname Filters 89
tabs
  Action 716
  Action Configuration 415, 572, 717, 851
  Additional Filters 183, 190-191, 198, 206-207

Auto-Discovery Rules 146
Community Strings 90
Custom Attributes 213, 887
Custom Poller Collection 932
Default Community String 73
Default Device Credentials 78
Device Credentials 94
Discovery Seeds 158
Excluded Interfaces 157
Interface Settings 675
Interface Type Filters 197
Management Event 728
Management Event Configuration 728
Node Settings 675
Pair Items 353
Pairwise Configuration 339, 348, 351
Policies 947
Rate Configuration 347, 569, 848
Regions 79-80, 86, 89-90, 92-93
Remote NNM 6.x/7.x Event 707
Remote NNM 6.x/7.x Event Configuration 628, 667
Remote NNM 6.x/7.x Events 716
Report Collection 950
SNMP Trap Configuration 436
Specific Node Settings 95, 106
State Mapping 945
Trap Forwarding Destinations tab 301
target status 244
tasks
auto-discovery 128-130, 132-134, 141, 143
configuring
communication protocols 66
management events 728
node group map settings 281
subnet connection rules 152
controlling SNMP object ID ranges 150
Custom Poller 930
discovery 136
incident
configuration 307
monitoring network health 215-216
remote NNM 6.x/7.x event configuration 583-584
SNMP traps
configuring incidents 437, 586
managing incoming 427
subnet connection rules 152
verifying discovery 166
team communication 277
Telnet menu item 264
temporary licenses 1019
testing
MIB expressions 924, 938
text files 196
threshold
configuring 243
incidents 426
monitoring 228, 231-232, 240
target status 244
thresholds
Custom Poller 943
Thresholds Settings form 228, 231, 240
time period
incident frequency 347
management events 846
remote NNM 6.x/7.x events 712
SNMP trap incidents 567
timeout
behavior 72-73

- options 1028
- toolbar
  - form 29
- tools menu 264
- tools, console
  - administrator 20, 22
  - running outside 954
- topology maps
  - building 868
  - island node groups 211
  - updating 113, 171
- Topology Maps Workspace view 965
- traceroute command 422, 1012-1013
- Traceroute menu item 264
- Tracked Objects
  - monitoring configuration 246
- tracking
  - incidents
    - frequency 347
  - licenses 1019
- traffic control 66
- traffic volume issues 64
- Trap Forward Configuration form 297, 299
- Trap Forward Filtering Expressions 300
- Trap Forwarding Destination form 302
- Trap Forwarding Destinations tab 301
- Trap Forwarding Filters form 300, 304
- traps, SNMP
  - configuring
    - dampening 545
    - deduplication 562
    - forwarding 297, 302
    - forwarding filters 300, 304
    - incidents 315, 436-438, 571
    - interface settings 453
    - suppression settings 454, 462, 465, 529, 542
    - trap forwarding 299
  - controlling incoming 431-432
  - displaying 443, 588
  - gathering incidents 307
  - loading
    - definitions 429
    - incident configurations 429-430
  - managing incoming 427
  - object ID formats 441-442
  - specifying Object IDs 441
  - vabinds 305
- tree, node name decision 118
- troubleshooting
  - access 273, 275
  - Causal Engine 308
  - Global Network Management 61
  - importing configuration files 1032
  - incident configurations 863
  - URLs 291
- Troubleshooting workspace 263
- Troubleshooting Workspace view 975
- Type Enforcer stage 309

U

- UI
  - configuring 259
- under-utilization, monitoring 232
- unique identifiers 889, 1032
- unique key values 196
- UNIX
  - background images 290
  - DNS prerequisite 125
  - environment variables 1021

log files 47, 274
nms-local.properties file 277
nms-roles.properties file 275
nms-users.properties file 273, 275
nnmbackup.ovpl command 1036
nnmcommload.ovpl command 106
nnmDocs_en.war file 51
nnmhaclusterinfo.ovpl command 47
nnmlicense.ovpl command 1020
nnmloadseeds.ovpl command 162, 164
nnmtrimincidents.ovpl command 1037
NTPD 52, 61
PathConnections.xml file 291
unloaded MIBs 915-916
unloading MIBs 915
Unmanaged Addresses view 253
Unmanaged Cards view 253
Unmanaged Interfaces view 253
Unmanaged Node Components view 254
Unmanaged Nodes view 252
unresolved incoming SNMP traps 432
unresponsive nodes, discovery 138
updating
incidents 429
topology maps 113, 171
URL access
launching
commands 1002-1003, 1006, 1009-1013, 1015
console 957
forms 990, 993-995, 997-998, 1000
line graphs 1004
menu items 1001
message report 1017
MIB browser 1014
NNMi 954
Sign In/Out Audit Log command 1016
Sign Out command 1016
views 957, 961, 963, 965, 972, 975, 982, 985, 988
limitations 264
troubleshooting 291
URL Actions
controlling 872
custom attributes 955
Custom Incident Attributes 888
database object IDs 889
URL form 898
URLs
Full URL syntax 883, 886, 889
W3C rules 882, 954
user-based security model
configuring 66, 77, 79, 92-93, 104
default settings 76
importing SNMPv3 settings 106
Quick Start Configuration wizard 23
SNMP prerequisite 124
SNMPv3 settings 297
User Account form 267, 270
user accounts configuration unique identifiers1032
User Interface Configuration form 259, 878, 890, 1000
User Principals view 271-272
users
auditing activity 274
changing passwords 266, 273, 275
determining account roles 263
disabling accounts 272
editing accounts 267, 270-271
out-of-box roles 262-263, 266

restoring system role 275
USM
configuring 66, 77, 79, 92-93, 104
default settings 76
importing SNMPv3 settings 106
Quick Start Configuration wizard 23
SNMP prerequisite 124
SNMPv3 settings 297
utilization
WAN and interface 232

V

varbind values 347, 428, 452, 569, 597, 714, 741, 848, 888
varbinds, out-of-box 305
variables
environment 955, 1021
MIB 911, 913, 925, 929, 935, 939, 943
vendor devices
discovering 132
vendor specific traps, SNMP 442
verifying
address connectivity 246
communication settings 108-111
configuration for IP addresses 106
discovery
inventory 169
Layer 2 170
Layer 3 170
progress 166
results 166
seeds 167
states 166
DNS configurations 125
Global Manager health 63
monitoring configuration settings 246
nms-roles.properties file 275
NNMi
health 1018
NNMi running 1017
node groups 181
processes running 45
Regional Manager
connection state 62
health 63
services running 47
user account deletion 272
version numbers
SNMP protocols 72
view
MIB Expressions 920
viewing
event configurations 328
incident configuration reports 864
incident pair configurations 339, 348
interface disabled incidents 426
interface group
definitions 197
Java action settings 897
JavaScript action settings 896
management mode for objects 251
MIB
available 911
expressions 920
files 915
loaded 912
variables 913
node group
definitions 181
remote management stations 583

status distribution graphs 249
views
Configuration Workspace 988
controlling incoming trap visibility 431
ifTypes 208
incidents 863
Incidents 961, 963
Interface Group 197
Inventory Workspace 982
launching outside NNMi 957
Management Mode Workspace 985
Management Stations 583
Monitoring Workspace 972
Node Groups 182
predefined view filters 209
Topology Maps Workspace 965
Troubleshooting Workspace 975
Unmanaged Addresses 253
Unmanaged Cards 253
Unmanaged Interfaces 253
Unmanaged Node Components 254
Unmanaged Nodes 252
User Principals 271-272
workspaces 263
virtual
OSPF interface change 315
server 125
terminal (Telnet from client) 30
virtual IP Addresses
IPv4 tunnel 122, 152, 171
Node Groups 209
Virtual Router Redundancy Protocol (VRRP) 246
visibility
incoming traps 431
parameter strings 578, 736
VLAN
form 578
VLAN form 415, 859
parameters 736
VLAN Interfaces group 212
VMware ESX 209
Voice Interfaces group 212
volume, traffic 64
VRRP
objects 246

W

W3C rules for URLs 882, 954
WANs
Island Node Group 211
threshold monitoring 232
web page
loading MIBs 916
Web portals, embedding views within 954
Web Service Client role 263
web site
launching NNMi 955
well-configured DNS 125
wide area networks
Island Node Group 211
threshold monitoring 232
wildcards, assigning hostnames 89
Windows
background images 290
DNS prerequisite 125
environment variables 1021
log files 47, 274
nms-local.properties file 277
nms-roles.properties file 275
nms-users.properties file 273, 275

nnmbackup.ovpl command 1036
nnmcommload.ovp commandl 106
nnmDocs_en.war file 51
nnmhaclusterinfo.ovpl command 47
nnmlicense.ovpl command 1020
nnmloadseeds.ovpl command 162, 164
nnmtrimincidents.ovpl command 1037
NTPD 61
PathConnections.xml file 291
wizard, Quick Start Configuration 23
workspaces
accessing 263
configuration 23
Configuration 988
Inventory 982
Management Mode 251, 985
Monitoring 972
Topology Maps 965
Troubleshooting 975
views 263
World Wide Web Consortium 882, 954

Z

zone, secure 67, 80, 96

Java™
COMPATIBLE