

HP Network Node Manager i-series Software

for the Windows®, HP-UX, Linux, and Solaris operating system

Software Version: 8.11

Deploying NNMi by Example

Document Release Date: March 2009

Software Release Date: January 2009



Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

© Copyright 2009 Hewlett-Packard Development Company, L.P.

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

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

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

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

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

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

Trademark Notices

DOM4J® is a registered trademark of MetaStuff, Ltd.

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

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

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

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

UNIX® is a registered trademark of The Open Group.

Oracle Technology — Notice of Restricted Rights

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

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

Documentation Updates

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

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

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

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

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

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

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

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

Support

Visit the HP Software Support web site at:

www.hp.com/go/hpsoftwaresupport

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

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

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

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

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

To find more information about access levels, go to:

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

Contents

Initially Logging on to NNMi and Creating Users	11
Initial Log On	11
Creating User Accounts and Roles	13
Setting up Communication Configuration	17
Monitoring Basics	25
Creating an Interface Group for Monitoring	30
Applying a Polling Policy to an Interface Group	33
Testing the Polling Policy	36
Making Exceptions to Monitoring	38
Configuring Incidents, Traps and Automatic Actions	39
Configuring Incidents	39
Configuring Traps	41
Configuring Automatic Actions	42
Configuring the NNMi Console	47
Node Group Configuration	47
Configuring the Node Group Map	52
Configuring the User Interface	54
Maintaining NNMi	56
Backing up and Restoring NNMi Data	56
Exporting and Importing NNMi Configurations	58
Trimming Traps from the Database	60
Checking NNMi Health	61
Miscellaneous Tips	64
Possible Usage Scenarios	65
Management by Exception	65
Map Based Management	68
List Based Management	70
Conclusion	71

1 Deploying NNMi by Example

This document takes you on a tour, deploying NNMi 8.1x on a small network. This tour does not show an NNMi 8.1x deployment into a production network, rather it shows an NNMi 8.1x deployment in a small test lab. The steps discussed during this tour are similar to those you would take to deploy NNMi in a production network. You can better prepare yourself to deploy NNMi by reading this document and reviewing the screen shots.

Although this document does not show you how to migrate NNMi configurations from another NNM 6.x/7.x server to an NNMi server, it does explain how to complete a new NNMi installation. HP will soon release a separate document showing you how to migrate configurations from another NNM 6.x/7.x server to an NNMi server.

We recommend that you read through this document, then use the *NNMi 8.1x Deployment and Migration Guide* as a reference, as it contains many details that extend beyond the technical scope of this document.



Visit <http://h20230.www2.hp.com/selfsolve/manuals> to find the latest Deployment and Migration Guide for NNMi.

The Basic Steps: A Roadmap

This document assumes you have already installed NNMi, and does not cover installation. However, you must make sure that your server meets all the system prerequisites. Especially check the patch requirements and kernel parameters shown in the *HP Network Node Manager i Software System and Device Support Matrix*. The NNMi installation script does not check that your server meets these requirements before installing NNMi. Ignoring these requirements can cause issues after you complete your installation.

This document shows examples of an NNMi installation on an HPUX server. You must convert the paths and commands if you are using NNMi that is installed on a Windows server.

Your tour through the document sections include the following topics:

- 1 [Initially Logging on to NNMi and Creating Users](#) on page 11.
- 2 [Applying the License](#) on page 15.
- 3 [Setting up Communication Configuration](#) on page 17.
- 4 [Configuring Discovery](#) on page 19.
- 5 [Monitoring Configuration](#) on page 25.
- 6 [Configuring Incidents, Traps and Automatic Actions](#) on page 39.
- 7 [Configuring the NNMi Console](#) on page 47.
- 8 [Maintaining NNMi](#) on page 56.
- 9 [Checking NNMi Health](#) on page 61.
- 10 [Miscellaneous Tips](#) on page 64.
- 11 [Possible Usage Scenarios](#) on page 65.

This document does not include the following topics:

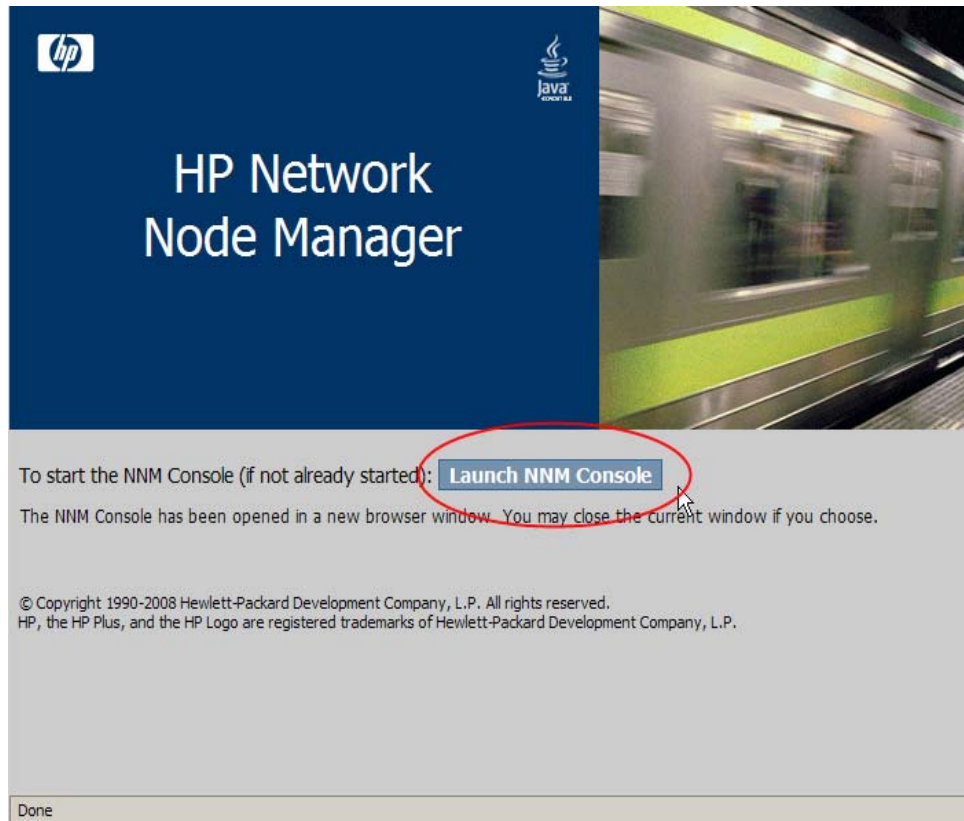
- Integration with other products such as HP OM, HP UCMDB, and other 3rd party products
- Configuring HA or Application Failover
- Configuring a remote Oracle database.
- NNM iSPIs such as NNM iSPI for Performance or NNM iSPI for MPLS).

Details for these steps can be found in the *Deployment and Migration Guide*.

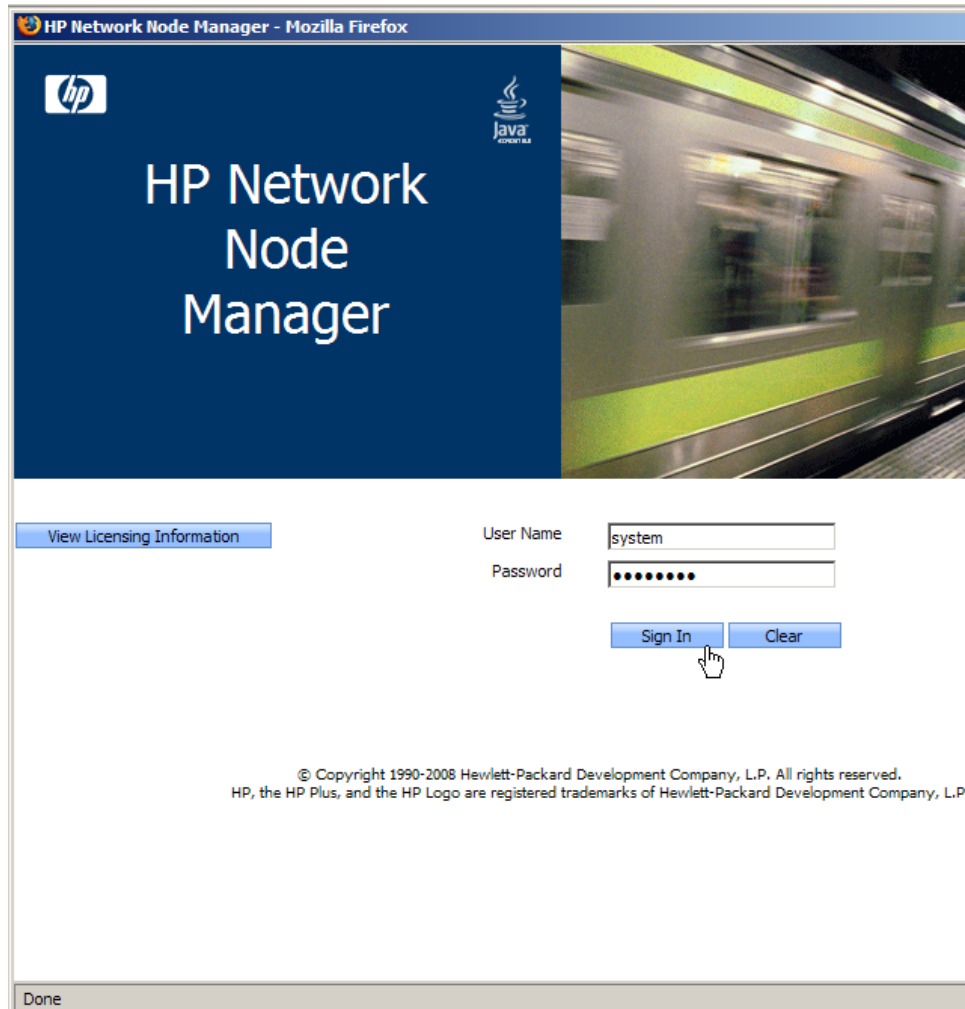
Initially Logging on to NNMi and Creating Users

Initial Log On

You access NNMi by using a browser such as Internet Explorer or Mozilla Firefox. Depending on the port you selected for communication when you installed NNMi, use a URL similar to
`http://<serverName>:<port number>/nnm.`



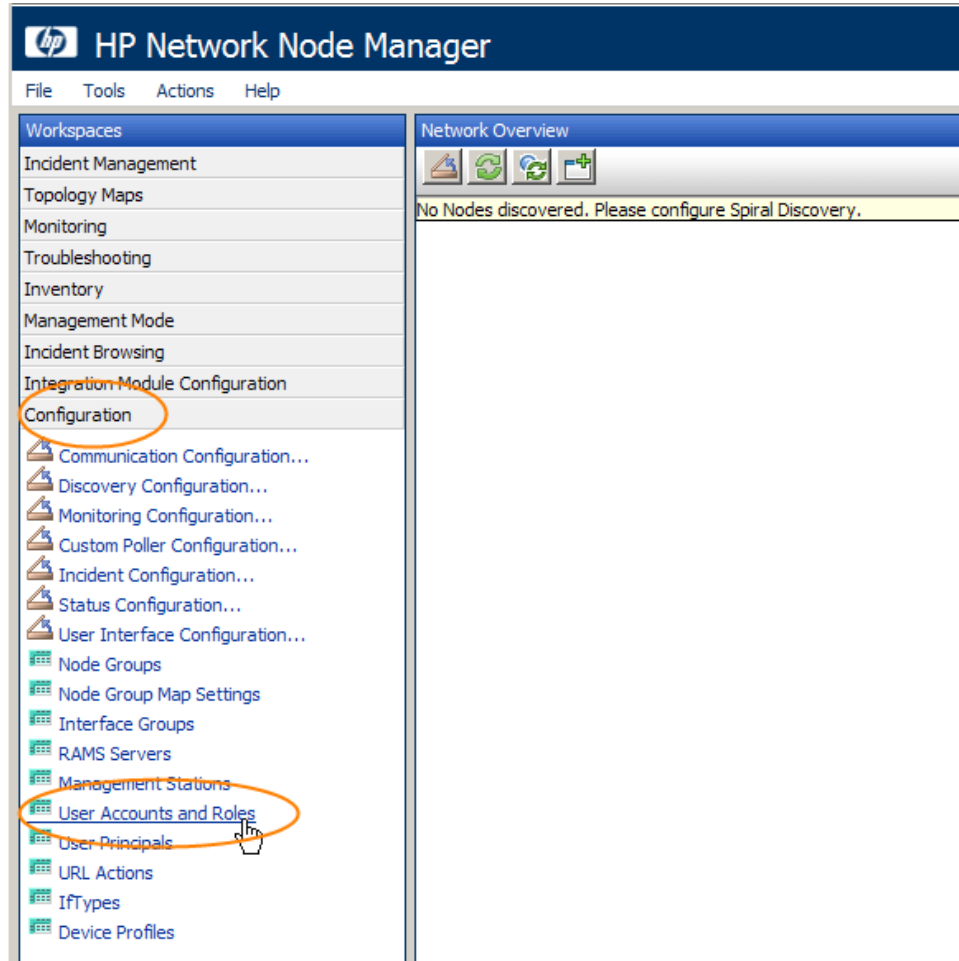
Click **Launch NNM Console** to log on to the NNMi console. Initially you must log on to the NNMi console with the system user name that you created during installation.



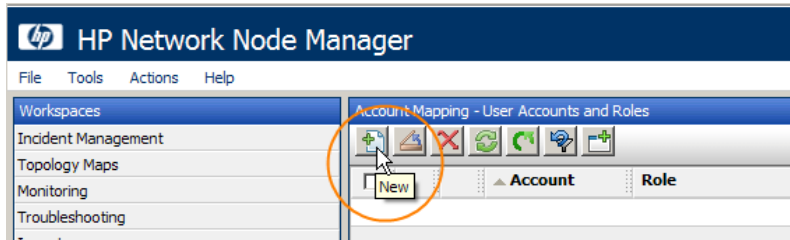
Creating User Accounts and Roles

HP recommends against using the system user name in most cases. You need to create and use an administrator account for most of your work. To do this, follow these instructions:

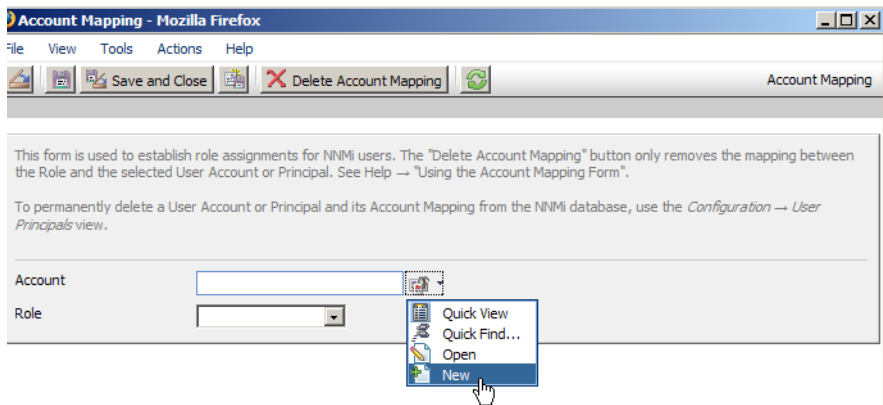
- 1 Click **Configuration** from the NNMi console.
- 2 Click **User Accounts and Roles**.



- 3 Click the **New** Icon to open the Account Mapping form.

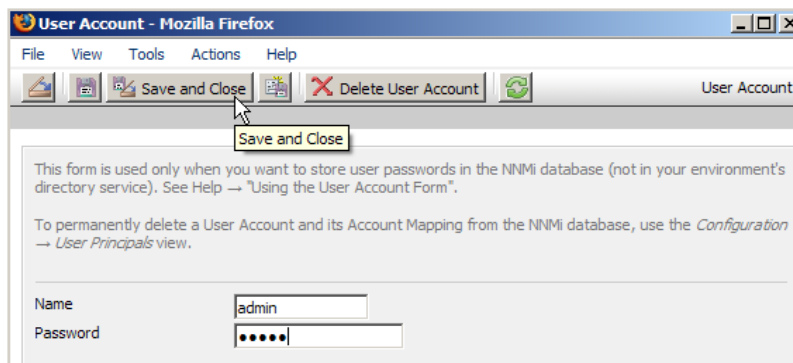


- 4 Use the pull-down menu to the right of the Account entry to select **New**.

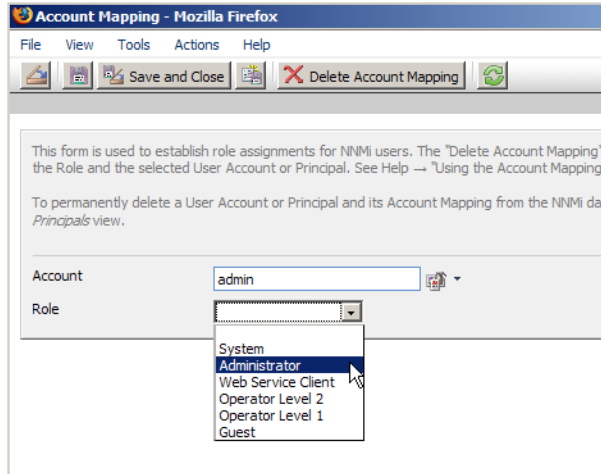


- 5 Type in a name and password. Although you can type in any name, for this example use **admin** for the name and **adminpw** for the password.

▶ NNMi now supports LDAP password accounts. You will not use LDAP during this tour. See the *NNMi 8.1x Deployment and Migration Guide* for further information.



- To configure this as an administrator account, select the **Administrator** Role, then click **Save and Close**.



You now have an Administrator account to use for the rest of this example.

Applying the License

For some deployments, you can use the instant-on license. The instant-on license enables NNMI for 250 nodes. If you want to do a larger test, you need to obtain a larger temporary license from the HP. After you get the temporary license from HP, you can easily apply it from the NNMI console. You can also install the license using the command line. The following command shows an example of installing the license using the `nnmlicense.ovpl` script:

```
nnmlicense.ovpl NNM -f ./mylicense.key
```

Backing up the Original Configuration

Your next step is to make a backup of the original configuration of NNMi before making any changes. This way, you can revert back to the original configuration should you need to. To do this, complete the following steps:

- 1 Create a directory on the NNMi server where you want to keep the original configuration files. For this example, create a directory called `/var/tmp/origconfig`.
- 2 Run the `nnmconfigexport.ovpl` command using the `-c` and `-f` options. The `-c` option specifies all configurations and the `-f` option specifies the directory.

The following command shows an example of running the `nnmconfigexport.ovpl` script: `nnmconfigexport.ovpl -u admin -p adminpw -c all -f /var/tmp/origconfig/`

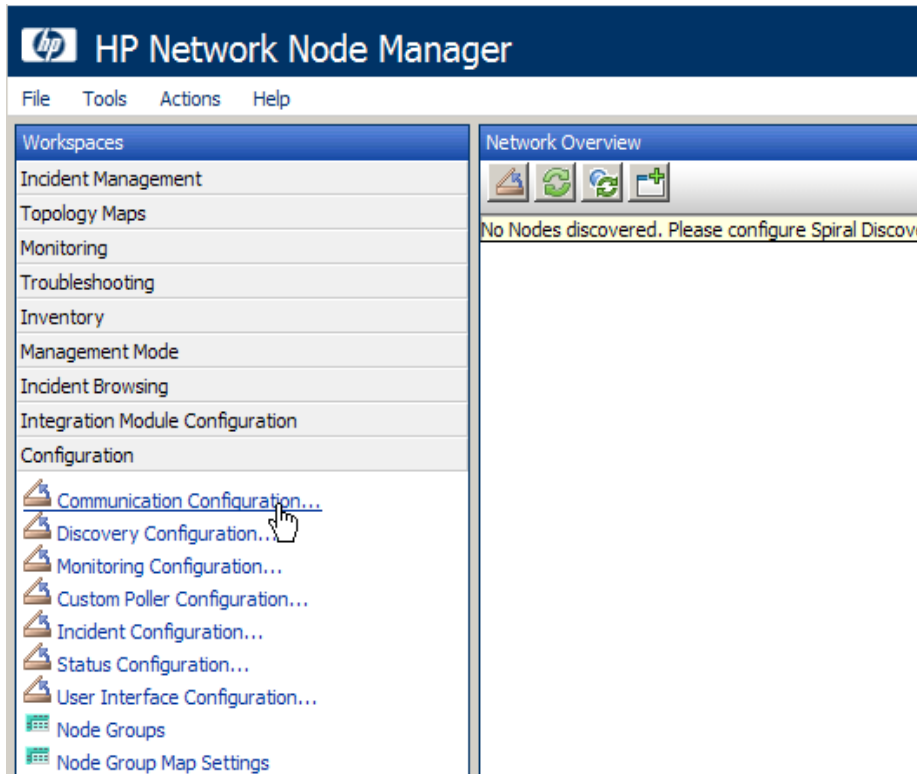
After you run the script as shown above, NNMi displays something similar to the following:

```
Successfully exported /var/tmp/origconfig/comm.xml.  
Successfully exported /var/tmp/origconfig/incident.xml.  
Successfully exported /var/tmp/origconfig/status.xml.  
Successfully exported /var/tmp/origconfig/urlaction.xml.  
Successfully exported /var/tmp/origconfig/ngmap.xml.  
Successfully exported /var/tmp/origconfig/ui.xml.  
Successfully exported /var/tmp/origconfig/ifgroup.xml.  
Successfully exported /var/tmp/origconfig/monitoring.xml.  
Successfully exported /var/tmp/origconfig/nodegroup.xml.  
Successfully exported /var/tmp/origconfig/custpoll.xml.  
Successfully exported /var/tmp/origconfig/station.xml.  
Successfully exported /var/tmp/origconfig/device.xml.  
Successfully exported /var/tmp/origconfig/rams.xml.  
Successfully exported /var/tmp/origconfig/account.xml.  
Successfully exported /var/tmp/origconfig/disco.xml.  
Successfully exported /var/tmp/origconfig/discoseed.xml.  
Successfully exported /var/tmp/origconfig/iftypexml.  
Successfully exported /var/tmp/origconfig/author.xml.
```


Setting up Communication Configuration

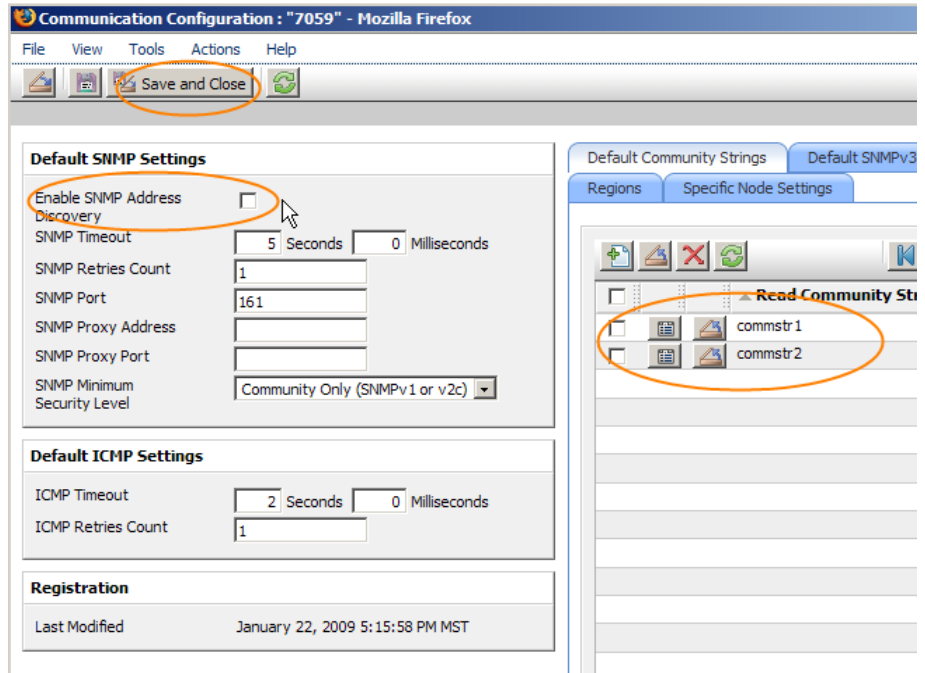
Your next step is to set up the communication configuration. By default, NNMi performs *SNMP community string discovery*. Most customers find the SNMP community string discovery to be a useful technique and find it easy to use. This next example shows you how to use this technique. Unlike previous versions of NNM, you do not configure a prioritized list of SNMP community strings. NNMi tries all possible strings simultaneously. The first community string that results in a response from a node is selected as the SNMP Community String for that node. For this example, you only configure the default community strings. You can implement more sophisticated solutions with this configuration, but in most cases, this is an adequate approach.

- 1 To begin, click **Configuration** from the NNMi console, then click **Communication Configuration**.



- 2 Click the **Default Community Strings** tab, then click the **New** icon.

- 3 Enter all of your SNMP read community strings here. The order of entry does not matter, as NNMi tries all possible strings simultaneously. You can also modify the default ICMP timeout and retry attempts here.
- 4 After making changes, click **Save and Close** to save your changes.



In patch 2 for 8.10 (versioned as NNMi 8.11), HP added a new feature called *Enable SNMP Address Discovery*. HP enabled this feature by default. When enabled, this is equivalent to the original SNMP address discovery behavior of NNMi 8.10. The benefit of adding this feature is that it provides the ability to disable SNMP address discovery when your situation requires you to do so. You can enable or disable this feature to achieve the following NNMi behaviors:

- *Enable SNMP Address Discovery 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.
- *Enable SNMP Address Discovery disabled*: If 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.

If you have Cisco devices using loopback addresses, consider unchecking this box to disable the SNMP address discovery feature. That way, NNMi will only try the loopback address for SNMP communication. Most NNMi users prefer this behavior over switching to a different address on the node.

Configuring Discovery

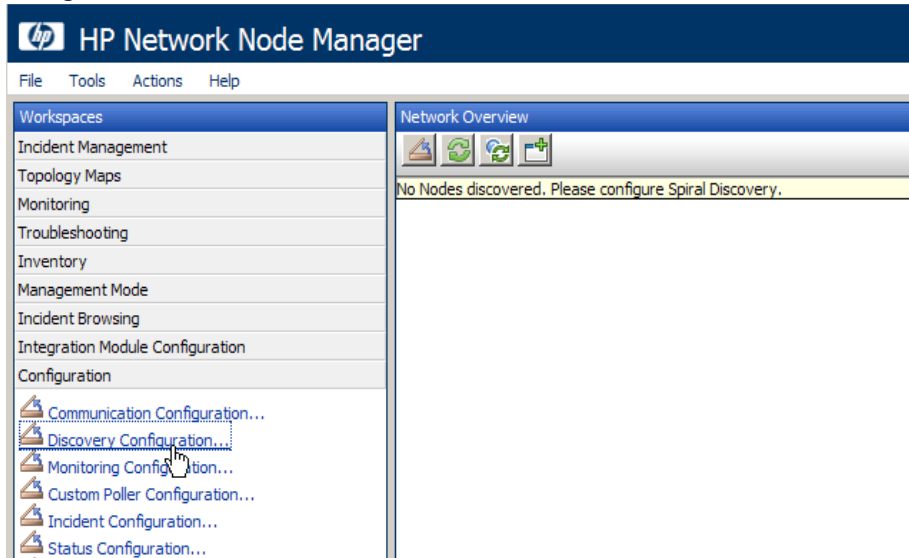
NNMi supports two methods of discovery, *automatic* and *list-based*. There are advantages to each method.

List-based discovery uses a list of node names or addresses as input and only discovers the nodes contained in that list. NNMi discovers no additional node names or addresses beyond those contained in this list. This method gives you control over what is discovered and managed by NNMi. Although each of these nodes is listed as a *seed*, nothing grows from these seeds. Another nice feature with seeds is that NNMi loads them even if their IP address is outside of the auto discovery range. If you load a seed as an IP address for a device, it is a good practice to specify the preferred management address (usually the loopback address with Cisco gear) as the seed.

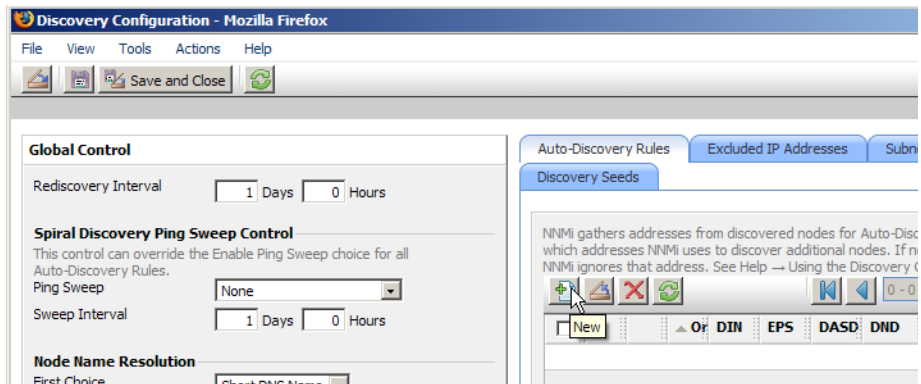
Automatic discovery finds nodes on the network based on user-specified criteria. you can configure NNMi to restrict discovered nodes based by address range, SNMP values (like system object ID), device type, and other methods. You can configure automatic discovery with a single seed node, although even this node is not required if you enable the optional *ping-sweep* feature.

The following example shows you an automatic discovery based on an address range. It also shows you how to load a couple of seed nodes as well.

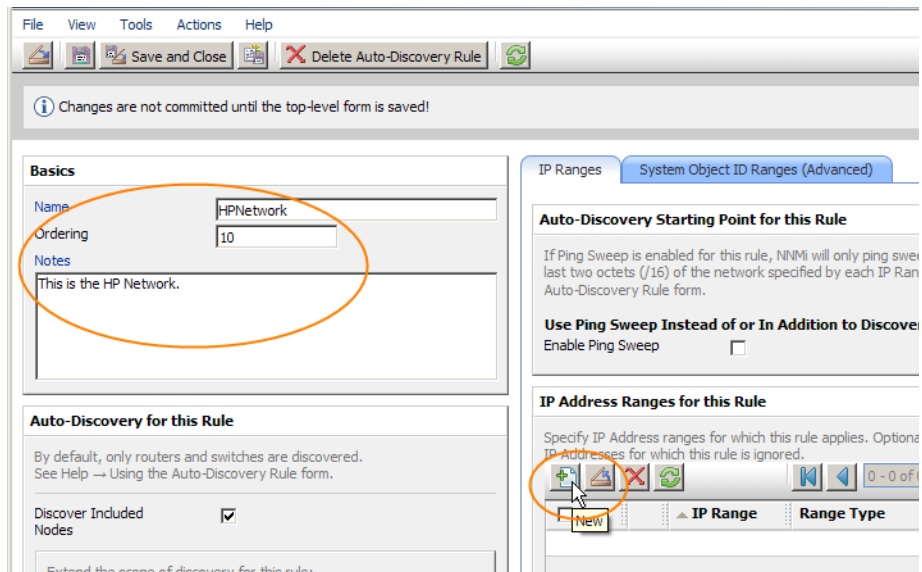
- 1 To begin, click **Configuration** in the NNMi console, then click **Discovery Configuration**.



- 2 Select the **Auto-Discovery Rules** tab, then click the **New** icon.



- 3 Fill out the Basics section, then click the **New** icon to open an entry screen for the IP Range in this rule. The value for Ordering doesn't matter as this example shows only one auto-discovery rule.

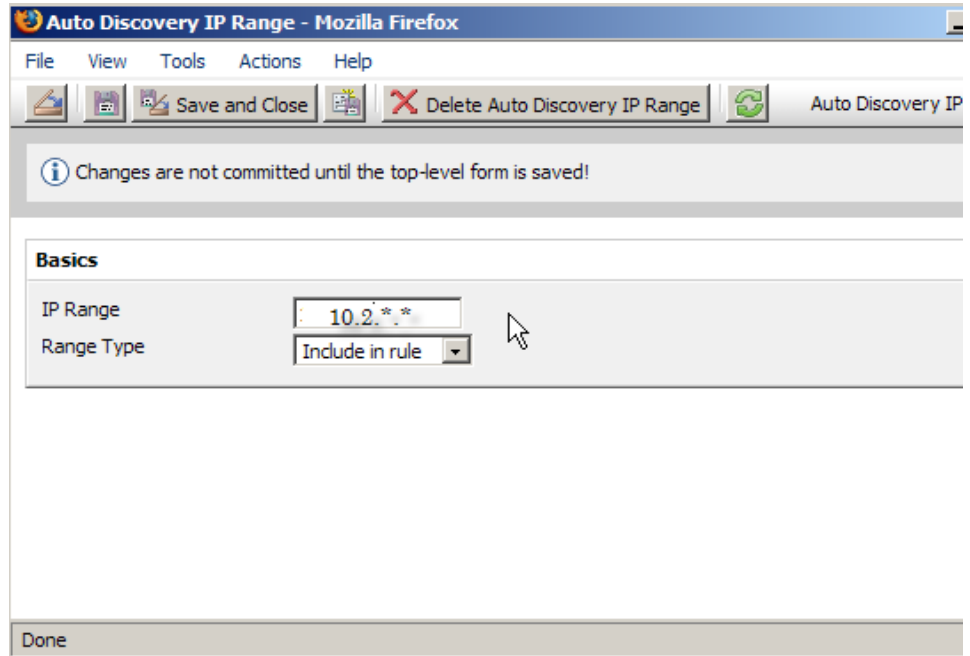


- 4 Enter in the IP range you want to discover. Notice that you can enter both inclusive rules (Include in rule) and exclusive rules (Ignored by rule). The exclusive rules take priority over the inclusive rules.
- 5 Click **Save and Close** for both this form as well as the parent form to save your work.

This example does not use the ping sweep feature. If you do choose to use it in your environment, note that NNMI only sweeps across a maximum of a class B network (like 10.10.*.*) for each discovery rule.

It is important to understand that, by default, NNMI only discovers routers and switches within the defined IP address range. If you want to discover nodes beyond switches and routers, then you should add system object ID ranges that will include your other devices. It is also important to understand that, if a node has multiple addresses, such as a router, only one of the

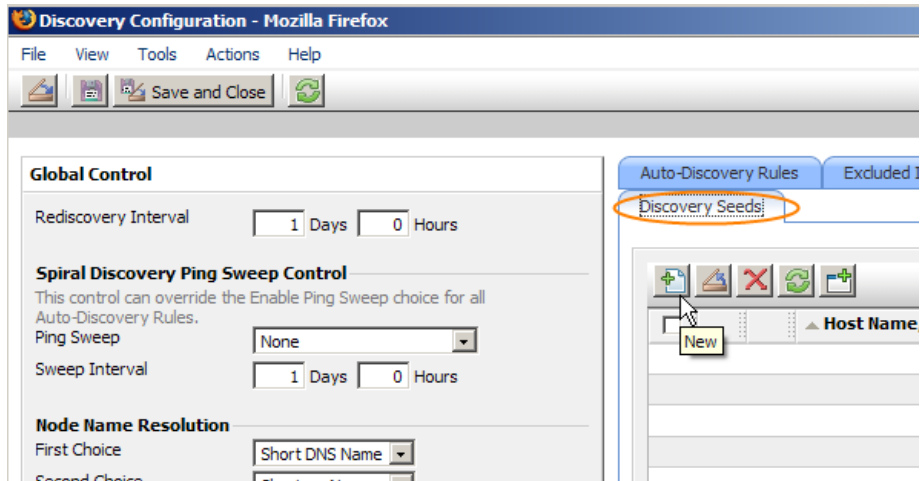
addresses is required to fall within the IP range. This address does not need to be the loopback address. Sometimes NNMi discovers more nodes than you initially expect if you enter addresses other than the loopback addresses.



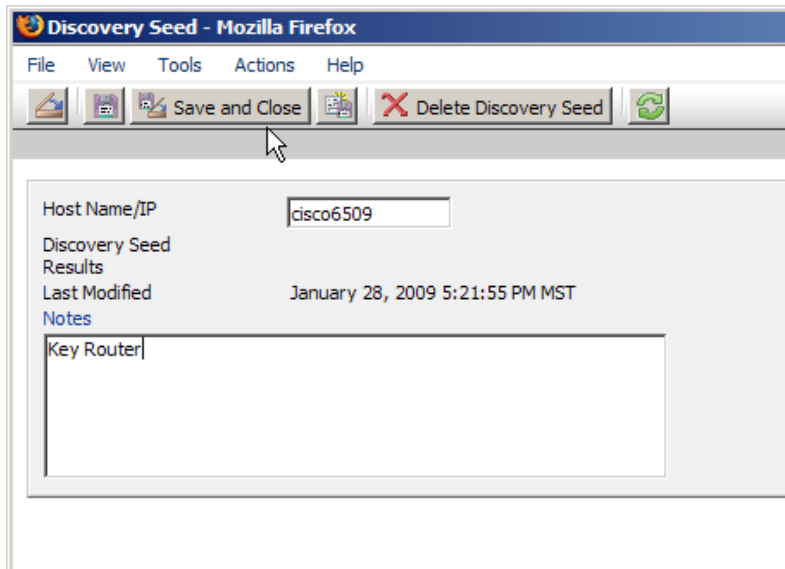
You now have one auto-discovery rule defined. In most cases you only need one rule since each rule can be quite sophisticated.

Next, this example shows you how to add a seed node to get things started. It is better if you to add a router as a seed rather than switch.

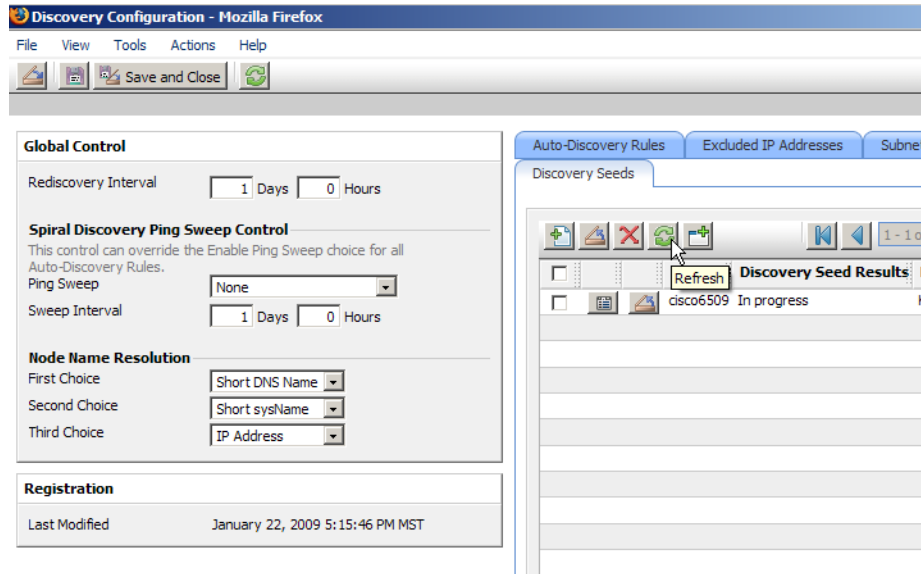
- 1 To begin, click the **Discovery Seeds** tab, then click the **New** icon.



- 2 Enter either the name or the IP address of the seed node into the form, click **Save and Close**.

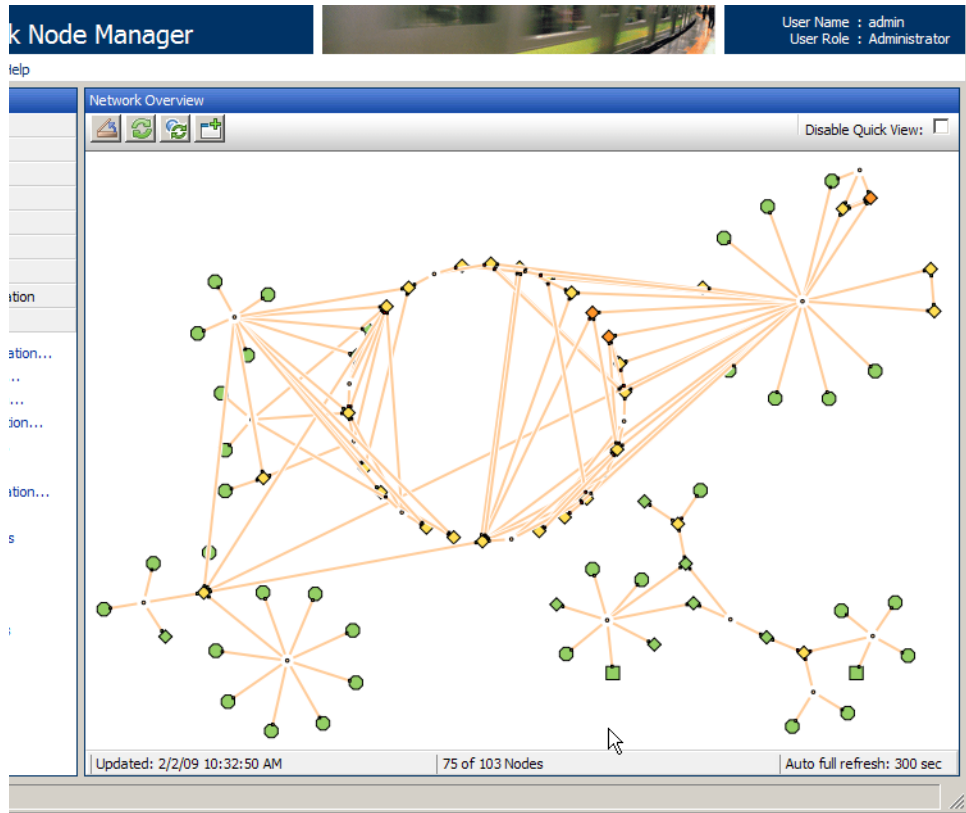


After you click **Save and Close**, NNMi begins discovering the node. NNMi displays the progress as *In progress*. This form does not automatically refresh the status, so click the **Refresh** button when you want a current status. Eventually the *Discovery Seed Results* will change to *Node Created*.



As an alternative to this approach, you can load a list of seeds from a file using the *nnmloadseeds.ovpl* command line tool. This tool enables you to load a large number of seed nodes. If you use list based discovery rather than auto-discovery rules, you can load all of your nodes using the *nnmloadseeds.ovpl* command line tool. See the *nnmloadseeds.ovpl* reference page or the UNIX manpage for more information.

Now auto discovery begins finding other switches and routers that have addresses within the address range. Initially NNMi shows nodes and having no status. Eventually NNMi shows a status for each discovered node. You can click the refresh button on the Network Overview map to have NNMi show the initial nodes. This Network Overview view is more useful in smaller environments, as it works better when displaying a small number of nodes and connections. Further down in this exercise, it shows you how to build some better maps, but for now, use this one knowing it has limitations in this configuration. This view is not a complete overview of your network nodes but is an abbreviated one. You can see in this example that it is only showing 75 of the 103 nodes.

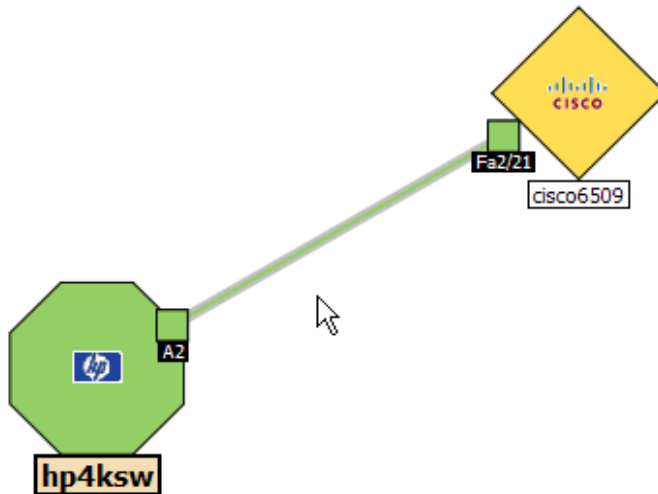


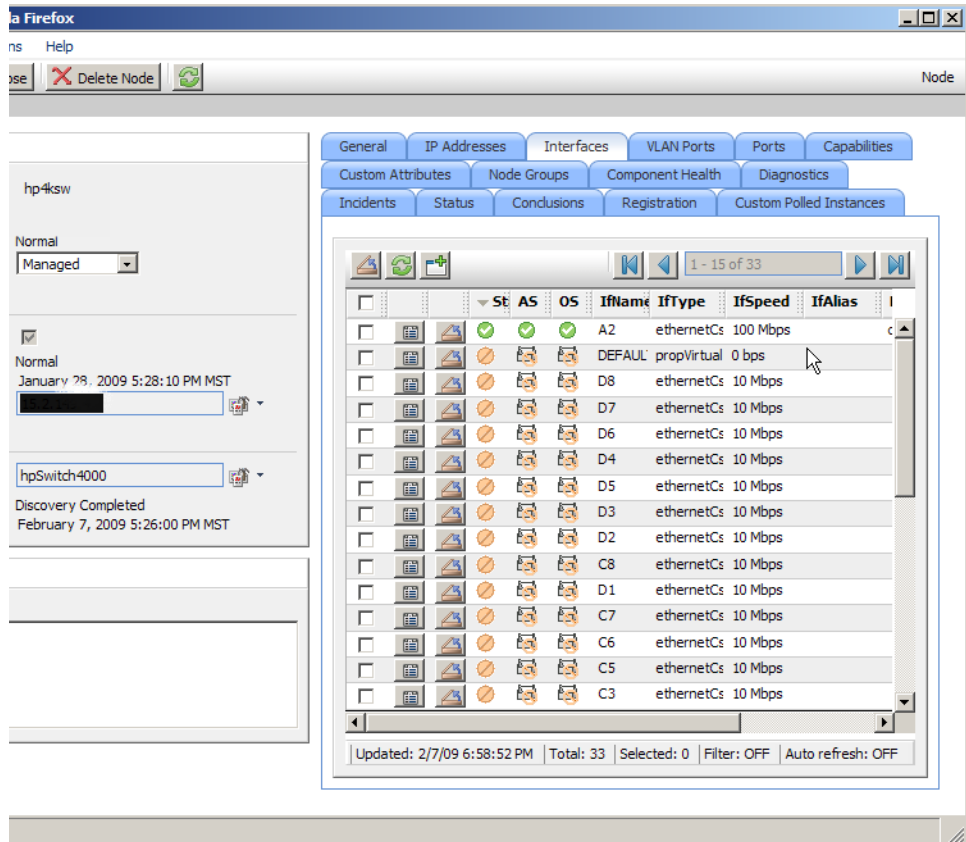
Monitoring Configuration

Monitoring Basics

Monitoring in NNMi is flexible and easy to configure. The first step is to show you the out-of-the-box behavior. By default, NNMi uses SNMP polling rather than ICMP (ping) polling. The exception to this is non-SNMP nodes. NNMi polls these nodes using ICMP. You can enable ICMP polling more broadly if desired.

By default, NNMi polls *connected* interfaces. A connected interface in NNMi is an interface that is connected in the NNMi topology. This does not always map to interfaces that have a wire connected. For example, suppose you have an access switch with 48 ports connected to desktop computers and one uplink port. Suppose you have discovered the uplink node in NNMi but have not discovered any of the desktop computers. In this case, only the uplink port will be considered *connected* to NNMi because it doesn't have a representation of the connection to the desktop computers. In most cases, this is the desired behavior. You usually will not want NNMi to notify you every time somebody shuts off their computer and goes home for the day. An example is shown below. The hp4ksw switch is an access switch with one uplink (A2). You can see in the Node form that only one interface is monitored.





The second default behavior applies specifically to routers. For routers, NNMi monitors most interfaces that host IP addresses. NNMi assumes that if an administrator took the time to configure an IP address on an interface, it is desirable to monitor that interface. In some cases, NNMi models these interfaces as being connected; however, in other cases NNMi models these interfaces as being unconnected. An example of this is a router that has an interface that connects to a WAN cloud. NNMi may not discover and model the connection to the cloud but NNMi monitors the router interface by default.

Modifying the default behavior is simple. You first need to understand the monitoring paradigm used by NNMi. NNMi enables you to modify monitoring settings in high volume. Throughout this document, you see *monitoring setting* and *polling policy* used interchangeably; these terms mean the same

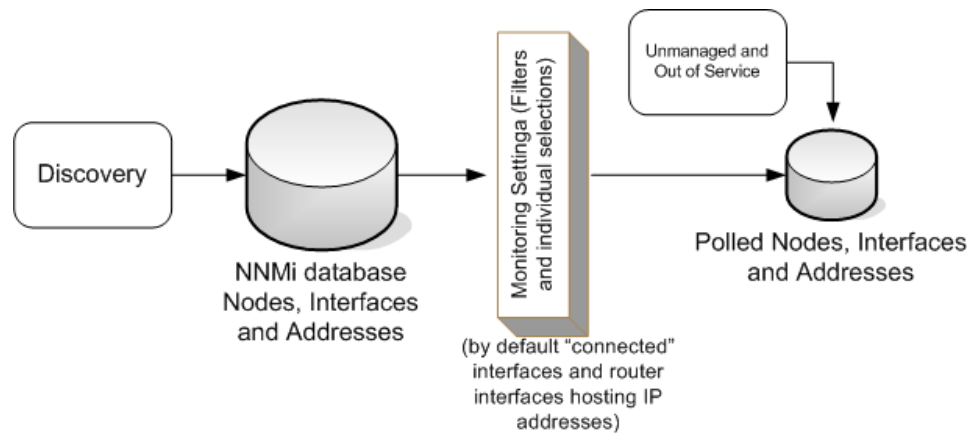
thing. NNMi does this by using filters to apply polling policies to individual nodes, interfaces, and addresses. These filters are the same filters available for the user interface.



NNMi monitors other entities such as Fans, HSRP groups, and others. This document focuses mostly on nodes and interfaces.

Use the following basic steps to modify the monitoring in NNMi:

- 1 Create a node group, an interface group, or both.
- 2 Associate a monitoring setting with the group.
- 3 Prioritize the monitoring setting (nodes and interfaces can match multiple groups).



The following example shows you how to apply these steps to modify the monitoring in NNMi. To create individual exceptions to these policies, set each node or interface you want to change to **Unmanaged** or **Out of Service**.

For this example, suppose that we have interfaces on some nodes with an `IfAlias` that begins with *tunnel to*. Suppose you determine that NNMi needs to monitor these interfaces if their speed is also 9 Kbs. For this example, you create a filter to identify any interfaces that match this criterion. After creating this filter you apply a polling policy to these interfaces.

General IP Addresses Interfaces VLAN Ports Ports Capabilities
 Custom Attributes Node Groups Component Health Diagnostics Incidents
 Status Conclusions Registration Custom Polled Instances

85 - 98 of 151

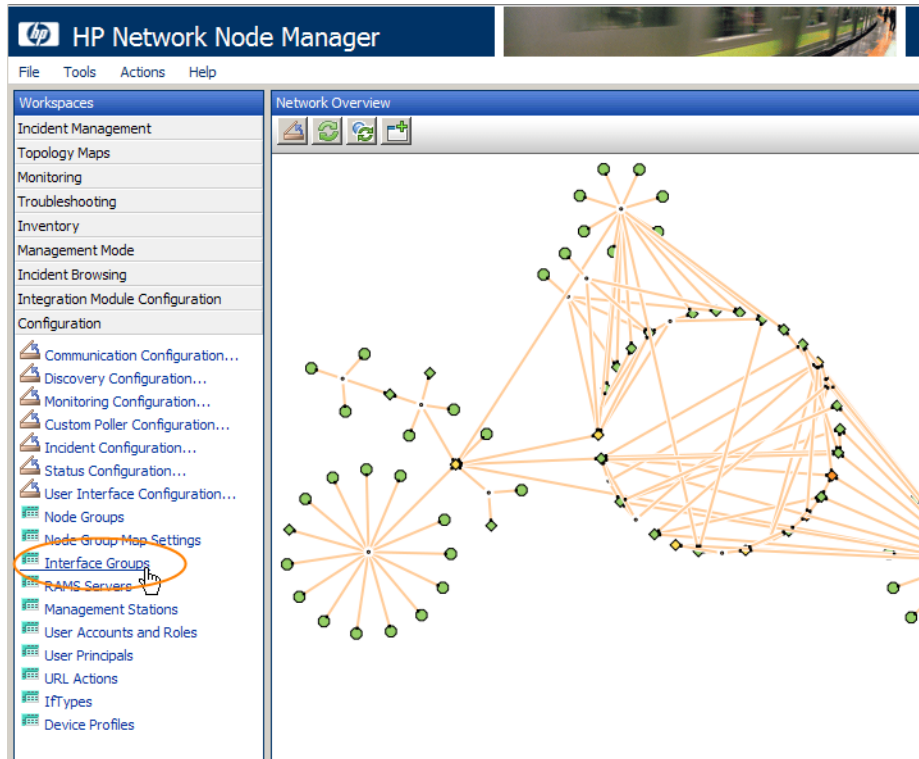
<input type="checkbox"/>	St	AS	OS	IfName	IfType	IfSpeed	IfAlias
<input type="checkbox"/>				Gi4/3	ethernetC	1 Gbps	
<input type="checkbox"/>				Gi5/2	ethernetC	1 Gbps	Gigabit link to ntc6kgw 1 p
<input type="checkbox"/>				EO0/C	propVirtua	100 Mbps	
<input type="checkbox"/>				Nu0	other	10 Gbps	
<input type="checkbox"/>				Tu0	tunnel	9 Kbps	Tunnel 0 for nmmtst15v3
<input type="checkbox"/>				Tu1	tunnel	9 Kbps	Tunnel 1 for padre.cnd.h
<input type="checkbox"/>				Tu2	tunnel	9 Kbps	Tunnel 2 for it2st10.cnd
<input type="checkbox"/>				Tu3	tunnel	9 Kbps	IPv6 tunnel to nautique.c
<input type="checkbox"/>				Tu4	tunnel	9 Kbps	tunnel to ftcgww6-01
<input type="checkbox"/>				Tu5	tunnel	9 Kbps	tunnel to ftcgww6-02
<input type="checkbox"/>				VI2	propVirtua	1 Gbps	
<input type="checkbox"/>				VI10	propVirtua	1 Gbps	
<input type="checkbox"/>				VI20	propVirtua	1 Gbps	
<input type="checkbox"/>				VI511	propVirtua	1 Gbps	.64 IPT S

Updated: 2/7/09 8:20:49 PM | Total: 151 | Selected: 0 | Filter: OFF | Auto refresh: OFF

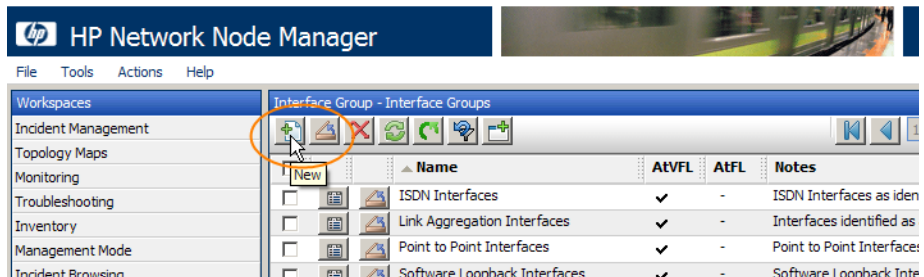
Creating an Interface Group for Monitoring

NNMi provides a sophisticated User Interface to create groups of nodes and interfaces.

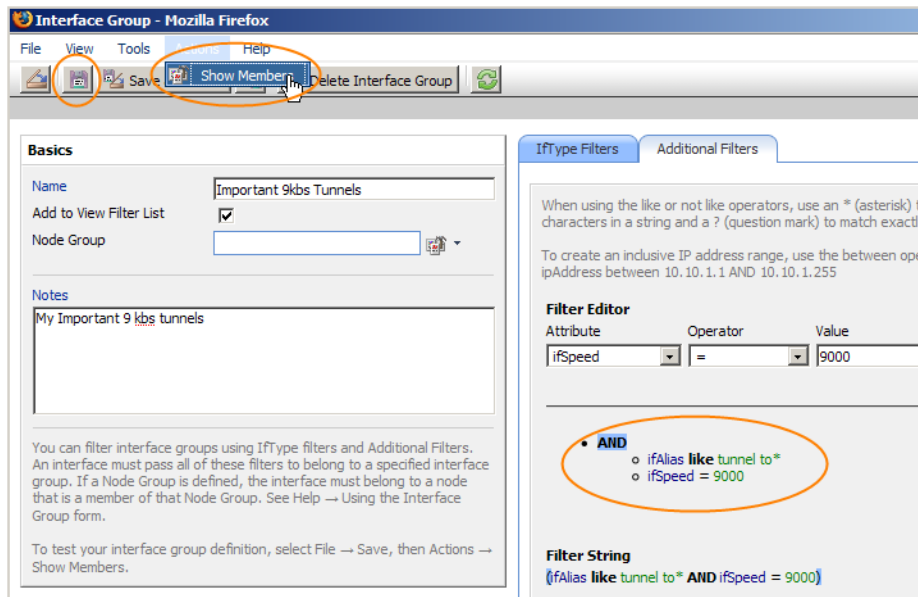
- 1 To begin, click **Configuration** from the NNMi console, then click **Interface Groups**.



- 2 Next, click the **New** icon.



- 3 In this example, title this group by entering **Important 9kbs Tunnels**, or some other descriptive name, in the Name field. Also, do not restrict this group to a specific Node Group, though in many cases you will do this.
- 4 Next, click the **Additional Filters** tab and define the logic. You do this by selecting an Attribute, an Operator and a value. You can use the like operator along with an asterisk for variable matching. You also apply logic to the expressions. In this case, you used an **AND** condition for the two attributes. It can be a bit tricky to use this interface, so you need to practice with it to understand its behavior. If you completely mess up, close the form without saving it to return to the last saved value. Then re-open the form and begin again.
 - ▶ One non-obvious feature is if you define an IFTYPE filter in the first tab, then it is always logically AND'ed with the Additional Filters in the second table.
- 5 After you specify your filter, save the filter but do not close it. After it is saved, verify that it is working as expected.
- 6 Choose the **Actions->Show Members** menu item. This displays a view showing everything that passed this filter.



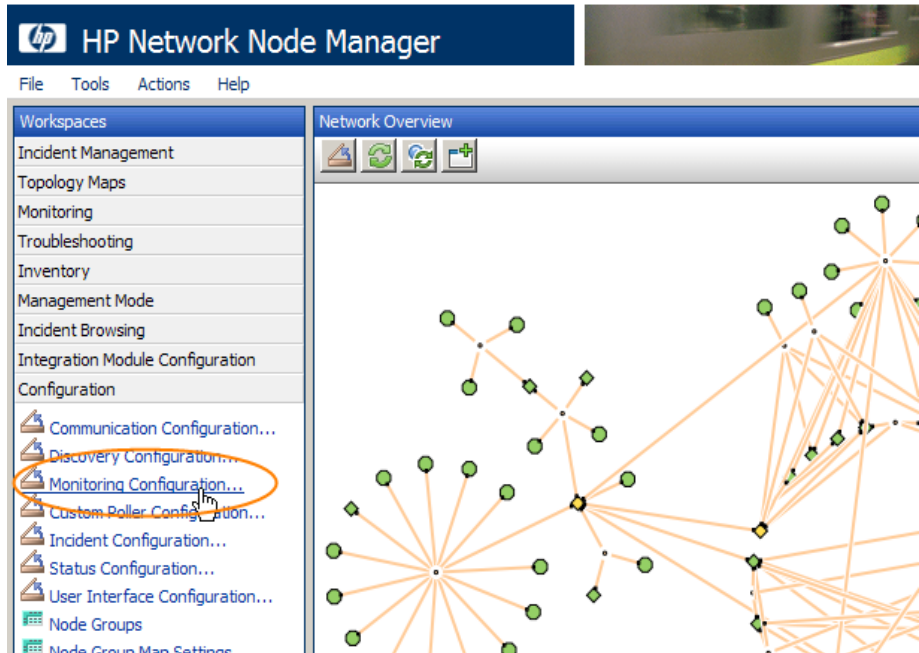
- 7 Verify the results. In this example, you can see that this matched three interfaces in the network. NNMi is already monitoring one of them (probably from the interface already hosting an IP address).
- 8 You can now go back and close the interface group form. Now you can associate a monitoring setting with this group.

<input type="checkbox"/>	Stat	AS	OS	Hosted On Node	IfName	IfType	IfSpeed	IfDescri	IfAlias	St
<input type="checkbox"/>				cisco6509	Tu4	tunnel	9 Kbps	Tunnel4	tunnel to ftcgww	Fe
<input type="checkbox"/>				cisco6509	Tu5	tunnel	9 Kbps	Tunnel5	tunnel to ftcgww	Fe
<input type="checkbox"/>				vwan_router-2	Tu1	tunnel	9 Kbps	Tunnel1	tunnel to dnali	Fe

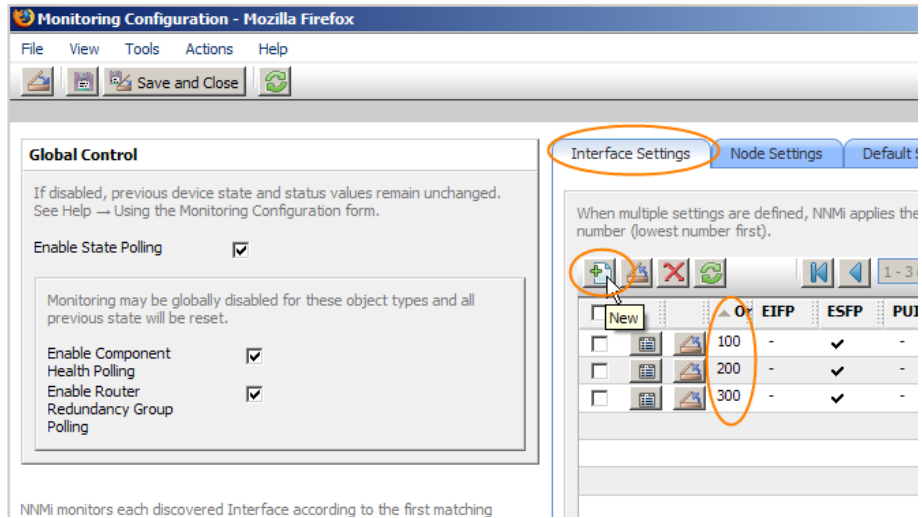
Applying a Polling Policy to an Interface Group

In order to poll the interfaces defined by this filter, you must apply a polling policy to this group. Polling policies can be applied to both node groups and interface groups. NNMi considers an interface setting to be a higher priority than a node setting.

- 1 To begin, click **Configuration** from the NNMi console, then click **Monitoring Configuration**.

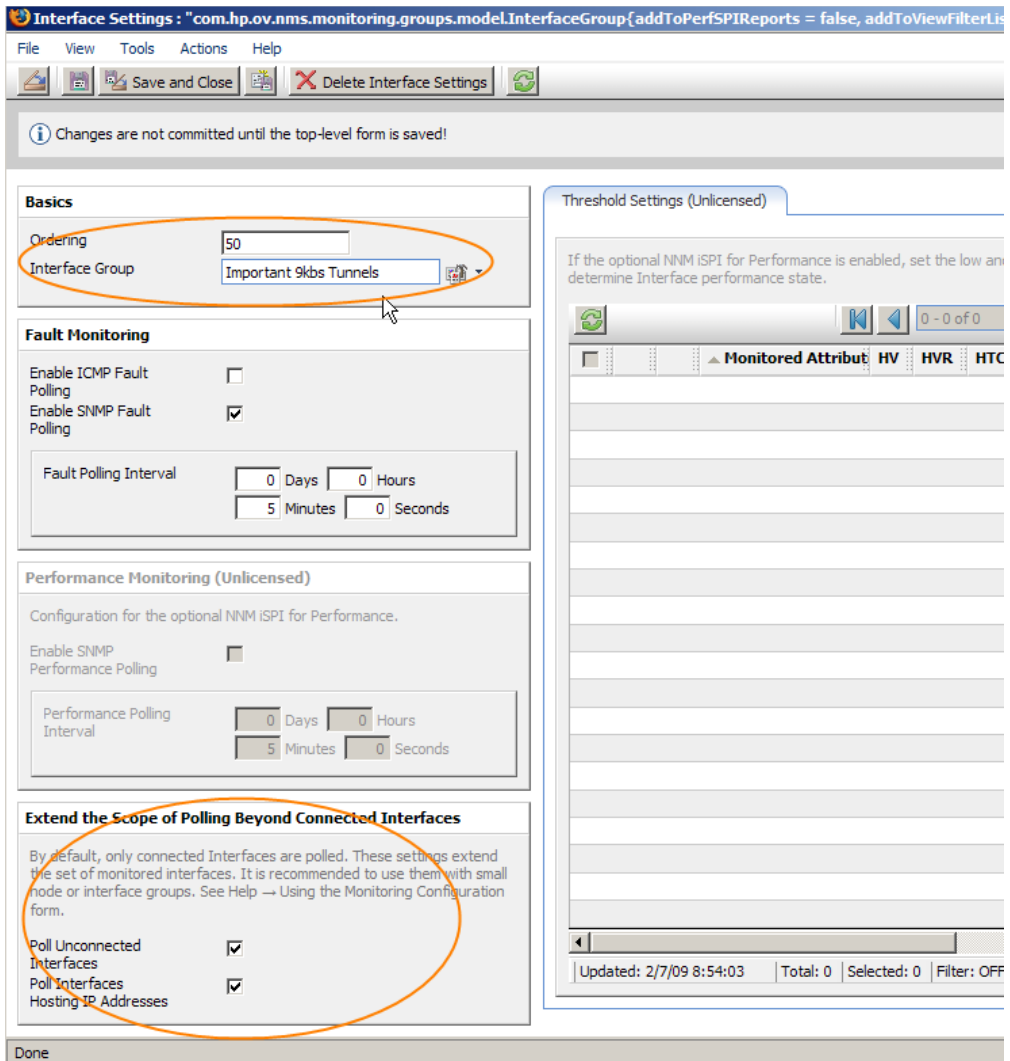


- 2 Since you defined an interface filter, click the **Interface Settings** tab. Take note of the current ordering values. These define priority if an interface were to fall into multiple groups. In this case, the highest priority is currently 100.



- 3 Next, click the **New** icon.
- 4 Now make some important selections.
 - a Choose an Ordering value that configures this setting to have a higher priority than other settings. That will ensure that these interfaces get polled. NNMI considers lower numbers to be higher priority. For this example, choose 50. This leaves you some room for future configuration. For example, if you set this number to 1, that sets the highest priority possible and limits your future entries.
 - b Extend the polling scope. Since you want NNMI to monitor these interfaces regardless of whether they are connected or not connected, click all the boxes.
 - c Use the Quick Find feature to select your newly created Interface Group. Then click **Save and Close**.

- d Click **Save and Close** at the top level Monitoring Configuration Form or this change will not become active.



Congratulations, you just applied a monitoring setting to everything in this group. NNMi now uses SNMP to poll any interface that matches the Important 9kbs Tunnels filter.

Testing the Polling Policy

You can test your new polling policy in many different ways. For this example, click **Inventory** from the NNMi console, then click **Interfaces**. Use the pull-down menu to select our new interface group, **Important 9kbs Tunnels**. This filters the table to only show the interfaces that interest you. Notice that some of the interfaces are still in an unpolled state. It can take a few minutes for these changes to flow through the system. To speed this up, perform a status poll command on one of the nodes hosting these interfaces. You should see them all begin to acquire status.



To perform a status poll on a node, click **Inventory** from the NNMi console, then click **Nodes**. Select the node you want to poll, then use the **Actions > Status Poll** command to start the status poll.

The screenshot shows the HP Network Node Manager interface. The left sidebar has 'Inventory' > 'Interfaces' selected. The main window displays a table of interfaces under the group 'Important 9kbs Tunnels (Interfaces)'. The table has columns for 'Stat', 'AS', 'OS', 'Hosted On Node', 'IfInan', 'IfType', 'IfSpeed', 'IfDescri', and 'IfAlias'. Three interfaces are listed, all with a 'Stat' of 'OK'.

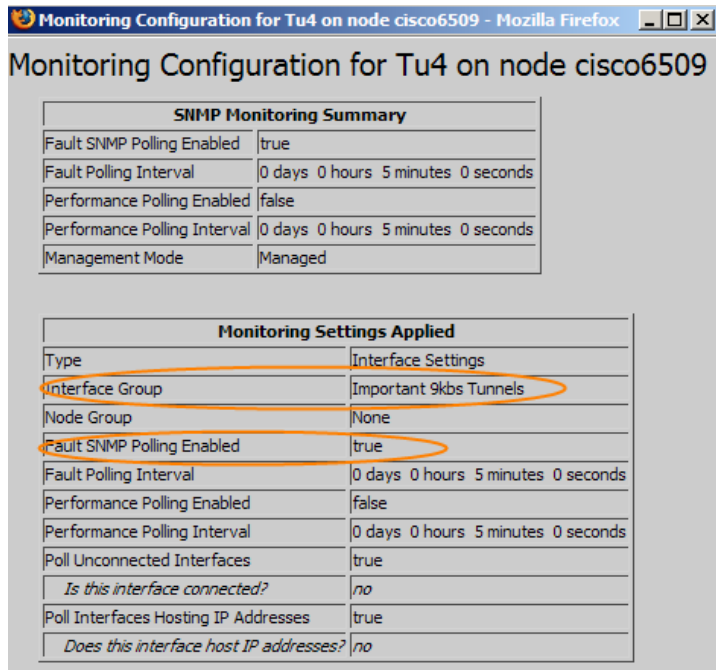
Stat	AS	OS	Hosted On Node	IfInan	IfType	IfSpeed	IfDescri	IfAlias
OK	✓	✓	cisco6509	Tu4	tunnel	9 Kbps	Tunnel4	tunnel to
OK	✓	✓	cisco6509	Tu5	tunnel	9 Kbps	Tunnel5	tunnel to
OK	✓	✓	vwan_router-2	Tu1	tunnel	9 Kbps	Tunnel1	tunnel to

The final and best way to confirm that your new polling policy worked is to open up one of these interfaces and check the monitoring settings. To do this, do the following:

- 1 Select one of the interfaces.
- 2 Click the **Open** icon in the table to open the interface form.
- 3 Use the **Actions > Monitoring Settings** menu item to view the monitoring settings.

The screenshot shows the 'Actions' menu open in the HP Network Node Manager. The 'Monitoring Settings' option is highlighted with a blue selection bar and a mouse cursor. Other options in the menu include 'Layer 2 Neighbor View', 'Layer 3 Neighbor View', 'Node Group Map', 'Path View', 'Manage', 'Manage (Reset All)', and 'Unmanage'.

Now NNMi displays a form with some important information. First, you can see that NNMi applied the monitoring settings for the Important 9kbs Tunnels group to this interface. This shows you that the polling policy is properly associated with this interface. Second, you can see that NNMi has Fault SNMP Polling Enabled set to true. You now know with confidence that you successfully applied your new monitoring setting to the Important 9kbs Tunnels group.



SNMP Monitoring Summary

Fault SNMP Polling Enabled	true
Fault Polling Interval	0 days 0 hours 5 minutes 0 seconds
Performance Polling Enabled	false
Performance Polling Interval	0 days 0 hours 5 minutes 0 seconds
Management Mode	Managed

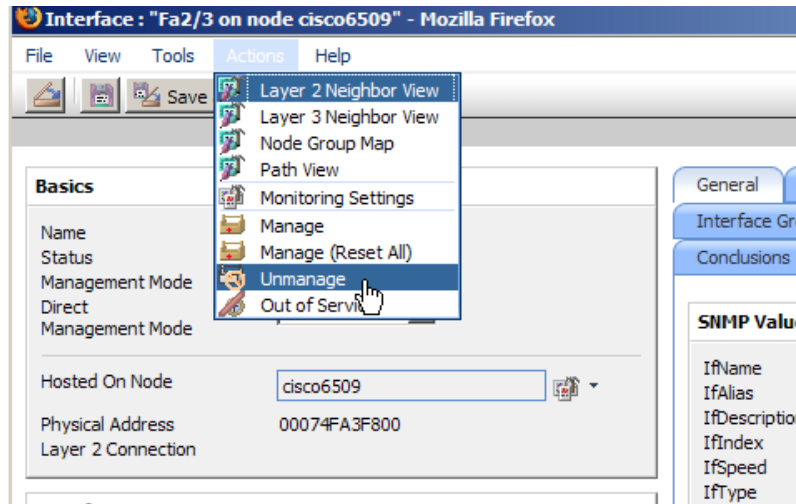
Monitoring Settings Applied

Type	Interface Settings
Interface Group	Important 9kbs Tunnels
Node Group	None
Fault SNMP Polling Enabled	true
Fault Polling Interval	0 days 0 hours 5 minutes 0 seconds
Performance Polling Enabled	false
Performance Polling Interval	0 days 0 hours 5 minutes 0 seconds
Poll Unconnected Interfaces	true
<i>Is this interface connected?</i>	<i>no</i>
Poll Interfaces Hosting IP Addresses	true
<i>Does this interface host IP addresses?</i>	<i>no</i>

There will be times when you are not sure why a particular interface or node is being polled. Use the Monitoring Settings menu item to help you diagnose this. You may have the ordering values set up so that NNMi is applying a different monitoring setting to your interface or node.

Making Exceptions to Monitoring

You can always force an interface or node to be unmonitored manually. From the Interface Form, use the Actions->Unmanage menu item to change to unmanage the interface. NNMi will no longer monitor this interface regardless of what the monitoring settings are set to.



NNMi does not presently have the same easy approach that NNM used to force an interface to be unmonitored. Unmanaging an interface is only a *negative override*. HP released a white paper, *How to Force an Interface to be Monitored*, that shows how to force NNMi to monitor an interface. The method in the white paper is not difficult to use, but requires a few more steps than simply unmanaging the node or interface.

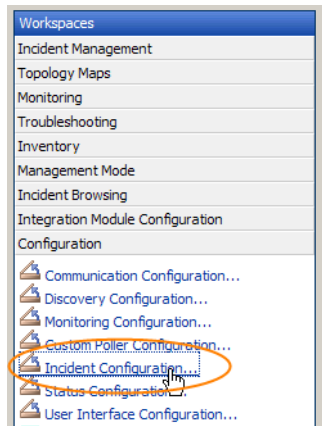
Configuring Incidents, Traps and Automatic Actions

Configuring Incidents

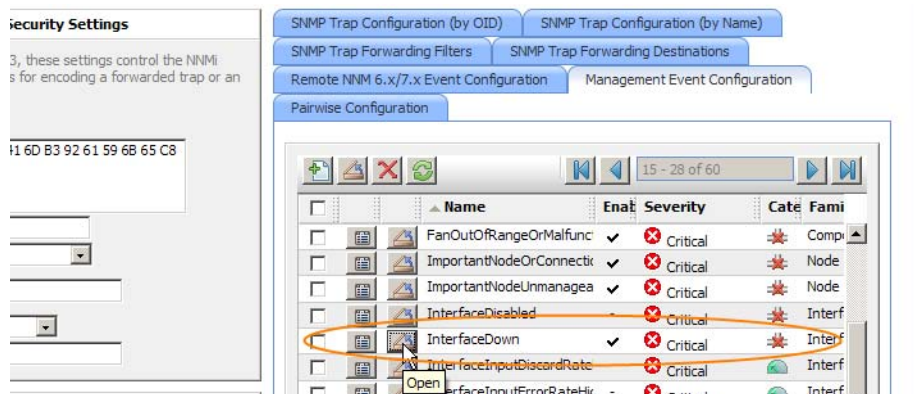
With NNMi, you can change various aspects of an incident. Some examples include enabling an incident, formatting a message, enabling de-duplication and enabling rate correlation.

For our example, suppose you want to enhance the `InterfaceDown` (Interface Down) incident to include the `Interface Alias` in the message.

- 1 To begin, click **Configuration** from the NNMi console, then click **Incident Configuration**.



- 2 Click the **Management Event Configuration** tab, then open the `InterfaceDown` incident.



- Before continuing, you can view the possible arguments that can be added to a message format in the NNMi help. See *Valid Parameters for Configuring Incident Messages* in the NNMi help. In this example, add the argument `$ifAlias` to the incident message as shown in the highlighted selection below.

Management Event Configuration : "InterfaceDown" - Mozilla Firefox

File View Tools Actions Help

Save and Close Delete Management Event Configuration

Changes are not comm Save and Close Level form is saved!

Basics

Name: InterfaceDown
 Enable:
 Category: Fault
 Family: Interface
 Severity: Critical

Message Format
 Interface Down with Alias = \$ifAlias

Deduplication Configuration Rate Configuration Acti

Configuration Per Node Group

Enable:
 Count: 2
 Hours: 0
 Minutes: 0
 Seconds: 0
 Correlation Incident Confn

- Finally, click the **Save and Close** icon on this form and in the outer form as well. Now all `InterfaceDown` incidents show the `$ifAlias` parameter. If there is no alias on the interface, it will show null for the alias.

If you look for new incidents that arrive in the browser, you'll see the new message format.

irrence	AT	Source Node	Source Objec	Cate	Fam	Orig	Message	Notes
3:02 PM		ntcgw2loop	Chassis Fan Tre				Fan on ntcgw2loop is malfunctioning	
54 AM		mplsce05	Fa0/1				Interface Down with Alias = null	
37 AM		core_6509-1	Fa9/48				Interface Down with Alias = connection to run server	
34 AM		cisco6509	Fa2/14				Interface Down with Alias = Connection to NetScout probe sv	
34 AM		mplsce04	Fa0/1				Interface Down with Alias = Connect to Customer #2 Site #1	
34 AM		cisco6509	Fa2/3				Interface Down with Alias = connect to PIX520-1 outside	
51 PM		ntcgw1loop	Gi4/4				Interface Down with Alias = Link to workstation ip 15.2.133.5	
49 PM		ntcgw1loop	Vi16				Interface Down with Alias = Link to end-node ntdht15 Lan0	
49 PM		ntcgw1loop	Gi4/30				Interface Down with Alias = null	
38 AM		internet_switch-5	Fa0/21				Interface Down	

Configuring Traps

HP released a white paper containing more details about working with traps in NNMi. See the *Step-by-Step Guide to Managing SNMP Traps in NNMi* for more complete details.

For this example, suppose you have some RuggedCom equipment that sends traps to NNMi. In order to receive a trap into the NNMi Incident Browser, you must load the MIB that contains the trap definitions into NNMi.

For this example, you need to load three MIBs to satisfy the dependencies. You first load the `ruggedcom.mib` file followed by the `rcsysinfo.mib` file. This enables you to load the traps from the `ruggedcomtraps.mib` file. Notice the two different arguments used (`-loadMib` and `-loadTraps`). The `-loadMib` argument loads the MIB definitions that NNMi requires so it can load the traps using the `-loadTraps` argument.

Begin by loading these MIBs into NNMi by using the `nnmincidentcfg.ovpl` command.

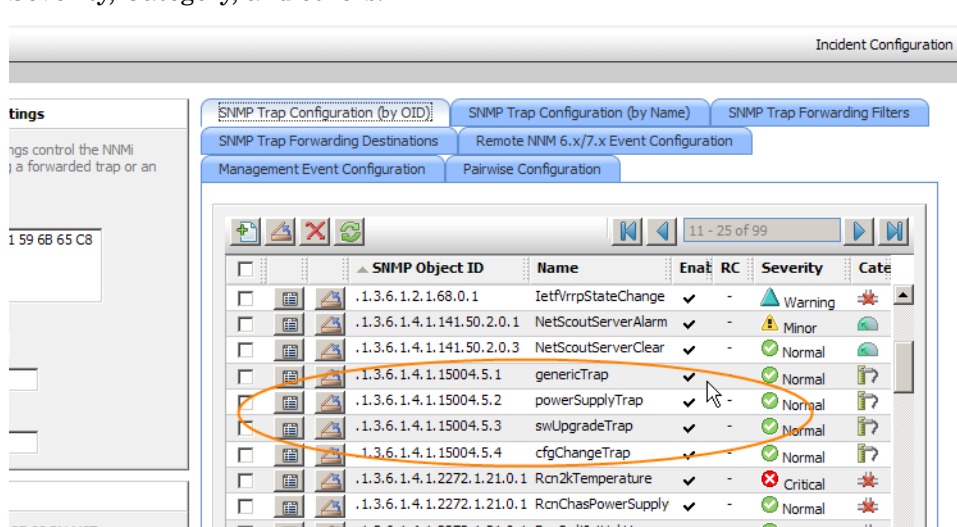
- 1 Run the `nnmincidentcfg.ovpl -u admin -p adminpw -loadMib ./ruggedcom.mib` command. This loads the `/var/tmp/mibs/./ruggedcom.mib` file.
- 2 Run the `nnmincidentcfg.ovpl -u admin -p adminpw -loadMib ./rcsysinfo.mib` command. This loads the `/var/tmp/mibs/./rcsysinfo.mib` file.
- 3 Run the `nnmincidentcfg.ovpl -u admin -p adminpw -loadTraps ./ruggedcomtraps.mib` file.

You will see a display similar to the following:

```
Number of traps: 4.
The following traps were added to incident configuration:
cfgChangeTrap - .1.3.6.1.4.1.15004.5.4
swUpgradeTrap - .1.3.6.1.4.1.15004.5.3
powerSupplyTrap - .1.3.6.1.4.1.15004.5.2
genericTrap - .1.3.6.1.4.1.15004.5.1
```

You now have four new traps defined in NNMi. To see these new traps, click **Configuration** from the NNMi console, click **Incident Configuration** then click either the **SNMP Trap Configuration (by OID)** tab or the **SNMP Trap Configuration (by Name)** tab depending on what is the easiest way for you to identify the traps. In this case, this vendor did not use the more standard convention of prefixing each name with an easily identifiable set of letters, however we can

easily see the four traps based on the OID. Notice that all of the traps are loaded as *enabled*. You may want to disable all but the ones you specifically want to receive. You can also modify other fields within the incident form like Severity, Category, and others.



Configuring Automatic Actions

Another common task you can do is to add automatic actions to incidents. Usually you only do this for management events rather than for SNMP traps as it is hard to predict the rate and volume of traps. NNMi automatic actions can be executable commands, command line scripts, or Python scripts. The Python scripts execute within NNMi's JVM so they execute quickly. Since NNMi uses a Java interpreter for Python, NNMi calls these scripts `Jython` in the forms. We encourage you to give Python a try as it is a language that executes quickly though using it may require some additional learning.

In NNMi, actions are based on lifecycle state changes for incidents. You could configure NNMi to take one action when an interface goes down and another action when the interface comes back up again. To do this, configure both actions on the `InterfaceDown` incident, but associate one action with the `Lifecycle State` set to `Registered` and the other action with the `Lifecycle State` set to `Closed`. There usually will not be an associated `up` incident.

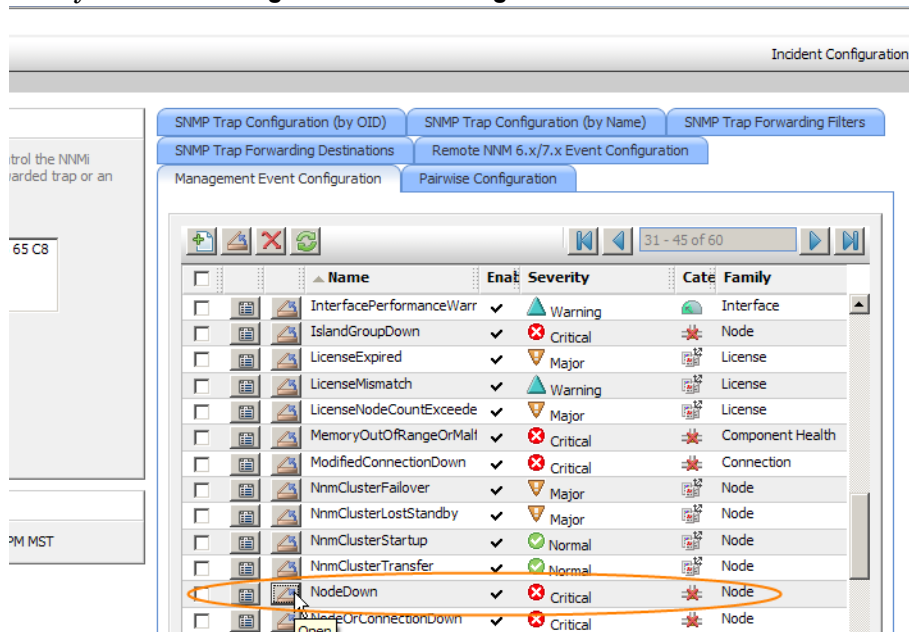
Suppose that you develop a Perl script that you want executed each time NNMi generates a NodeDown incident. When NNMi generates an incident, it assigns the Registered state to the incident. This is similar to what you might consider to be an *open* state. Do the following to configure NNMi to run your script to when it received a NodeDown incident:

- 1 To begin configuring an incident to run your Perl script, you must place your script into the actions directory. For security reasons, you must be root or administrator to access this directory. For this example, assume the actions directory appears in the following location:

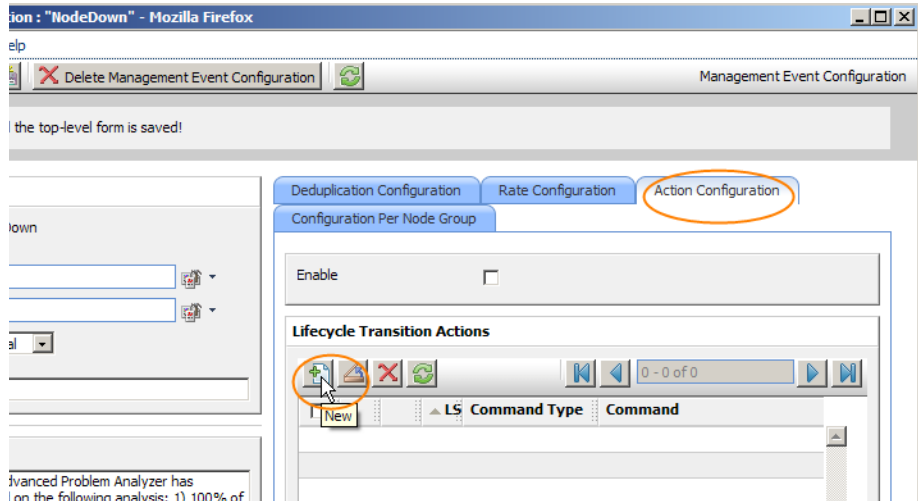
- **Windows:** \Documents and Settings\All Users\Application Data\HP\HP BTO Software\shared\nnm\actions
- **UNIX:** /var/opt/OV/shared/nnm/actions

The actions directory can be in a different location depending on how you installed NNMi. For this example, suppose you called your script `writelog.ovpl`. Copy this script into the actions directory. Make sure that your script is executable.

- 2 Now you need to associate this script with an action on this incident. First, click **Configuration** from the NNMi console, then click **Incident Configuration**, finally click the **Management Event Configuration** tab.

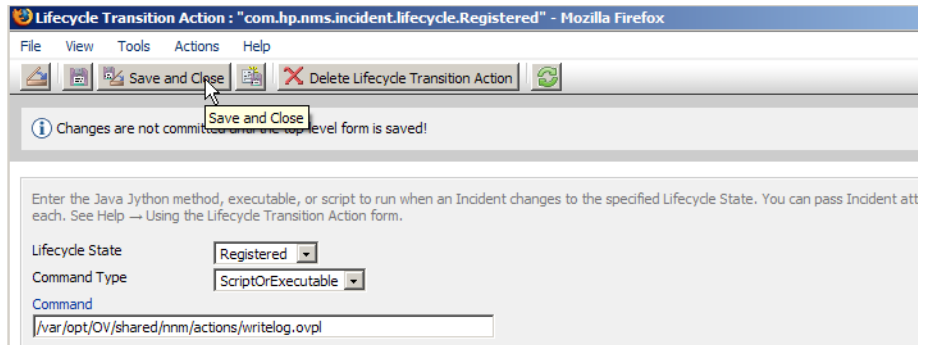


- 3 Select the NodeDown Incident, then click the **Open** icon.
- 4 Click the **Action Configuration** tab, then click the **New** icon.

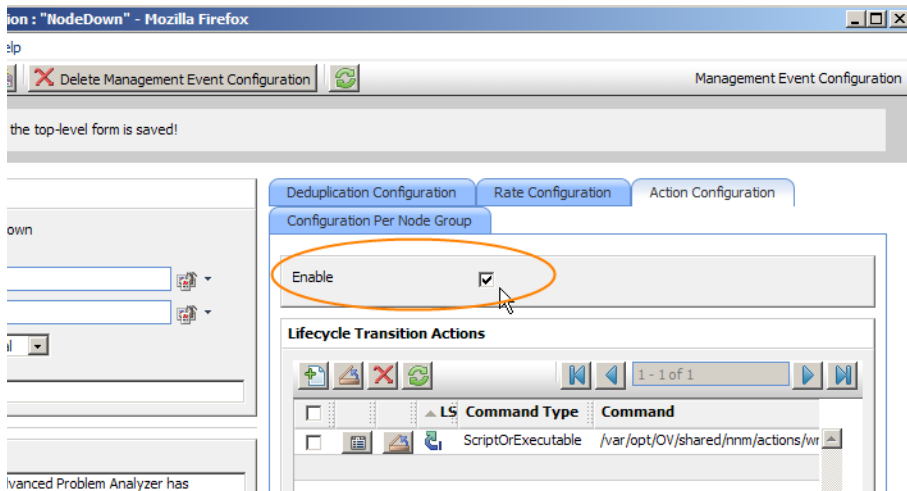


- 5 Select the appropriate lifecycle state (Registered in this example).
- 6 Set the Command Type to ScriptOrExecutable.
- 7 Enter the name of the command, including the complete path to the executable, then click **Save and Close**.

► You can pass several arguments to the script. See the NNMi help for a complete list of possibilities.



- 8 Finally, you need to select the `Enable` box to enable the action.

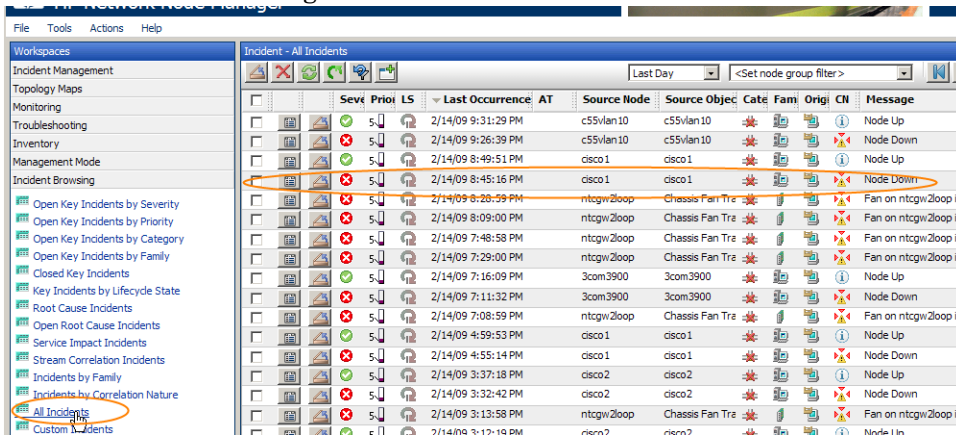


- 9 Click **Save and Close** on this form and on the outer form as well. If you do not save all the way to the outer form, NNMi will not properly set up the action.

Now you need to test the action. The easiest way to do this is to look for a previous occurrence of the `NodeDown` incident:

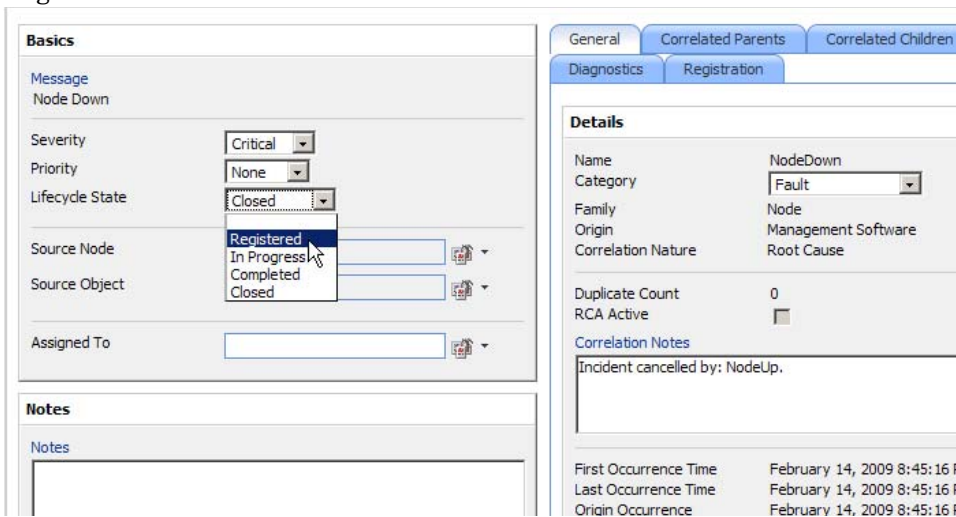
- 1 Click **Incident Browsing** from the NNMi console, then click **All Incidents**.
- 2 Open a `NodeDown` incident that NNMi closed. In this example `Closed` means that the interface is back up. NNMi automatically closes an incident when a fault is cleared. By working with a closed incident, you can re-open the incident by setting the `Lifecycle State` to `Registered`.

After you take this action, NNMi behaves as if the incident is opened for the first time when executing actions.



Practice running this action by setting the Lifecycle State back to Registered. This causes your action to execute after you save this form (saving the Lifecycle State change). If you change the Lifecycle State without saving the change, NNMi takes no action. You must click the **Save** button after each Lifecycle State change.

After saving your change, verify that your action ran as expected. In this case, look at the log file that this script wrote to. After you finish testing, set the Lifecycle State back to Closed, then save the incident to return it to its original state.

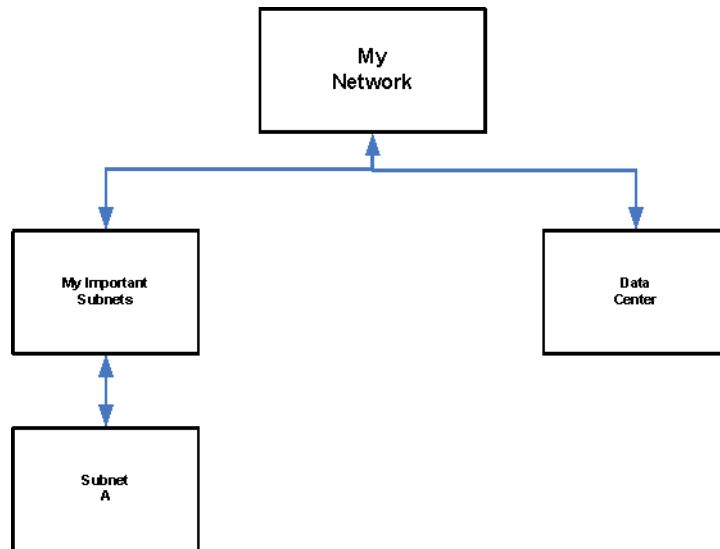


Configuring the NNMi Console

Node Group Configuration

To enhance diagnostics, create container maps which will show the nodes that a node group contains. HP released a white paper containing more details about working with node and node group maps in NNMi. See the *Step-by-Step Guide to Configuring Node Groups and Node Group Maps* for more complete details. The following explanation shows you a slightly different example than the one contained in the white paper.

Suppose you need to create some logical containers for a few different subnets. You want these containers to refer to management addresses rather than any address on the node. You also want these containers to contain nodes based on names. Nodes can be in multiple node groups. Suppose that you need to create the following hierarchy of groups:



Subnet A = Management Address of 192.168.*.*

Data Center = nodes that have a system name beginning with “data_center”

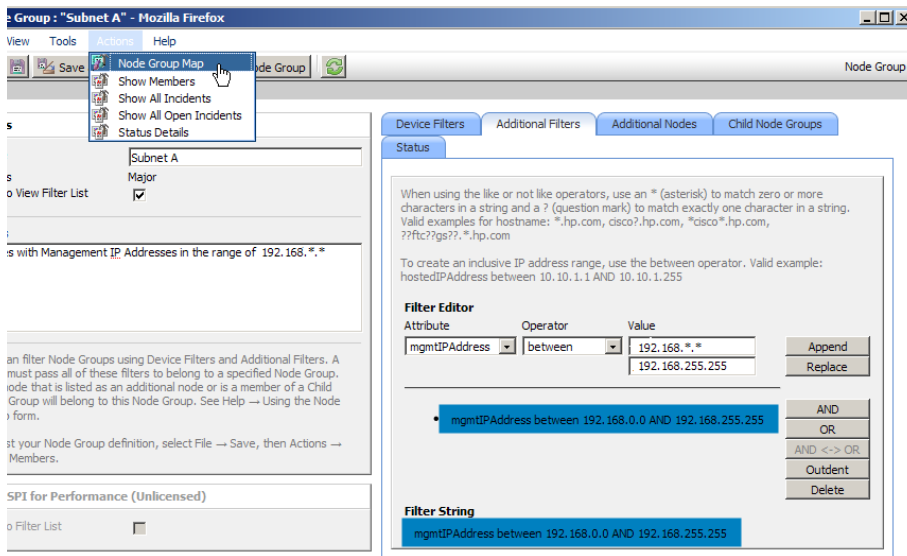
Begin by creating node groups for each of these containers. Note that only the leaf groups will be populated with nodes. The other containers should only show structure in the hierarchy and will only be populated with a child node group. It is easiest to begin from the leaf node groups and work your way up the hierarchy. Look below for a few examples. Notice the unique expression for IP Address ranges. Remember to test the membership after you save each Node Group with the **Actions > Show Members** menu item.

The screenshot shows the configuration for a Node Group named "Subnet A". The "Basics" tab is active, showing the name "Subnet A" and a note: "Nodes with Management IP Addresses in the range of 192.168.*.*". The "Additional Filters" tab is also active, showing a filter editor with the attribute "mgmtIPAddress", operator "between", and value "192.168.*.*". The filter string is "mgmtIPAddress between 192.168.0.0 AND 192.168.255.255".

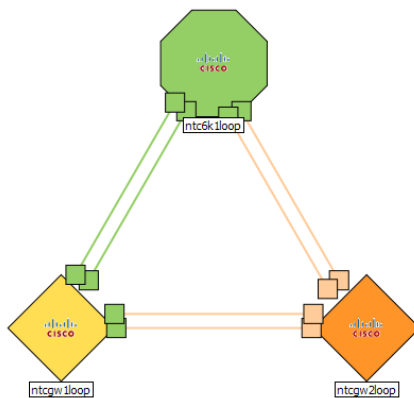
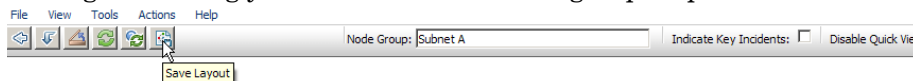
The screenshot shows the configuration for a Node Group named "Data Center". The "Basics" tab is active, showing the name "Data Center" and a note: "Nodes with a system name beginning with data_center". The "Additional Filters" tab is also active, showing a filter editor with the attribute "sysName", operator "like", and value "data_center*". The filter string is "sysName like data_center*".

Since you are going to place these node groups onto a map, after you test the population of the node groups you must take some steps to create to create an initial instance of a map for each group.

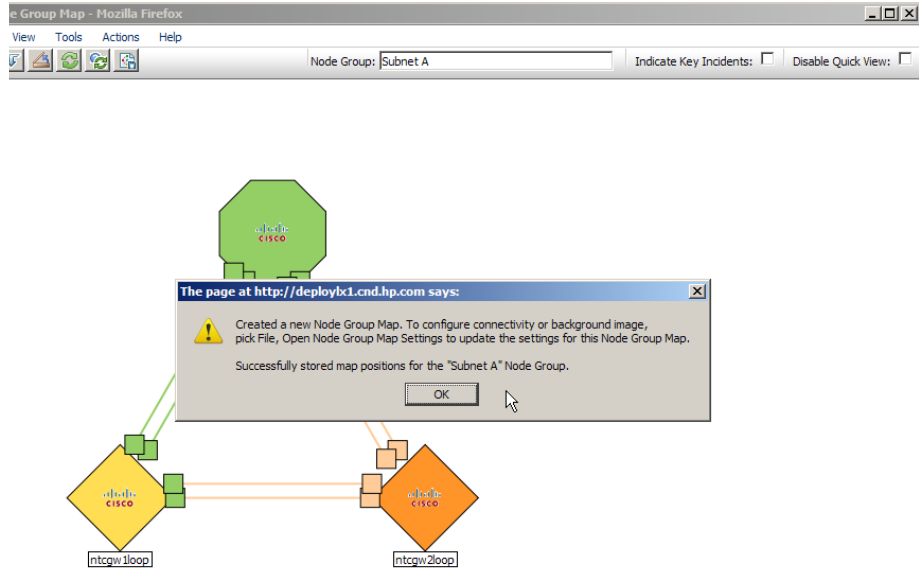
- 1 First select Actions->Node Group Map to open the map.



- 2 Then click **Save Layout**. After you save the change, NNMi displays a message informing you that it created a node group map.

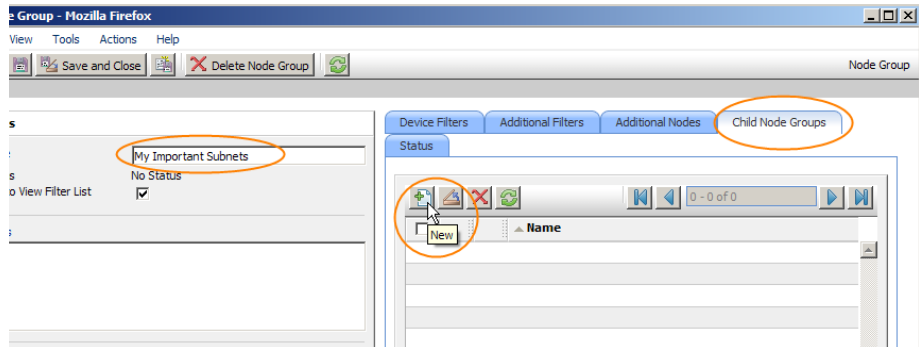


3 Click OK.

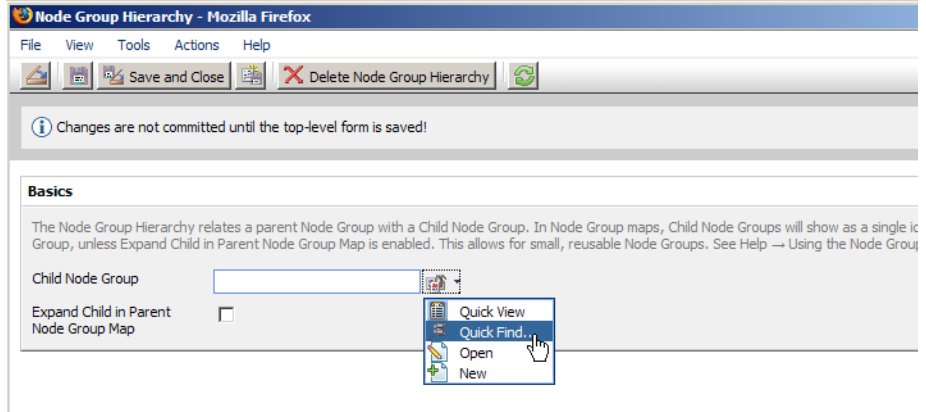


For the structure Node Groups, they don't need any filters but require Child Node Groups to define the hierarchy. For example, create the Node Group called *My Important Subnets* as follows:

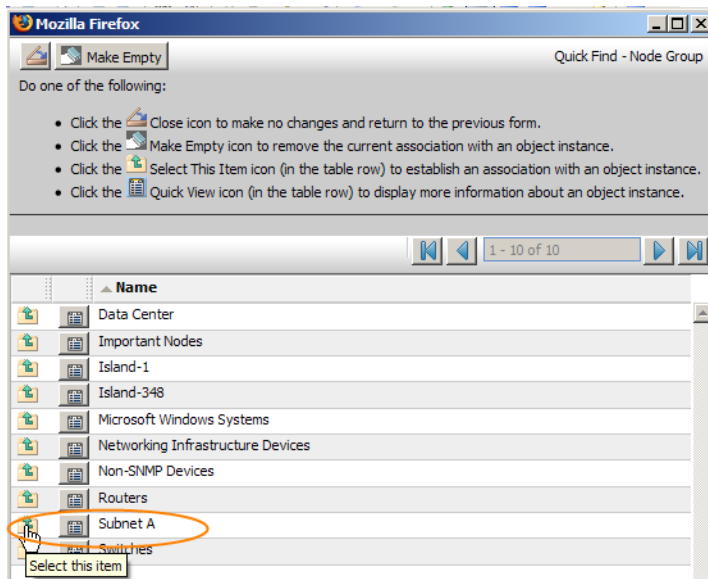
1 Click the **Child Node Groups** tab, then click **New**.



- 2 Now pull down the arrow to select `Quick Find`. Always use this method to select an existing object.

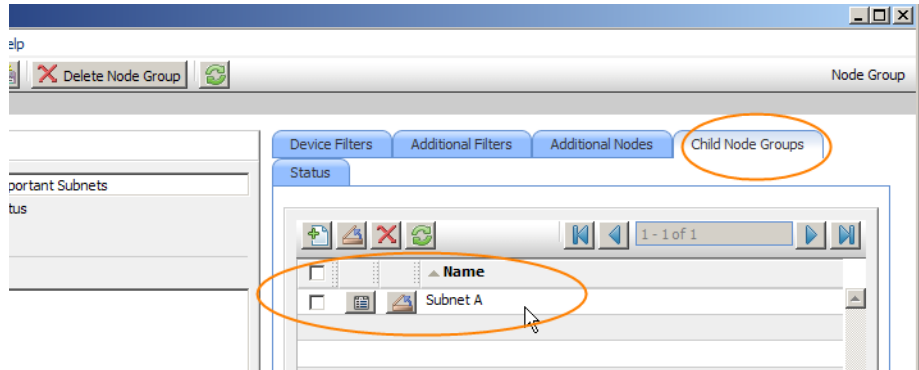


- 3 Select the child node group. That is *Subnet A* in this example.



- 4 Click **Save and Close**. You just created a child node group, *Subnet A*, for the *My Important Subnets* node group.

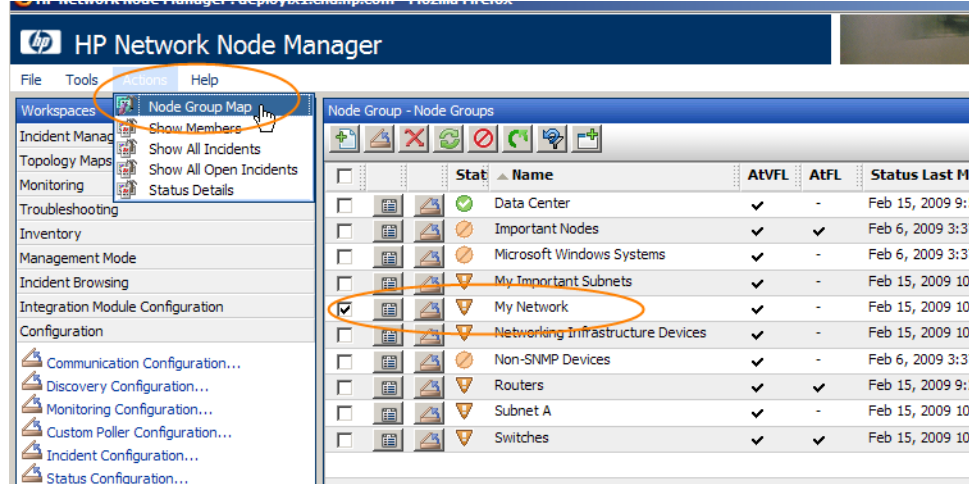
5 Be sure to save this form as well.



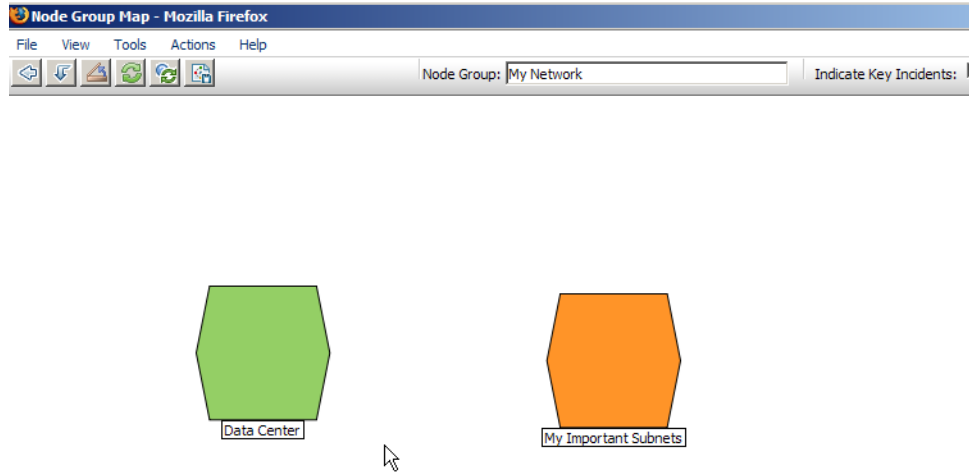
Repeat this same process for the entire hierarchy. It may take time for status to fully propagate to the node groups.

Configuring the Node Group Map

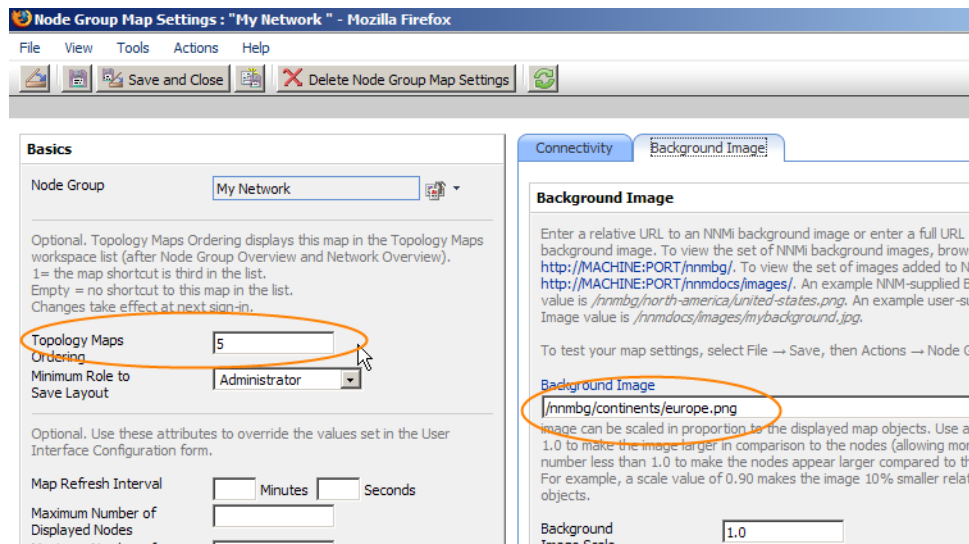
You now have a map hierarchy that you drill into and back out. In this example, you can open the node group map for the node group My Network.



From this map, you can drill down (double-click) and back with the arrows.

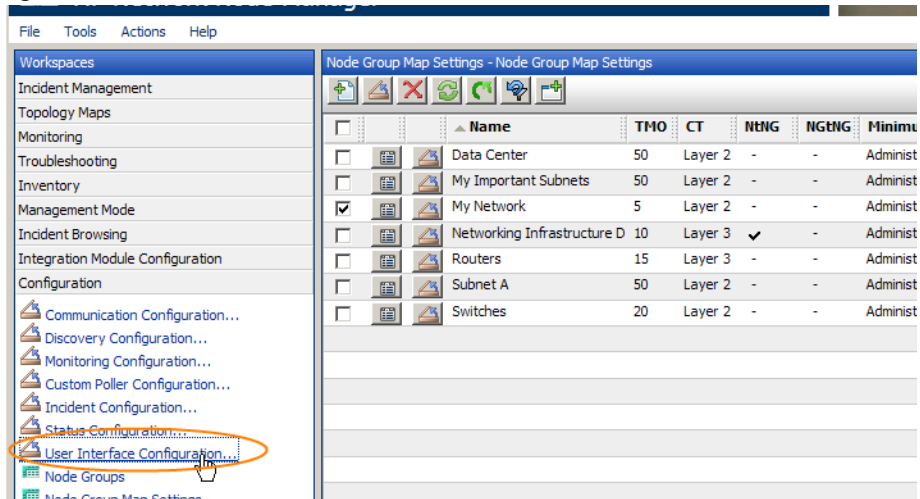


In addition to positioning, you can also add background graphics, change connectivity options, and many other options. See the *Step-by-Step Guide to Configuring Node Groups and Node Group Maps* for more complete details. For this example, change the Topology Maps Ordering (so we can make it our default map) then put a background graphic on the map as shown in the next screen shot. Make sure to click **Save and Close** to save your work.

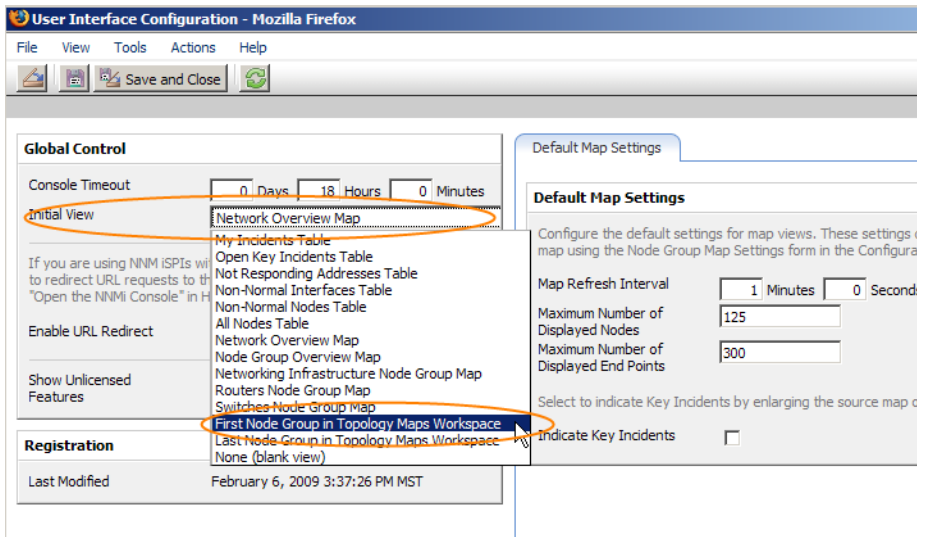


Configuring the User Interface

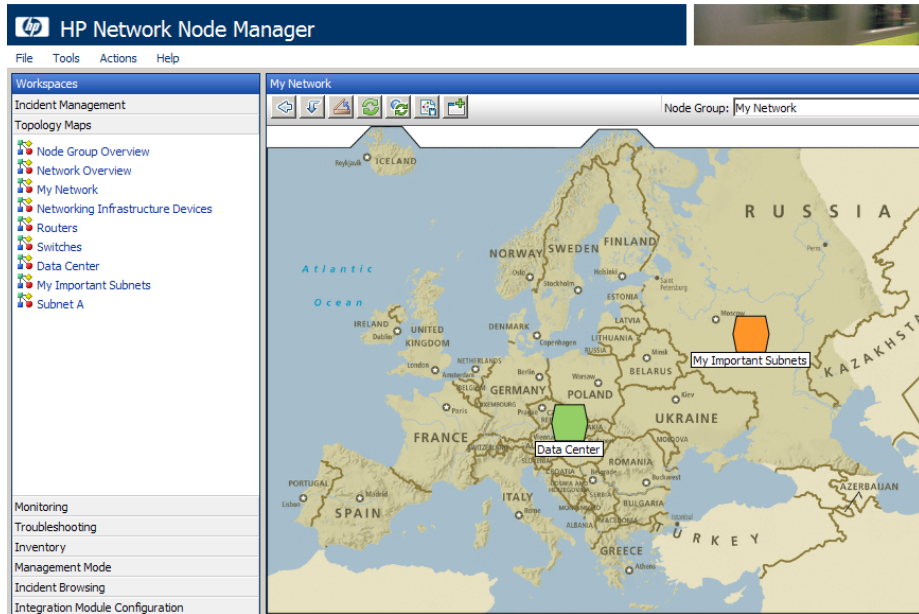
Click **Configuration** from the NNMi console, then click **User Interface Configuration**.



Change the Initial View selection to the first node group in the Topology Maps workspace. This is your My Network map because we set the order to 5.



Now when you log back into NNMi, the initial view is the My Network map.



Maintaining NNMi

Backing up and Restoring NNMi Data

NNMi provides two backup and restore scripts to help you protect your data.

The first backup script is called `nnmbackup.ovpl`. You can use this script either *online* or *offline*. The online option enables you to run the script without stopping NNMi. Running this script generates a backup with a date and time stamp in the file name so you can specify the same target directory each time. This command backs up four different data sets: `configuration`, `topology`, `event` and `all`. You can decide if you only want to back up a subset. HP recommends using the `all` option. This backup is very complete, and contains everything you will need to restore NNMi.



The `nnmbackup.ovpl` script backs up a number of binary files (usually `.ear` files). This can make the backup large so plan accordingly.

Although the `nnmbackup.ovpl` script backs up a lot of data, it completes quickly.

The following command shows an example of using the backup script:

```
nnmbackup.ovpl -type online -scope all -force -archive  
-target /var/tmp/mybackups/patch4
```

This creates a file with a name similar to `nnm-bak-20090222163003.tar`.

The associated restore script is `nnmrestore.ovpl`. This command is easy to use, and requires the backup file or directory created from the `nnmbackup.ovpl` script. You must stop NNMi with a `ovstop -c` command to be able to run this script.

An example `nnmrestore.ovpl` script usage is listed below:

```
nnmrestore.ovpl -force -source /var/tmp/mybackups/patch4/  
nnm-bak-20090222163003.tar
```

The source directory should contain all of the files from the backup or the single tar file. If the source is a tar file, the script extracts the tar file to a temporary folder in the current working directory. The script removes the temporary folder after it completes the restore.



Never restore a backup across NNMi patch versions or restore a backup from a previous patch level of NNMi. For example, suppose you are running patch 4 on your NNMi management server. After you run a backup, you upgrade to patch 5. At this point, you should not restore the backup from the NNMi management running patch 4 onto the patch 5 code. This will cause fatal errors for NNMi. You might want to track which version of the patch you are running in the backups using a naming convention for the directories. You see a possible way of doing this above by naming the backup directory patch4.

The second script used for backup is `nnmbackupembdb.ovpl`. Only use the `nnmbackupembdb.ovpl` script to back up NNMi servers configured to use the embedded database. Also, NNMi must be running before you run the `nnmbackupembdb.ovpl` script. The `nnmbackupembdb.ovpl` script does not back up any additional files or executables. The result of using this script is almost as good as doing a full backup, but avoids the resource overhead that comes with a full backup. NNMi stores all of its topology, configuration and events in the NNMi database, so the `nnmbackupembdb.ovpl` script could be sufficient for you. The `nnmbackupembdb.ovpl` script generates a single file along with a time stamp, and can be compressed to a very small size. See the `nnmbackupembdb.ovpl` reference page or the UNIX manpage for complete details.



Make sure to experiment with the `nnmbackupembdb.ovpl` script to familiarize yourself with its capabilities.

An example `nnmbackupembdb.ovpl` script usage is listed below:

```
nnmbackupembdb.ovpl -force -target /var/tmp/mybackups/patch4
```

Running the above script creates a file similar to `nnm-bak-20090222165029.pgd`.

You can restore this backup by running the `nmrestoreembdb.ovpl` script. An example `nmrestoreembdb.ovpl` script usage is listed below:

```
nmrestoreembdb.ovpl -force -source /var/tmp/mybackups/  
patch4/nnm-bak-20090222165029.pgd
```

Consider using a mixture of both types of backups. For example you could complete a weekly backup using the `nnmbackup.ovpl` script and complete a daily backup using the `nnmbackupembdb.ovpl` script.

Exporting and Importing NNMi Configurations

Configuring NNMi is one of the most important tasks you do. Although your configuration is backed up as part of the `nnmbackup.ovpl` and `nnmbackupembdb.ovpl` scripts mentioned earlier, consider using the `nnmconfigexport.ovpl` and `nnmconfigimport.ovpl` scripts included in NNMi. Using these scripts provides flexibility when it comes to restoring NNMi configuration. These scripts enable you to take a snapshot of the present NNMi configuration. It also divides the configuration into small pieces. This enables you to restore just one piece of NNMi configuration if you need to revert back to a recent snapshot.

For example, suppose you need to create a lot of node groups. Use the `export` script to take a snapshot of the configuration at strategic points along the way so you can revert back if you make a significant mistake.

The backup script is `nnmconfigexport.ovpl`. Use the `nnmconfigexport.ovpl` script to specify a configuration area such as discovery, node groups, incidents, and many others. There is also an `all` option to export all of the configuration information. See the `nnmconfigexport.ovpl` reference page or the UNIX manpage for complete details.

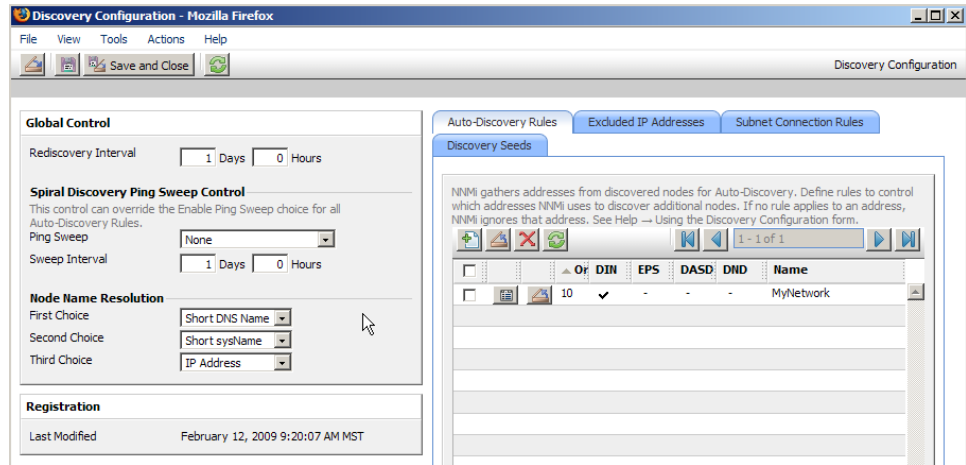
An example `nnmconfigexport.ovpl` script usage is listed below:

```
nnmconfigexport.ovpl -u admin -p adminpw -c nodegroup -f /  
var/tmp/myconfigs/nodegroup.xml
```

In this example, NNMi displayed the following message:

```
Successfully exported /var/tmp/myconfigs/nodegroup.xml.
```

Each exported configuration roughly corresponds to one configuration area in the NNMi console. For example, there is a `nnmconfigexport.ovpl` script option called `disco`. This option corresponds to the `Discovery Configuration` form shown below. NNMi `Auto-Discovery Rules` are part of this configuration export as are the `Rediscovery Interval`, `Node Name Resolution`, and other data. The one exception is that the export does not include data shown in the `Discovery Seeds` tab. `Discovery Seeds` have their own `discoseed` option for the `nnmconfigexport.ovpl` script.



You could run the `nnmconfigexport.ovpl` script daily choosing the all option. The `nnmconfigexport.ovpl` script duplicates NNMi data within the backup in XML format. With the data in XML format, you can selectively restore configuration snapshots that you cannot do individually with the `nnmbackup.ovpl`, `nnmrestore.ovpl`, `nnmbackupembdb.ovpl` and `nnmrestoreembdb.ovpl` scripts.

Be aware that the `nnmconfigexport.ovpl` script does not generate a date and time stamp on the files. If you want to automate this command, put the data and time stamp on the directory name. These XML files are very small so storage space is not an issue.

To restore the configuration, use the `nnmconfigimport.ovpl` script. You do not need to specify a configuration area because this is implied by the file contents.

An example `nnmconfigexport.ovpl` script usage is listed below:

```
nnmconfigimport.ovpl -u admin -p adminpw -f /var/tmp/  
myconfigs/disco.xml
```

As with the `nnmbackup.ovpl` and `nnmbackupembdb.ovpl` scripts, you should not use these scripts across patch versions. The good news is that NNMi validates the configuration file and rejects it during the import if it is invalid for the current version of NNMi. The `nnmconfigimport.ovpl` script will never corrupt your present configuration but it will override it if the format is okay, so use caution when using this script.

Trimming Traps from the Database

Traps that pass all of the NNMi filters are eventually stored in the NNMi database. Traps can come in high volume and bog down the NNMi database. HP recommends that you regularly trim traps from your NNMi database. Use the `nmtrimincidents.ovpl` script to trim traps from your database. You can archive these traps if necessary.

An example `nmtrimincidents.ovpl` script usage is listed below:

```
nmtrimincidents.ovpl -u admin -p adminpw -age 1 -incr weeks  
-origin SnmpTrap -trimOnly -quiet
```

This example usage trims any traps older than 1 week. This usage does not archive the traps. See the `nmtrimincidents.ovpl` reference page or the UNIX manpage for more options. Use `nmtrimincidents.ovpl` in a cron job to clear out old unnecessary trap incidents on a regular basis. NNMi eventually forces you to do this by stopping storage of traps after it reaches a limit of 100,000 traps in the NNMi database.

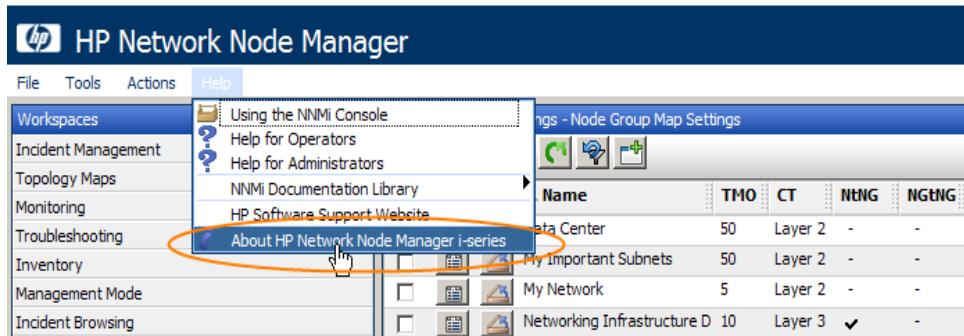


This reference to the NNMi database is not the same as the trap datastore. See *Step-by-Step Guide to Managing SNMP Traps in NNMi* for more information.

Checking NNMi Health

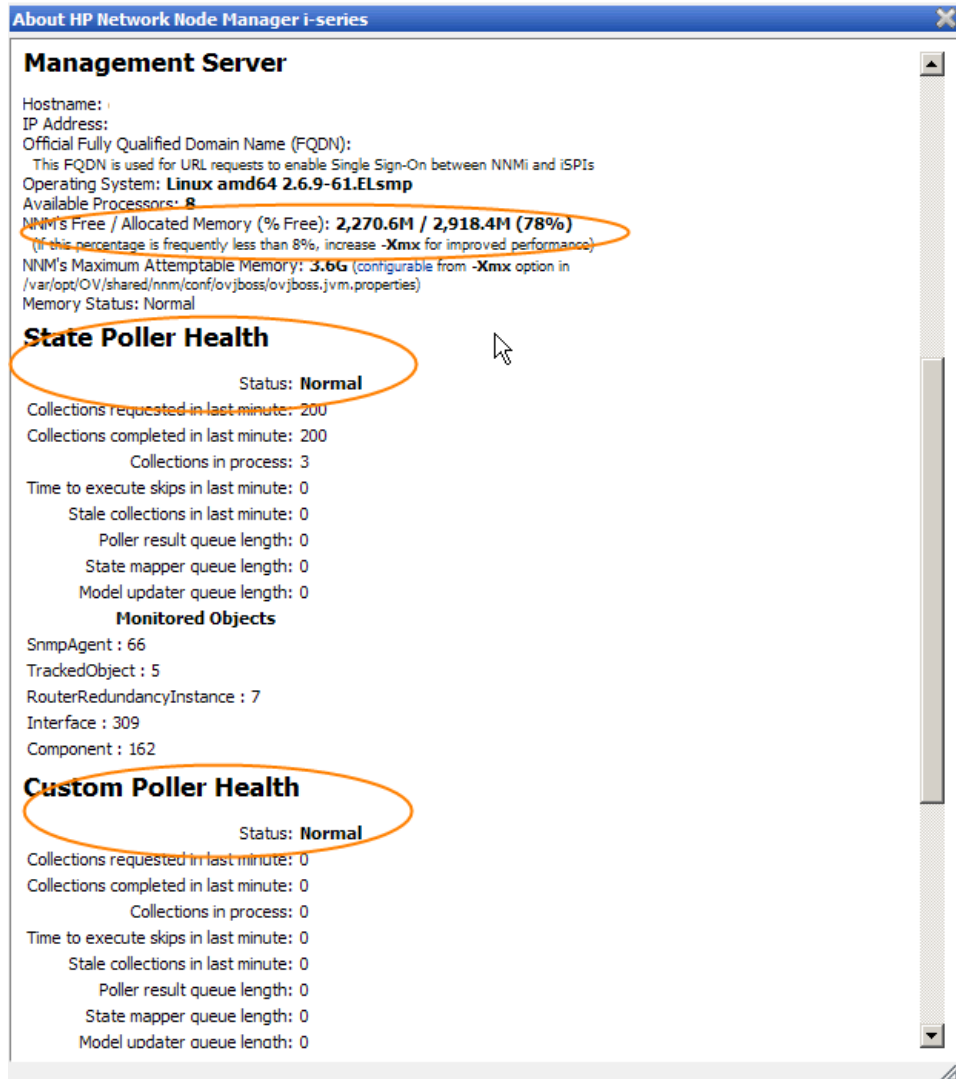
You can check the general health of NNMi with a few different tools.

From the NNMi console, use the Help->About HP Network Node Manager i-series menu item for a listing of some important data.



After NNMi displays the results, check NNMi's free memory. This number should not go below 8%. If it does, increase your memory as displayed in the form.

You can also check the State Poller Health and the Custom Poller Health. These should be in a Normal status at all times. If they are not Normal, then the State Poller is behind. Be aware that the State Poller is separate from APA (Active Problem Analyzer Service). It is possible that the State Poller is keeping up, but that APA is behind. NNMi does not currently display a health check of APA.



You can also check the NNMI health by running the `ovstatus` command. It is important to run the `ovstatus` command using the `-v` (verbose) option against the `ovjboss` process. To do this, run the **`ovstatus -v ovjboss`** command and view the display to check the health of the `ovjboss` process. A typical health output would look something like the following:

```
# ovstatus -v ovjboss
```

```
object manager name: ovjboss
state:                RUNNING
PID:                  20413
last message:         Initialization complete.
exit status:          -
additional info:
```

SERVICE	STATUS
CPListener	Service is started
CommunicationModelService	Service is started
CommunicationParametersStatsService	Service is started
CustomPoller	Service is started
EventsCustomExportService	Service is started
ExtensionDeployer	Service is started
InstanceDiscoveryService	Service is started
IslandSpotterService	Service is started
KeyManager	Service is started
ManagedNodeLicenseManager	Service is started
ModelChangeNotificationAdapter	Service is started
MonitoringSettingsService	Service is started
NMSLogManager	Service is started
NamedPoll	Service is started
NetworkApplication	Service is started
NmsApa	Service is started
NmsDisco	Service is started
NmsEvents	Service is started
NmsEventsConfiguration	Service is started
NmsExtensionNotificationService	Service is started
NmsModel	Service is started
NmsWorkManager	Service is started
NnmTrapService	Service is started
PolicySynchronizer	Service is started
RbaConfig	Service is started
RbaManager	Service is started
SpmdbossStart	Service is started
StagedIcmp	Service is started
StagedSnmp	Service is started
StatePoller	Service is started
TrustManager	Service is started

Miscellaneous Tips

Some additional recommendations that you might want to consider are listed here:

- Use NNMi's embedded database, even for large scale. Tests show that Postgres is highly scalable. You do not need to consider Oracle just because you have a large network. Some features, such as application failover, only work with Postgres. Postgres is highly reliable and recommended by HP as the preferred database for NNMi. Do not worry if you lack Postgres database expertise, as you really do not need it. Postgres is embedded into NNMi and NNMi provides any required tools you need.
- Use caution when adjusting the SNMP timeout configuration. This timeout value increments with each timeout and can grow quickly beyond your original intention.
- From the NNMi console, click one of the topology map selections. After you see the resulting display, double-click one of the nodes to open a node form. Click the `Conclusions` tab and review the data to better understand why the current status is set for the node.
- Reduce the number of connections between node groups using the `End Points Filter` in the `Node Group Map Settings` form. Highly connected maps display slowly and NNMi will drop connections if necessary on the map.
- Do not use an `@` symbol in your SNMP strings. This is a reserved character for Cisco devices and causes unpredictable NNMi behavior.

Possible Usage Scenarios

Now that you successfully configured NNMi, you need to learn how to use it. This section presents three simple scenarios. The majority of NNMi users use trouble ticketing systems. For simplicity, these scenarios do not include these trouble ticketing tools. Instead, these scenarios assume you only have NNMi available.

Management by Exception

NNMi is excellent at identifying root cause problems associated with a network fault. These problems are presented as Key Incidents. From the NNMi console, click **Incident Management**, then click **Open Key Incidents**. By monitoring the Open Key Incident browser, you can pinpoint the exact cause of a network problem and begin working toward a solution. HP refers to this as *management by exception* since the incident browser shows these *exceptions* (or outages).

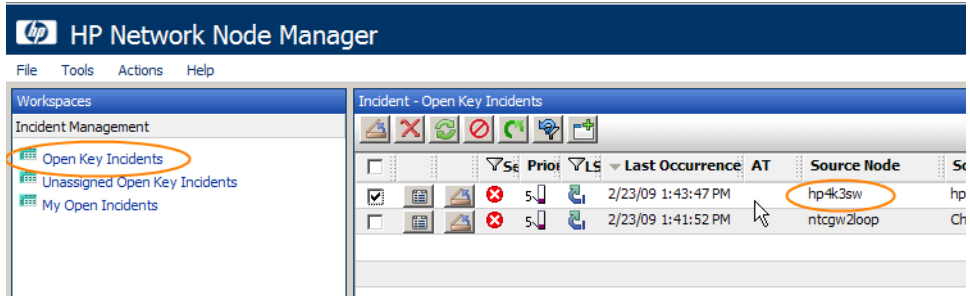
The *management by exception* approach includes the following advantages:

- You can quickly see the root cause of the problem.
- You can easily identify the source of the problem as the *source object*, such as an interface, address, node, or other possible sources.
- NNMi can forward Key Incidents to other products, such as HP OM.

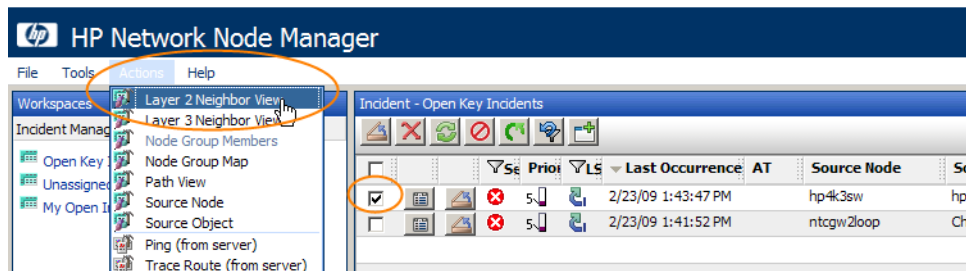
The *management by exception* approach includes the following challenges:

- It can be difficult for you to recognize the scope of an outage. A key `NodeDown` incident shows only the root cause, but the root cause node being down could affect connectivity to many other nodes.
- It can be difficult to prioritize incidents (which one to work on first). Not all `NodeDown` incidents are of equal importance.

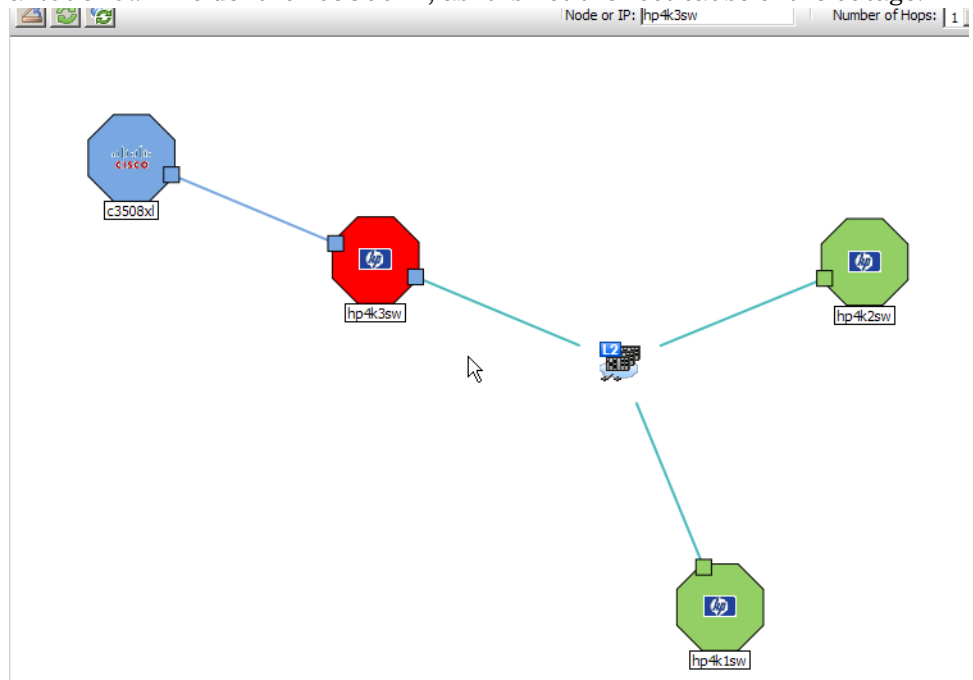
You see an example of Key Incidents below. From the NNMi console, click **Incident Management**, then click **Open Key Incidents**. NNMi displays all of the outstanding key incidents in your network and will update this list every 30 seconds. See the NNMi help for the definition of a key incident. Notice that NNMi filters this view is filtered by time, so you may need to use the pull-down menu to select an appropriate time value. The example below shows key incidents in the last hour. You can see that you had a node go down within the last hour.



Suppose you want to learn more about this outage. There are many actions that you can run from the actions menu. To better understand the scope of the outage, you launch a Layer 2 Neighbor View.



This displays a neighbor view centered around the source node, hp4k3sw. You could expand the number of hops if you want to see all of the affected nodes. In this case, one node c3508x1, is *behind* the outage. NNMi did not generate a NodeDown incident for c3508x1, as it is not the root cause of the outage.



Double-click the critical node, hp4k3sw, for further details. Suppose you want to see a history of the incidents on this node to try to establish a pattern. For example, you wonder if this node is node going down each evening. If you click the **Incidents** tab, NNMi displays a history of the incidents related to this node.

Node: "hp4k3sw" - Mozilla Firefox

File View Tools Actions Help

Save and Close Delete Node

Basics

Name: hp4k3sw
 Hostname: hp4k3sw.cnd.hp.com
 Management Address: 15.2.131.61
 Status: Critical
 Node: Managed
 Management Mode: Managed

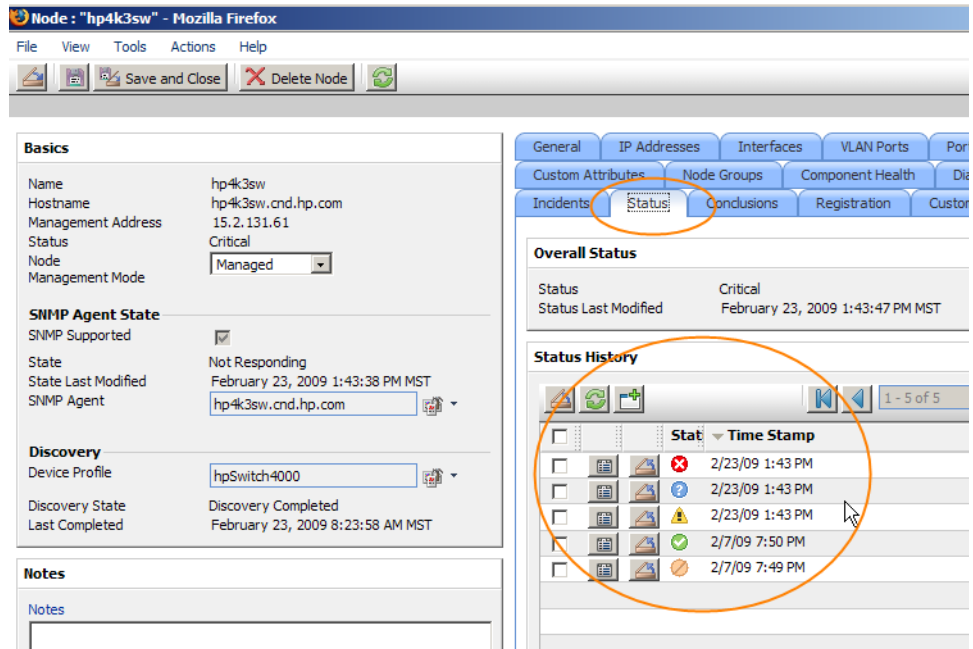
SNMP Agent State

SNMP Supported:
 State: Not Responding
 State Last Modified: February 23, 2009 1:43:38 PM MST

General IP Addresses Interfaces VLAN Ports Port
 Custom Attributes Node Groups Component Health Dia
Incidents Status Conclusions Registration Custom

	Seve	LS	Last Occurrence	CN	Mess
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2/23/09 1:43:47 PM	<input type="checkbox"/>	Node
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2/7/09 7:54:31 PM	<input type="checkbox"/>	Node

If you click the **Status** tab, NNMi displays a history of the status related to this node. All of these steps enable you to better understand the problem and begin looking for a solution. NNMi provides you with many other tools such as ping, telnet, and trace route to the node from the NNMi management server. You can also look at upstream nodes to validate that they continue to operate properly.



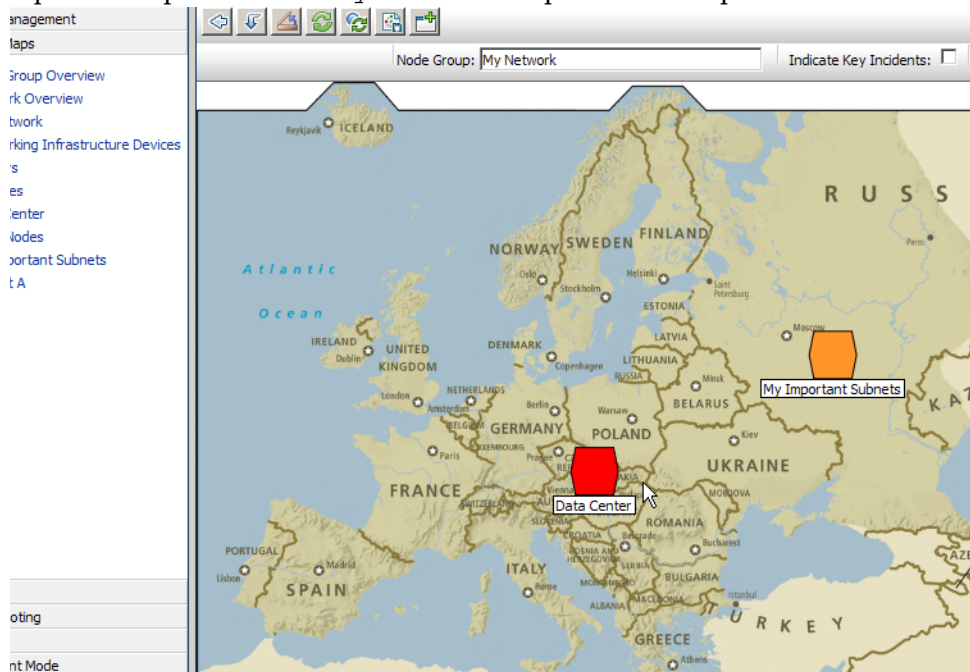
NNMi enables you to annotate incidents with notes so you can keep a log of information about the debugging progress.

Map Based Management

Another method of network management is to create maps that you can watch for status change. Usually these maps will correspond to geography or building but they could be arranged in many ways. These maps represent a hierarchy of node group maps. The status is propagated from the *leaf nodes* up to the parent node group maps. By default, NNMi propagates the most critical node status in the node group up the hierarchy. This enables you to monitor status from a high level. When a top level node group map changes color from green to red, yellow or orange, you can drill into the node group

maps until you find the problem node. After you reach the problem node, you can take actions similar to those described in the previous section to debug the problem. Similar to incidents, nodes and interfaces can be annotated with notes if you want to keep a log of information about the debugging progress.

The following screen shot shows an example of the data center having a problem that you need to correct. You drill down into this node group map to find the faulting node. To operate in this mode, you need a *default map* that NNMI displays after you complete the initial log on. Another way to navigate to this map from the NNMI console is to click **Topology Maps**, then click your top-most map. This is the My Network map in this example.



The benefits to this approach include:

The *map-based management* approach includes the following advantages:

- You can easily scope the outage. It becomes obvious quickly if other nodes are affected based on the status of neighboring nodes.
- You can easily identify the affected location. It is harder for you to identify this from an incident. The helps you decide what to work on first.

The *map-based management* approach includes the following challenges:

- To find the source of the problem, you need to drill into the map more than when using the management by exception approach. More drilling, means you have to open the node and look at the conclusions tab to see what exactly the problem is with the node.
- NNMi does not propagate node status to other tools such as HP OM.
- NNMi does not color a node group map *more red* if one node is already down, then another node goes down in the same group.

List Based Management

The final method is to manage your network from a dynamic list. NNMi provides dynamically updated tables that show nodes or interfaces experiencing problems. NNMi usually updates this list every 15 seconds. You can easily get a list of nodes or interfaces that are experiencing problems. From this list, you can use tools, as shown in the previous methods, to diagnose and fix the problem. Because this list is dynamic, NNMi clears the nodes or interfaces from this list as the nodes or interfaces return to a normal status.

The following screen shot shows an example. From the NNMi console, click **Monitoring**, then click **Critical Nodes**.

The screenshot shows the NNMi console interface. On the left, a sidebar menu is visible with the following categories: Workspaces, Incident Management, Topology Maps, and Monitoring. Under Monitoring, several options are listed, including 'Critical Nodes', which is circled in red. The main window displays a table titled 'Node - Critical Nodes'. The table has columns for 'Name', 'Hostname', 'Management Ad', and 'System Locati'. Two rows are visible: one for 'hp4k3sw' and one for 'unnum1'. The 'hp4k3sw' row is highlighted, and a mouse cursor is pointing at it. The 'System Locati' for 'hp4k3sw' is '5U E. CPU Room' and for 'unnum1' is 'ipv6.net'.

Name	Hostname	Management Ad	System Locati
hp4k3sw			5U E. CPU Room
unnum1			ipv6.net

The *list-based management* approach includes the following advantages:

- You know how many nodes or interfaces you need to investigate.
- This is a much simpler approach as you do not need to drill into NNMi maps to troubleshoot your network.

The *map-based management* approach includes the following challenges:

- There is no easy *electronic trail* to see patterns of outage as NNMi does not keep a long history of status.
- It is difficult to scope the size of an outage as NNMi only shows critical nodes. NNMi does not mark *downstream* nodes as critical.
- It is difficult to know where the node is physically located.
- NNMi does not propagate node status to other tools such as HP OM.

Conclusion

This document explained a simple NNMi deployment completed on a small test network. You read about tasks that included installing a license, creating users, configuring communication, discovery, incidents, traps, actions, and the NNMi console. This document also explained how to complete maintenance tasks for NNMi, how to monitor NNMi health, provided some miscellaneous tips, and explained some possible use scenarios for NNMi. Hopefully you have been able to get a feel for how simple it is to get NNMi running and how NNMi can significantly improve your ability to maintain a healthy network by giving quick, precise alerts to problems in your network.

