HP Network Node Manager i-Series Software

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

Software Version: NNMi 8.1x patch 4 (8.12)

Online Help: Custom Poller for Administrators

Document Release Date: June 2009

Software Release Date: June 2009

PDF Version of NNMi Online Help

This document is a PDF version of the NNMi 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.

For information about third-party license agreements, see the license-agreements directory on the product installation media.

Copyright Notices

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

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

This product includes Commons Discovery software developed by the Apache Software Foundation (http://www.apache.org/). Copyright © 2002-2009 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.

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

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

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

Table of Contents

Custom Poller for Administrators

The Custom Poller feature enables you to take a proactive approach to network management by using SNMP MIB variables to specify additional information that NNMi should poll. You can also specify States that should be assigned to polled 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 about COM (communication) port, Loopback interface, or Ethernet Adapter Status (hrDeviceStatus).

Note: The MIB variables that you want NNMi to poll must be loaded on the NNMi management server.

Custom Poller for Administrators provides information for how to configure the Custom Polling feature. See Custom Poller for Operators for information on viewing the results of Custom Poller.

As an Administrator, you want to perform the following tasks:

"Load the Required MIBs" (on page 6)

"Enable or Disable Custom Poller" (on page 7)

"Create a Custom Poller Collection" (on page 7)

"Configure Basic Settings for a Custom Poller Collection" (on page 9)

"Specify the MIB Variable Information for a Custom Poller Collection" (on page 10)

"Configure Threshold Information for a Custom Poller Collection" (on page 12)

"Configure Comparison Maps for a Custom Poller Collection" (on page 13)

"Create a Policy" (on page 15)

Refer to the "Step-by-Step Guide to Using the NNMi Custom Poller" whitepaper for more details about configuring Custom Poller.

Load the Required MIBs

NNMi requires that a MIB be loaded on the NNMi management server before you can specify that you want to poll one of its MIB variables.

NNMi automatically loads a set of MIBs on the NNMi management server during installation.

To load additional MIBs, use the nnmloadmib.ovpl command.

For example, to load the HOST-RESOURCES-MIB, you would enter a command similar to the following:

```
nnmloadmib.ovpl -u <NNMiadminUsername> -p <NNMiadminPassword> -load /temp/HostResources.mib
```

After the MIB is loaded on the NNMi management server, you can view the MIB when creating a Custom Poller Collection. See "Create a Policy" (on page 15) for more information.

Related Topics

"Enable or Disable Custom Poller" (on page 7)

"Create a Custom Poller Collection" (on page 7)

Enable or Disable Custom Poller

Custom Poller Configuration workspace 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 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.

The Custom Poller Collections tab enables you to create a Custom Poller Collection. See "Create a Custom Poller Collection" (on page 7) for more information.

The Policies tab enables you to create one or more policies for a Collection. See "Create a Policy" (on page 15) for more information.

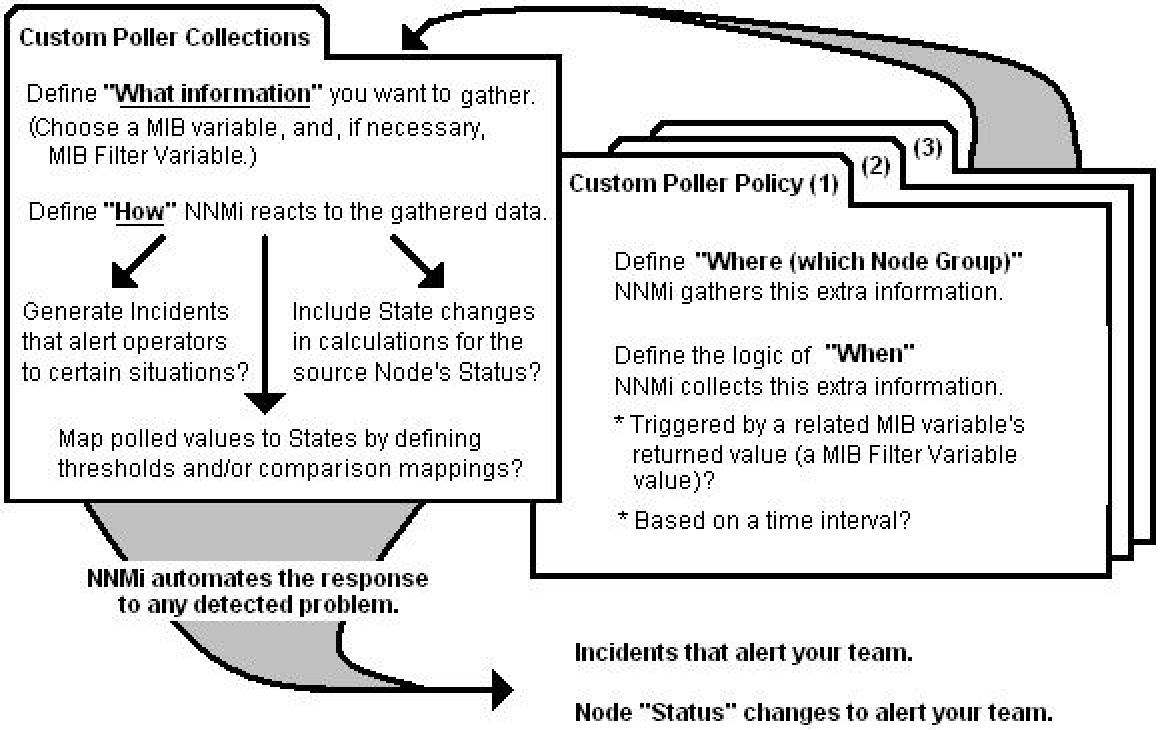
Create a Custom Poller Collection

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 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 the specified MIB variable is discovered, the results appear in a Custom Polled Instance object. The Polled Instance object is updated whenever a change in the MIB Poll Variable's State occurs and includes the polled value that caused the State to change.

Click here for a diagram that describes Custom Poller Collections and their associated Policies:


Custom Poller Collections
Define "What information" you want to gather. (Choose a MIB variable, and, if necessary, MIB Filter Variable.)
Define "How" NNMi reacts to the gathered data.
Generate Incidents that alert operators to certain situations?
Include State changes in calculations for the source Node's Status?
Map polled values to States by defining thresholds and/or comparison mappings?
NNMi automates the response to any detected problem.
Custom Poller Policy (1)
(2)
(3)
Define "Where (which Node Group)" NNMi gathers this extra information.
Define the logic of "When" NNMi collects this extra information.
* Triggered by a related MIB variable's returned value (a MIB Filter Variable value)?
* Based on a time interval?
Incidents that alert your team.
Node "Status" changes to alert your team.

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 15) for more information.

Custom Poller Collection Configuration Tasks

Task	How
"Configure Basic Settings for a Custom Poller Collection" (on page 9)	Provide the basic information for a Custom Poller Collection configuration.

Task	How
"Specify the MIB Variable Information for a Custom Poller Collection" (on page 10)	You specify the MIB variable you want to poll. Navigate the MIB tree to select the MIB Poll Variable you want to use.
"Configure Threshold Information for a Custom Poller Collection" (on page 12)	*Optional*. Specify minimum and maximum threshold values for the MIB Poll Variable and assign these thresholds to States.
"Configure Comparison Maps for a Custom Poller Collection" (on page 13)	*Optional*. Use Comparison Maps to assign a State value to a potential polled value of a MIB Poll Variable.

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 States are Normal, NNMi then 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.

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.

Basics for this Custom Poller Collection

Attribute	Description
Name	The name for the Custom Poller Collection configuration. **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 the specified MIB Poll Variable is discovered, the results appear in a Polled Instance object. The Polled Instance object is updated whenever a change in State occurs and includes the 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 **Crit-**

Attribute	Description
	ical 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**.

See "Specify the MIB Variable Information for a Custom Poller Collection" (on page 10) for information about the Variable attributes.

Specify the MIB Variable Information for a Custom Poller Collection

When specifying the MIB variable information, note the following:

- The MIB Poll Variable must be loaded on the NNMi server.
- You specify only one MIB Poll Variable per Custom Poller Collection.
- You navigate the MIB tree to select the MIB Poll Variable.

Variable Attributes

Attribute	Description
Name	The name you want to use for the MIB Poll Variable. This name can be the same name as the MIB variable, or you can enter a name of your choice.
MIB Poll Variable	Use this attribute to select the MIB variable that you want NNMi to poll. Valid types for MIB Poll Variables include the following: • INTEGER, Integer32 • Unsigned32 • Counter, Counter32 • Counter64 • Gauge, Gauge32 • TimeTicks • OCTET STRING When configuring thresholds, the MIB Poll Variable must be a numeric type. (OCTET STRING type is not supported.) When evaluating Threshold or Comparison Map configurations that include MIB Poll Variables of type Counter (for example, Counter32 and Counter 64), NNMi performs state mapping using the difference in value between the most recent poll and the poll before it. You must select a value 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 Poller Collection form, allow 1 minute for NNMi to cache the new MIB information, then open the MIB tree again. After you select the MIB variable, NNMi displays the MIB variable's Object Identifier

Attribute	Description
	(OID). **Note**: If you choose a MIB Poll 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: 1. MIB Poll Variable: **hrDeviceStatus** 2. MIB Filter Variable: **hrDeviceDescr** 3. MIB Filter: **COM*** See "Create a Policy" (on page 15) for more information about the MIB Filter. Note the following: - The MIB containing the variable must be loaded on the NNMi management station. - 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 Poll Variable and the MIB Filter Variable.

Attribute	Description
MIB Filter Variable	The MIB Filter Variable is the MIB variable whose value you want to use as a filter to determine when NNMi collects additional information. If you specify a MIB Filter Variable, you must also specify a MIB Filter that determines which polled value causes NNMi to collect the additional information. Valid types for MIB Filter Variables include the following: • INTEGER, Integer32 • Unsigned32 • Gauge, Gauge32 • OCTET STRING • IpAddress 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 Policies that are associated with the Custom Poller Collection. b. Edit the Custom Poller Collection to remove the MIB Filter Variable. c. Recreate the Policies for the Custom Poller Collection, omitting the MIB Filter value.

Configure Threshold Information for a Custom Poller Collection

Prerequisite: You must have specified the MIB Variable you want to poll. See "Specify the MIB Variable Information for a Custom Poller Collection" (on page 10) for more information.

Thresholds specify minimum and maximum values for the MIB Variable that is polled. These values are used to determine when to generate and 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 Threshold settings to determine State values. If the threshold evaluates to Normal, NNMi checks for a non-Normal State using any Comparison Maps configuration.
- The MIB Poll Variable must be a numeric type. (OCTET STRING type is not supported.)
- When evaluating Threshold configurations that include MIB Poll Variables of type Counter (for example, Counter32 and Counter 64), NNMi performs state mapping 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.

Threshold Attributes

Attribute	Description
High State	Value used to indicate the Polled Instance State when the High Value is exceeded. This value is required if a High Value is defined.
High Value	Value used to indicate the high threshold has been reached. This value is required if a High State is defined.
High Value Rearm	Value used to indicate 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 High Value must be exceeded to transition to the High State. The default value is 1.
Low State	Value used to indicate the Polled Instance State when the value is lower than the Low Value. This value is required if a Low Value is defined.
Low Value	Value used to indicate the low threshold has been reached. This value is required if the Low State is defined.
Low Value Rearm	The value used to indicate the Polled Instance is no longer in the Low State. The default value is the Low Value.
Low Trigger Count	The number of consecutive times the Low Value must be exceeded 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 Poll Variable is polled.

Custom Poller enables you to map the returned value of a MIB Poll Variable to a Custom Poller Polled Instance State. These values are used to determine when to generate and 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 whose criteria is met defines the State for the Polled Instance.
- You can configure Comparison Maps, which also 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 Normal, NNMi checks for a non-Normal State using any Comparison Maps configuration.See "Configure Threshold Information for a Custom Poller Collection" (on page 12) for more information about configuring thresholds.
- When evaluating Threshold or Comparison Map configurations that include MIB Poll Variables of type Counter (for example, Counter32 and Counter 64), NNMi performs state mapping using the difference in value between the most recent poll and the poll before it.

To configure Comparison Maps for a MIB Poll 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.
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 15) for more 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.
Polled Value	The value returned when the MIB Poll Variable is polled.

Attribute	Description
Comparison Operator	Operator used to evaluate the Polled Value and subsequently determine its State. For example, the < (less than) Comparison Operator indicates the polled value must be less than the Polled 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.
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**.

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 the Node Group whose MIB variable you want NNMi to poll.

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 Custom Poller 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.

Custom Poller Policy Attributes

Attribute	Description
Name	The Name of the Policy configuration. **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 `ifOper` as the MIB Poll Variable. • One Policy uses the Routers Node Group and the second Policy uses the Switches Node Group. • Each Policy has a different Polling Interval. In the example scenario above, NNMi would poll the switch routers 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	The Active State for the Custom Collect Policy. The Active State enables you to specify which Custom Poller Policies you want to enable. You can also use this option to temporarily disable a Custom Poller 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 the Custom Poller Policy is temporarily disabled. 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.
MIB Filter	The MIB Filter value to be used as the filter for determining the Polling Instances. **Note**: 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`. **Note:** If you are polling a MIB variable that has repeating 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. 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*.

Attribute	Description
	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 (!) in your search, use a leading backslash (\) before the special character.
Polling Interval	The interval in which to perform the Custom Poll.

Appendix A: Glossary Terms

A

Anycast Rendezvous Point IP Address

Rendezvous Point addresses are loopback addresses used for routers in multi-cast network configurations.

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.

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.

Appendix B: Index

hp®

invent

Java™
COMPATIBLE