Step-by-Step Guide to Using the NNMi Custom Poller

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

Software version: NNMi 8.11

Introduction

This whitepaper steps through an example of setting up the Custom Poller to monitor a MIB that NNMi does not monitor by default.

Note the following:

- The Custom Polling feature is only available with NNMi 8.11 or greater.
- This example uses a Unix NNMi server, but can be followed on a Windows server.
- Some screen captures might be slightly different than those that appear in the NNMi graphical user interface.

This whitepaper guides you through the following steps for setting up your MIB and a Custom Poll:

Setting Up Your MIB
Step 1: Identify the MIB Variable You Want to Poll
Step 2: Ensure the MIB Includes Supported Types
Step 3: Load the Required MIB
Step 4: Use an SNMP Query to View Current MIB Variable Values

Setting Up a Custom Poll
Step 1: Enable Custom Poller
Step 2: Create a Custom Poller Collection
Step 3: Create a Policy for a Custom Poller Collection

Examine the Results of Your Custom Poll
Step 1: View the Custom Node Collections Associated with Custom Poller Policies
Step 2: View Details of a Custom Node Collection
Step 3: View Details of a Polled Instance
Step 4: Evaluate the Results of the Custom Poll

Setting Up Your MIB

Step 1: Identify the MIB Variable You Want to Poll

To begin, identify a MIB variable that you want to poll.

For this example, we monitor the disk usage on Microsoft PCs using the rfc2790-HOST-RESOURCES-MIB. This MIB is shipped with NNMi under the following directory:
`/var/opt/OV/share/snmp_mibs/Standard`

This example uses rfc2790-HOST-RESOURCES-MIB for the following reasons:

- It is supported on most Unix platforms
- The availability of Microsoft PCs makes this example easy to test
- You can easily increase disk space usage to change the resultant query and trigger a State change

Step 2: Ensure the MIB Includes Supported Types

Make sure you are familiar with the MIB you will be using. This is especially important because the variables used must have a type that NNMi supports. See Troubleshooting Tips for a list of supported MIB variables.

Ways to study the MIB include:

- Use an SNMP browser. Some free browsers are available on the web. Many commercially available SNMP browsers are excellent.
- Use the HP tool nnmsnmpwalk.ovpl.
- Read through the MIB with a text editor.

When you study rfc2790-HOST-RESOURCES-MIB, notice it includes a table called hrStorageTable. This table has the following members that we can use to identify the disk and see the storage used:

- hrStorageDescr
- hrStorageUsed

An excerpt from the rfc2790-HOST-RESOURCES-MIB is shown below:

```
HrStorageEntry ::= SEQUENCE {
hrStorageIndex Integer32,
hrStorageType AutonomousType,
hrStorageDescr DisplayString,
hrStorageAllocationUnits Integer32,
hrStorageSize Integer32,
hrStorageUsed Integer32,
hrStorageAllocationFailures Counter32
}
```

As shown in the example excerpt, hrStorageDescr is of type DisplayString and hrStorageUsed is of type Integer32. The NNMi Custom Poller supports both of these types.

Note:
DisplayString is an alias for OCTET STRING as per rfc1213-MIB-II.

Step 3: Load the Required MIB

NNMi's Custom Poller requires that the MIB be loaded onto the NNMi management server.

To check whether the MIB is already loaded, run the "`nnmloadmib –list`" command:

```
# /opt/OV/bin/nnmloadmib –list
```

Your results should be similar to the following output:

```
The following MIBs are loaded:
rfc1213-MIB-II
rfc1493-BRIDGE
rfc2863-IF-MIB
rfc1269-BGP
rfc1850-OSPF-MIB
rfc2127-ISDN-MIB
CISCO-SMI.my
CISCO-VTP-MIB.my
CISCO-HSRP-MIB.my
CISCO-ENVMON-MIB.my
```

As shown in the example output, rfc2790-HOST-RESOURCES-MIB is not loaded. To load this MIB, use the "`nnmloadmib.ovpl`" command:

```
# nnmloadmib.ovpl -u admin -p pwd -load rfc2790-HOST-RESOURCES-MIB
```

Note:
This command often returns without feedback.

Next, verify the MIB is loaded using the "`nnmloadmib –list`" command again:

```
# /opt/OV/bin/nnmloadmib –list
```

You should see the rfc2790-HOST-RESOURCES-MIB in the return list as shown in the example output below:

```
The following MIBs are loaded:
rfc1213-MIB-II
rfc1493-BRIDGE
rfc2863-IF-MIB
rfc1269-BGP
rfc1850-OSPF-MIB
rfc2127-ISDN-MIB
CISCO-SMI.my
CISCO-VTP-MIB.my
CISCO-HSRP-MIB.my
CISCO-ENVMON-MIB.my
rfc2790-HOST-RESOURCES-MIB
```

Step 4: Use an SNMP Query to View Current MIB Variable Values

Next, we perform an SNMP query on the Windows PC to become familiar with the values returned from a real node.

Because we want to check the storage used on the C: drive, look for the string from the hard disk that begins with "C:\". As shown in the example output below, one of the string begins with "C:\".

So that we can set a threshold for this example, note the current storage value. As shown in the example output below, the current storage as `14186770`.

```
# nnmsnmpwalk.ovpl -u usr -p pwd -c public ksmith32 hrStorageTable

hrStorageIndex.1 : ASN_INTEGER: 1
hrStorageIndex.2 : ASN_INTEGER: 2
hrStorageType.1 : ASN_OBJECT_ID: .1.3.6.1.2.1.25.2.1.4
hrStorageType.2 : ASN_OBJECT_ID: .1.3.6.1.2.1.25.2.1.7
hrStorageDescr.1 : ASN_OCTET_STR: C:\ Label:PC COE Serial Number 8 483d1e1
hrStorageDescr.2 : ASN_OCTET_STR: D:\
hrStorageAllocationUnits.1 : ASN_INTEGER: 4096
hrStorageAllocationUnits.2 : ASN_INTEGER: 0
hrStorageSize.1 : ASN_INTEGER: 19535032
hrStorageSize.2 : ASN_INTEGER: 0
hrStorageUsed.1 : ASN_INTEGER: 14186770
hrStorageUsed.2 : ASN_INTEGER: 0
hrStorageAllocationFailures.1 : ASN_COUNTER: 0
hrStorageAllocationFailures.2 : ASN_COUNTER: 0
```

We now have the information we need to set up a Custom Poll.

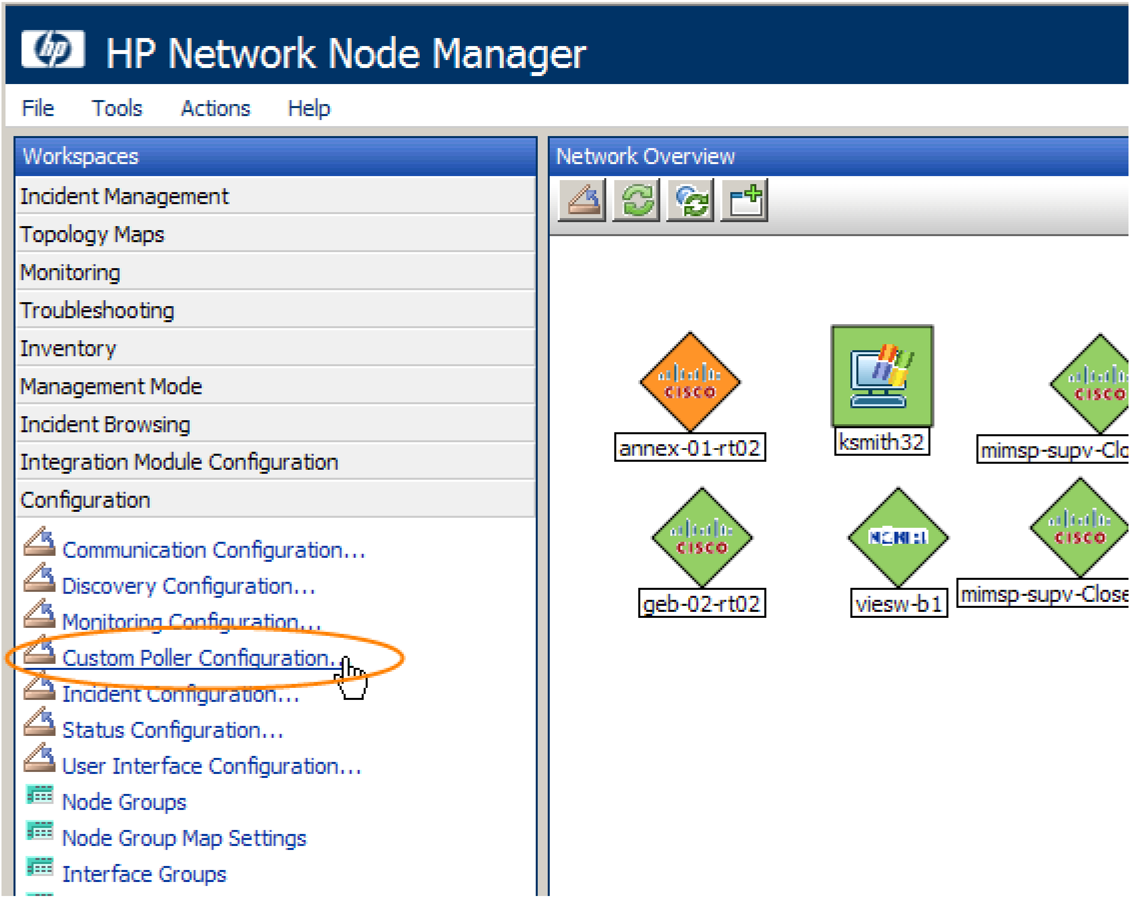
Setting Up a Custom Poll

Step 1: Enable Custom Poller

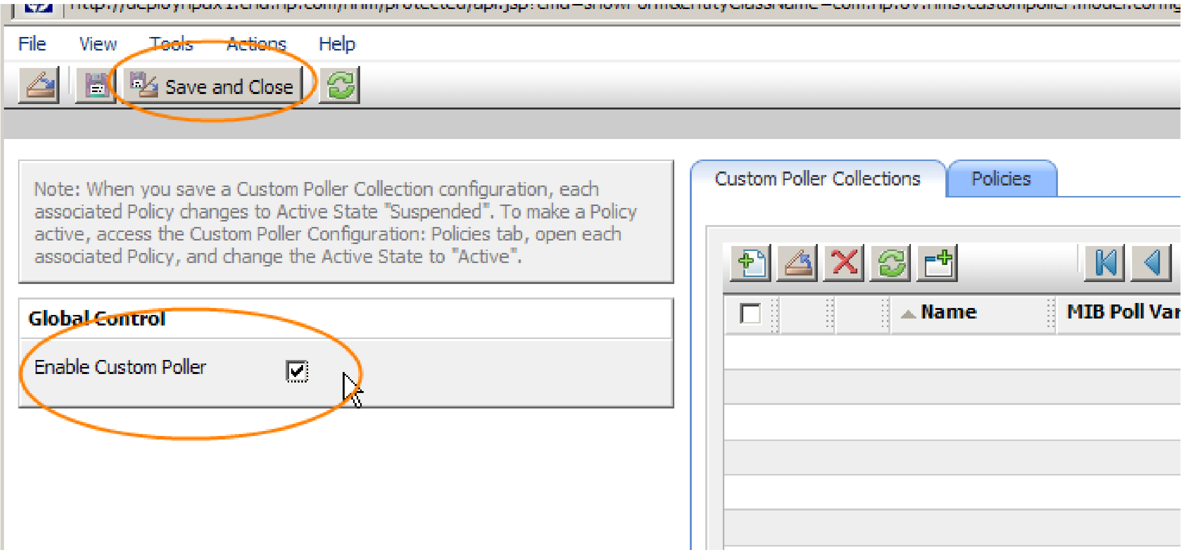
Custom Poller is not enabled by default.

To enable Custom Poller:

1. Navigate to the **Configuration** workspace.
2. Select **Custom Poller Configuration**.



3. Click to check **Enable Custom Poller**.
4. Click **Save and Close**.



Note: When you save a Custom Poller Collection configuration, each associated Policy changes to Active State "Suspended". To make a Policy active, access the Custom Poller Configuration: Policies tab, open each associated Policy, and change the Active State to "Active".

Global Control

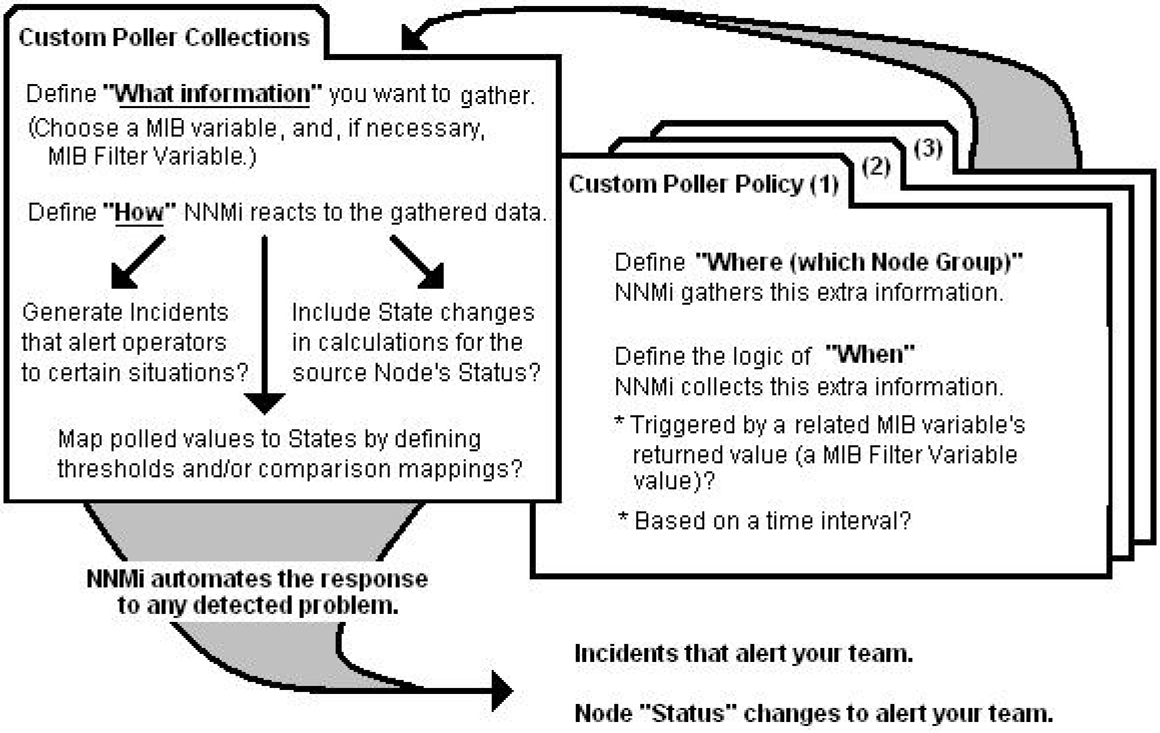
Enable Custom Poller ☑

Step 2: Create a Custom Poller Collection

After you enable Custom Poller, you are ready to 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.

In addition to a Custom Poller Collection, you should define at least one Custom Poller Policy. Each policy specifies the Node Group on which you apply the Custom Poller Collection.

A diagram of the hierarchy of Custom Poller Collections is depicted below:


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.

For each Custom Poller Collection, you specify the following:

- A meaningful name for the Custom Poller Collection configuration. The Custom Poller Collection name appears in any incidents generated for the collection. Specify a name that will help you to identify the MIB information being polled.
- Whether you want each Polled Instance to affect the Node Status for the associated Node Group.
- Whether you want NNMi to generate an incident based on a threshold or based on a specified polled value or value range.
- The MIB variable information, including the following:
  - The MIB variable that you want to poll (MIB Poll Variable)
  - The MIB variable you want to use as a filter (MIB Filter Variable) to determine which instances of the collected MIB Poll Variable will result in NNMi collecting additional information

For our example, we are choosing a MIB Poll Variable that has multiple instances. This is because a Windows PC can have multiple disks. Therefore, the MIB variables containing disk information has multiple instances, one for each disk.

Note:

If you choose a MIB Poll Variable that has multiple instances, you MUST specify a MIB Filter Variable and MIB Filter. (The MIB Filter is defined in the Policy.)

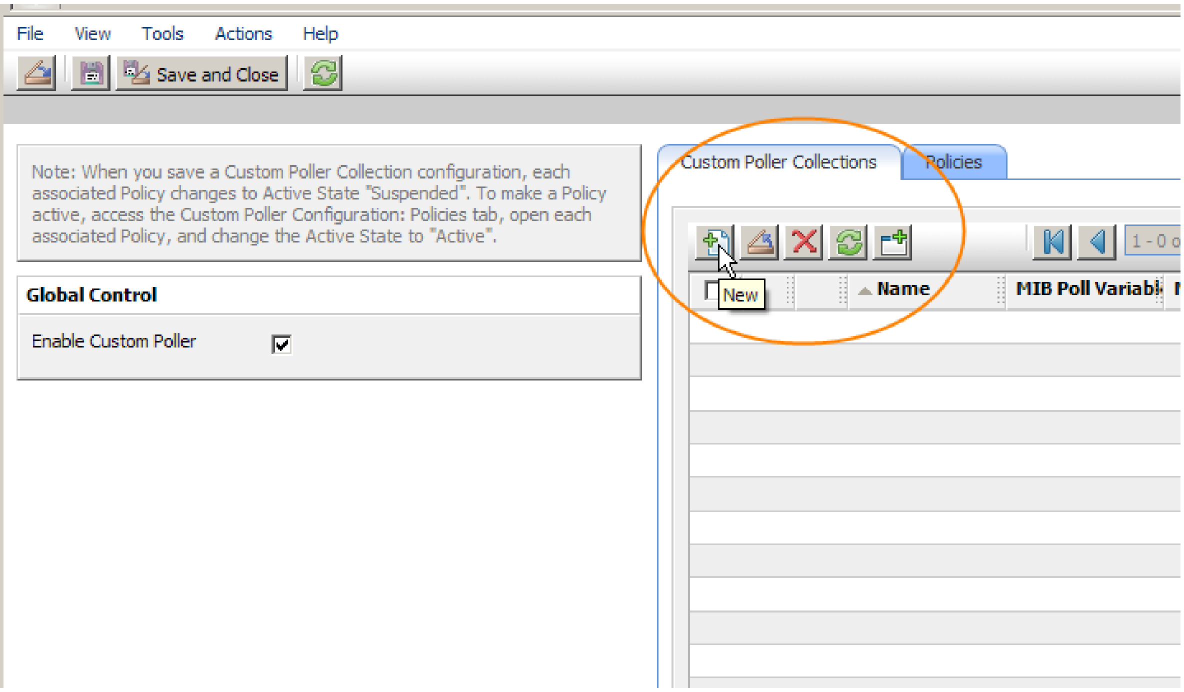
In our example, we are required to provide a MIB Filter value to select the disks we want NNMi to monitor. If we do not specify a MIB Filter Variable and MIB Filter, NNMi assumes the MIB variable does not have multiple instances.

To create our Custom Poller Collection:

1. Open the same **Custom Poller Configuration** form as described in Step 1:

   a. Navigate to the **Configuration** workspace.

   b. Select **Custom Poller Configuration**.
2. Navigate to the **Custom Poller Collections** tab.
3. Click the **New** icon.



Note: When you save a Custom Poller Collection configuration, each associated Policy changes to Active State "Suspended". To make a Policy active, access the Custom Poller Configuration: Policies tab, open each associated Policy, and change the Active State to "Active".

Global Control

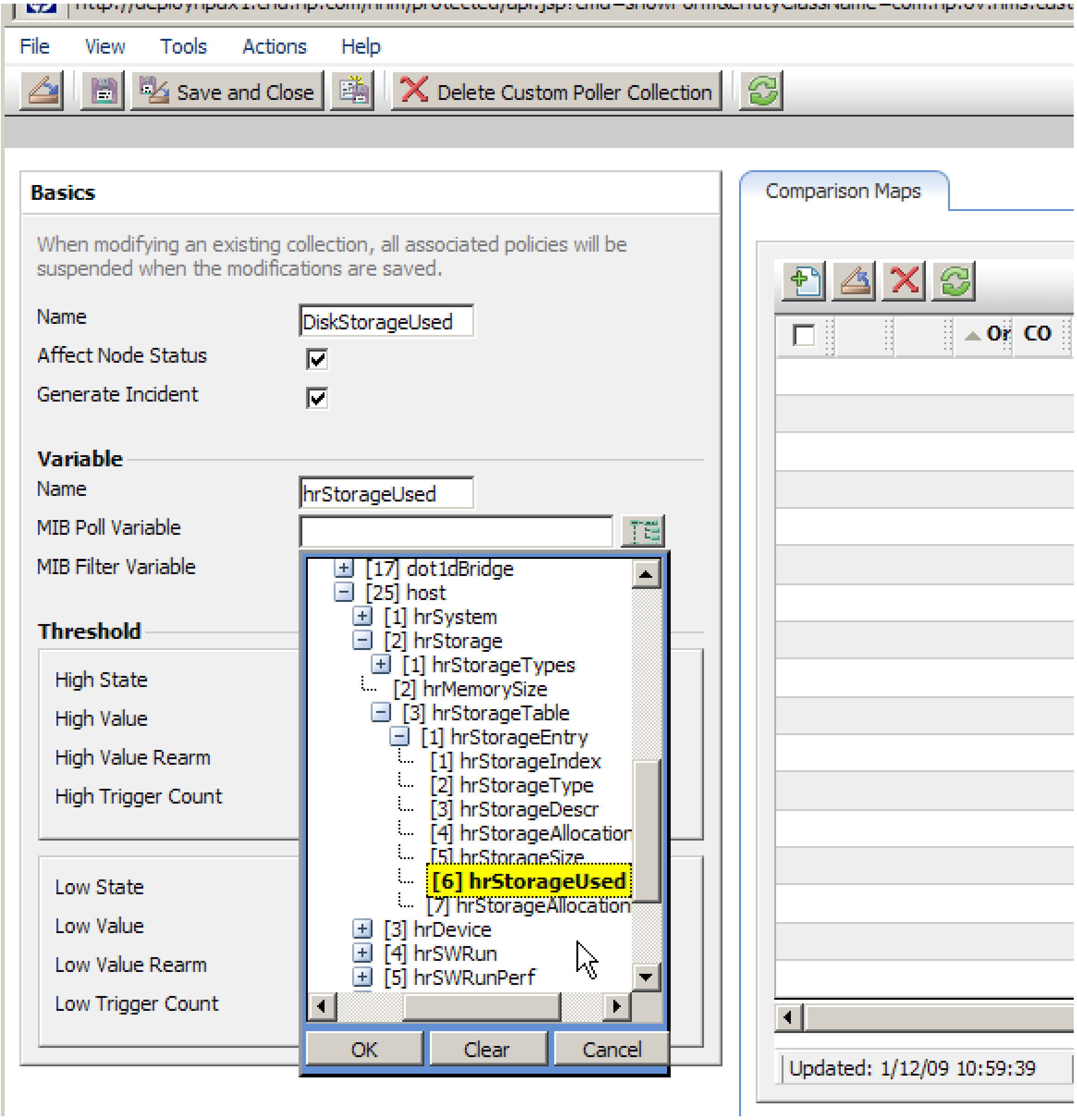
Enable Custom Poller ☑

4. In the Name attribute of the **Custom Poller Collections** form, name the Collection **DiskStorageUsed**.
5. Click to check **Affect Node Status** and **Generate Incident**.

Next, we need to specify the MIB Poll Variable. We are going to use the **hrStorageUsed** variable. You need to know where this variable is located in the MIB tree.

Note:
You cannot type in the OID directly. You must navigate the tree and select the variable as shown in the screen capture that follows.

6. In the **MIB Poll Variable** attribute, use the MIB tree to select **hrStorageUsed**.



Basics

When modifying an existing collection, all associated policies will be suspended when the modifications are saved.

Name: DiskStorageUsed

Affect Node Status ☑

Generate Incident ☑

Variable

Name: hrStorageUsed

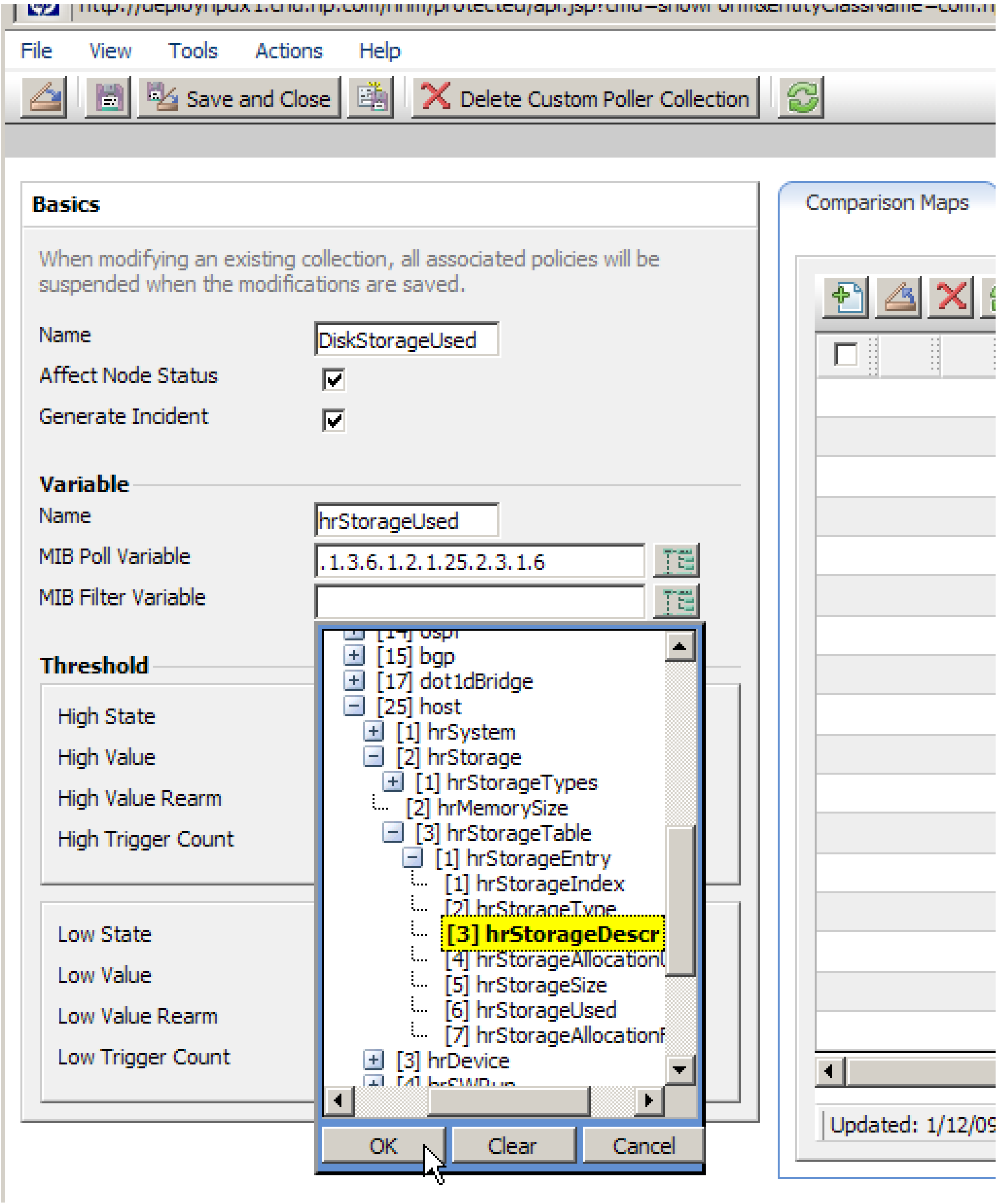
MIB Poll Variable:

MIB Filter Variable

Updated: 1/12/09 10:59:39

We also need to set up a MIB Filter Variable because there are multiple instances (one for each disk). We will filter against hrStorageDescr to select the correct instance.

7. In the **MIB Filter Variable** attribute, use the MIB tree to select **hrStorageDescr**. We define the filter itself later when creating the Custom Poller Policy.



Basics

When modifying an existing collection, all associated policies will be suspended when the modifications are saved.

Name: DiskStorageUsed

Affect Node Status: ☑

Generate Incident: ☑

Variable

Name: hrStorageUsed

MIB Poll Variable: .1.3.6.1.2.1.25.2.3.1.6

MIB Filter Variable:

Threshold

High State

High Value

High Value Rearm

High Trigger Count

Low State

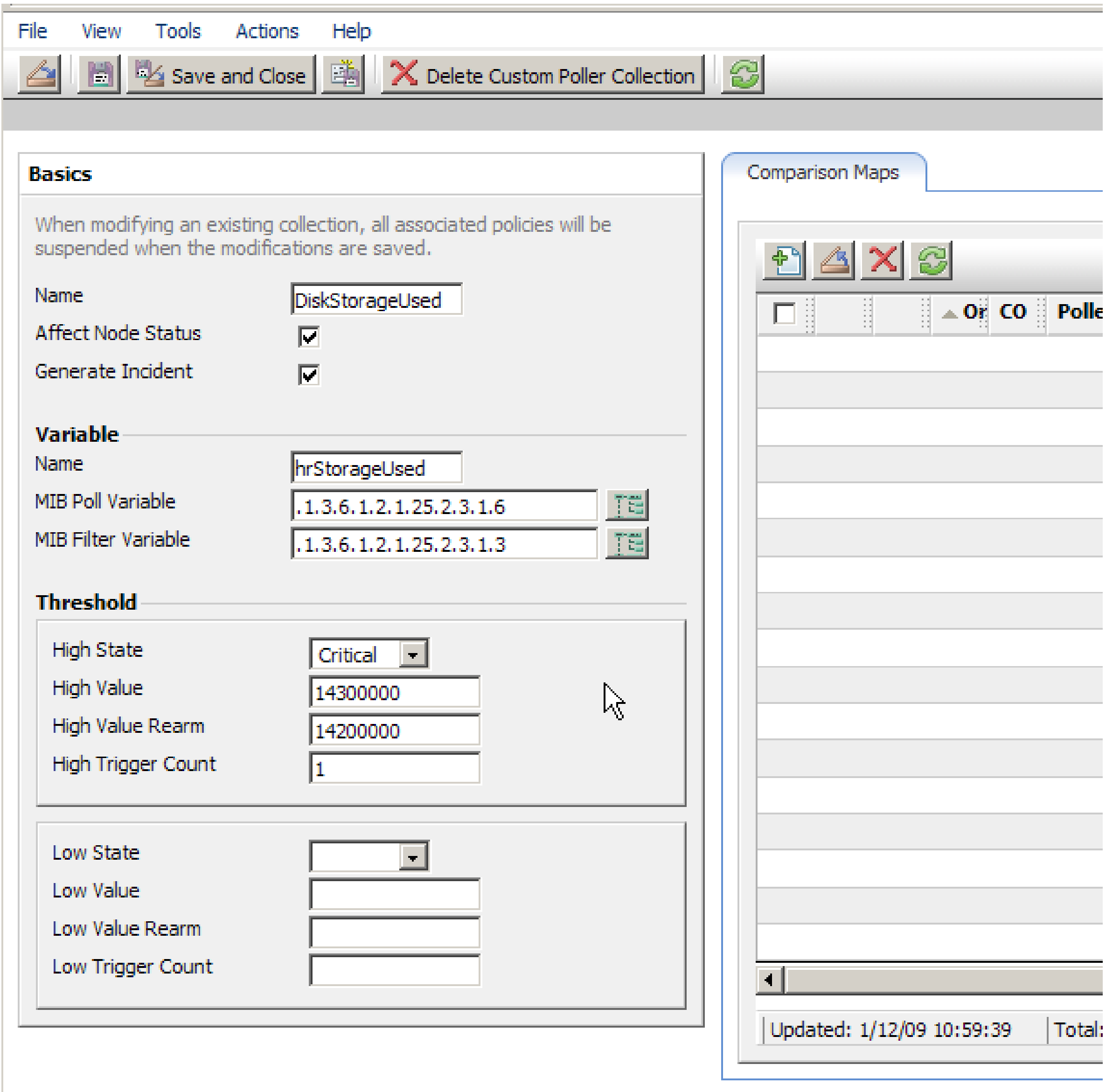
Low Value

Low Value Rearm

Low Trigger Count

Finally, we set up a High Value for the threshold. As shown in the example screen capture below, make this value slightly above the current disk space usage on the target PC, so we can easily cross the threshold by copying a large file on to the disk.

8. In the **High State** attribute, select **Critical** from the drop-down list.
9. In the **High Value** attribute, enter **14300000**.
10. In the **High Value Rearm** attribute, enter **14200000**.
11. In the **High Trigger Count** attribute, accept the default value of **1**.



Basics

When modifying an existing collection, all associated policies will be suspended when the modifications are saved.

Name	DiskStorageUsed
Affect Node Status	☑
Generate Incident	☑

Variable

Name	hrStorageUsed
MIB Poll Variable	.1.3.6.1.2.1.25.2.3.1.6
MIB Filter Variable	.1.3.6.1.2.1.25.2.3.1.3

Threshold

High State	Critical
High Value	14300000
High Value Rearm	14200000
High Trigger Count	1
Low State	
Low Value	
Low Value Rearm	
Low Trigger Count	

Updated: 1/12/09 10:59:39 | Total:

12. Click **Save and Close**.

We now have a Custom Poller Collection defined.

Note:
For the purposes of this example, we did not need to define any Comparison Maps for. You want to configure Comparison Maps when you use variables that place meaning on specific values.

Step 3: Create a Policy for a Custom Poller Collection

A Policy defines which Node Groups are participating in this Custom Poller Collection and how often the variable is polled. It is also how we define the MIB filter to select specific instances. You can create more than one Policy associated with a Custom Poller Collection.

Begin by naming the Custom Poller Policy.

1. In the Name attribute, enter **WindowsUsedDiskSpace**.

NNMi displays this name in the incident browser, whenever an associated incident is generated for the Custom Poll.

Next specify the order.

Because you can create more than one Policy associated with a Custom Poller Collection, you need to specify an order to remove ambiguity when a node is matched to more than one policy.

For example, you might want a Custom Poller Collection to run against a Node Group for Routers and a Node Group for Switches. You also might want to poll the Routers every 5 minutes and the switches every 8 minutes. If a node is both a switch and a router, then it is not allowed to be polled twice by the same collection. You might decide that if a node is both a switch and a router, it is important that it be polled more quickly (using the Router policy). Therefore, you use a higher priority order number for the Router Policy. When you specify an order, the node is matched against the highest priority policy (lowest Ordering number).

Because we have only one Policy for a Collection, we do not need to be concerned about this order.

2. In the Ordering attribute, accept the default value of **1**.
3. Use the Quick Find pull down menu to select the Collection that we previously defined. Change the Active State to **Active**.
4. Use the Quick Find pull down menu to select the **Microsoft Windows Systems** Node Group.

Next, specify the MIB Filter.

5. Recall that the values from our SNMP walk, the C drive always started with *C:*. Therefore, in the MIB Filter attribute, enter **C:*** as the filter.

Note:
Avoid matching all instances by entering the asterisk (*) as the filter. This could lead to a large number of matches and adversely affect NNMi Custom Poller performance.

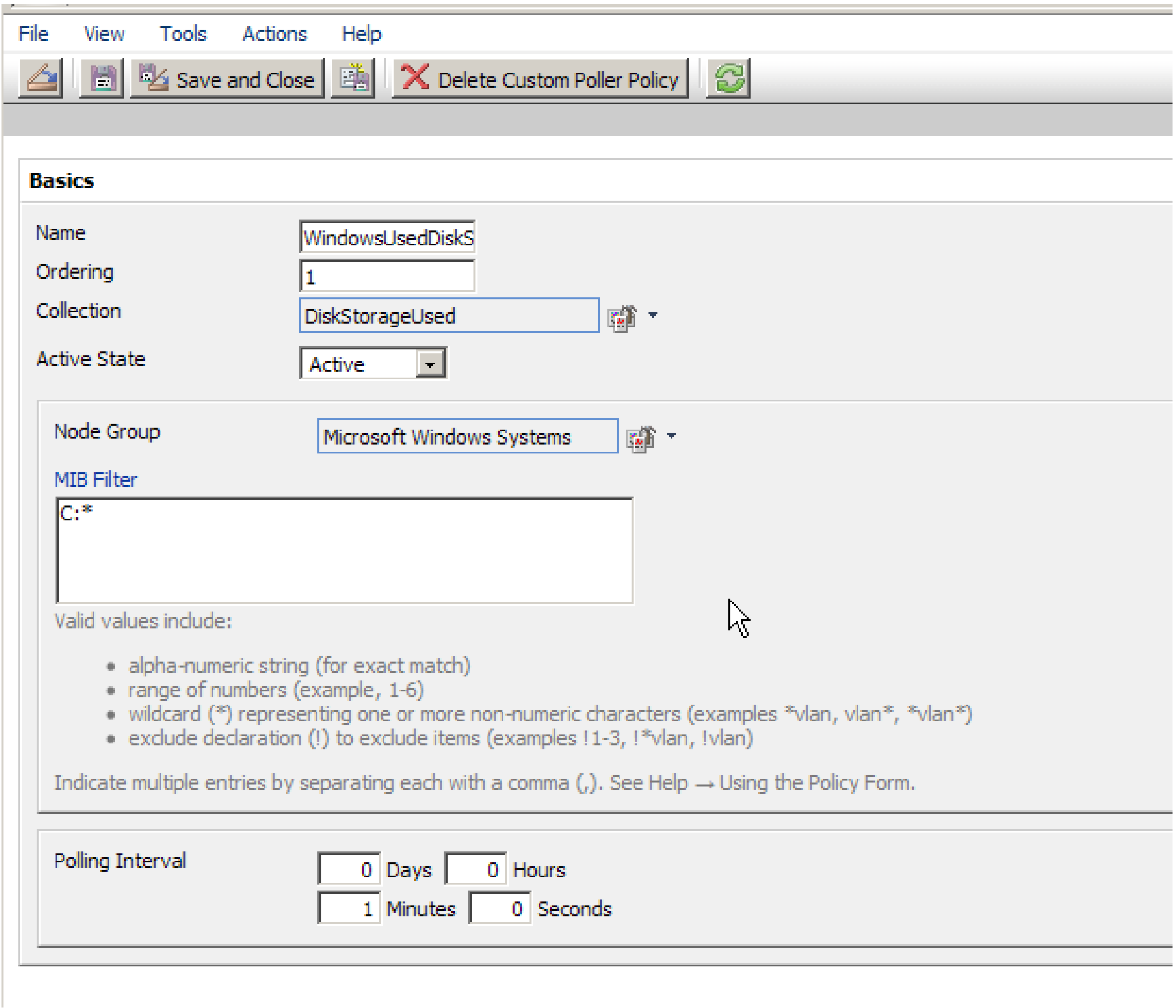
Finally, we specify the Polling Interval.

In NNMi, you cannot force a Custom Poll. (The Status Poll or Configuration Poll options from the Actions menu do not include the Custom Poller.) Therefore, while initially setting up your Custom Polls, you might want to set the Polling Interval fairly short, so you do not have to wait a long time for each poll cycle.

6. For the purposes of this example, set the Polling Interval attribute to **1** minute.

Note:
As a best practice, after you have set up your Custom Polls as desired, select a longer Polling Interval. Using short Polling Intervals can adversely affect NNMi Custom Poller performance.

7. Click **Save and Close**.



Basics

Name: WindowsUsedDiskS

Ordering: 1

Collection: DiskStorageUsed

Active State: Active

Node Group: Microsoft Windows Systems

MIB Filter: C:*

Valid values include:

- alpha-numeric string (for exact match)
- range of numbers (example, 1-6)
- wildcard (*) representing one or more non-numeric characters (examples *vlan, vlan*, *vlan*)
- exclude declaration (!) to exclude items (examples !1-3, !*vlan, !vlan)

Indicate multiple entries by separating each with a comma (,). See Help → Using the Policy Form.

Polling Interval: 0 Days 0 Hours 1 Minutes 0 Seconds

Note:
If you make a change to a Custom Poller Collection configuration after you save it (for example, change the High Value), NNMi automatically changes the Active State of the associated Custom Poller Policies to **Suspended**. If this occurs, open the Policy configuration and change the Active State back to **Active**.

View the Results of Your Custom Poll

Step 1: View the Node Collections Associated with Custom Poller Policies

After you configure your Custom Poller Collections, you can view the Custom Poller objects. NNMi identifies these objects as Custom Node Collections (not to be confused with Custom Poller Collections).

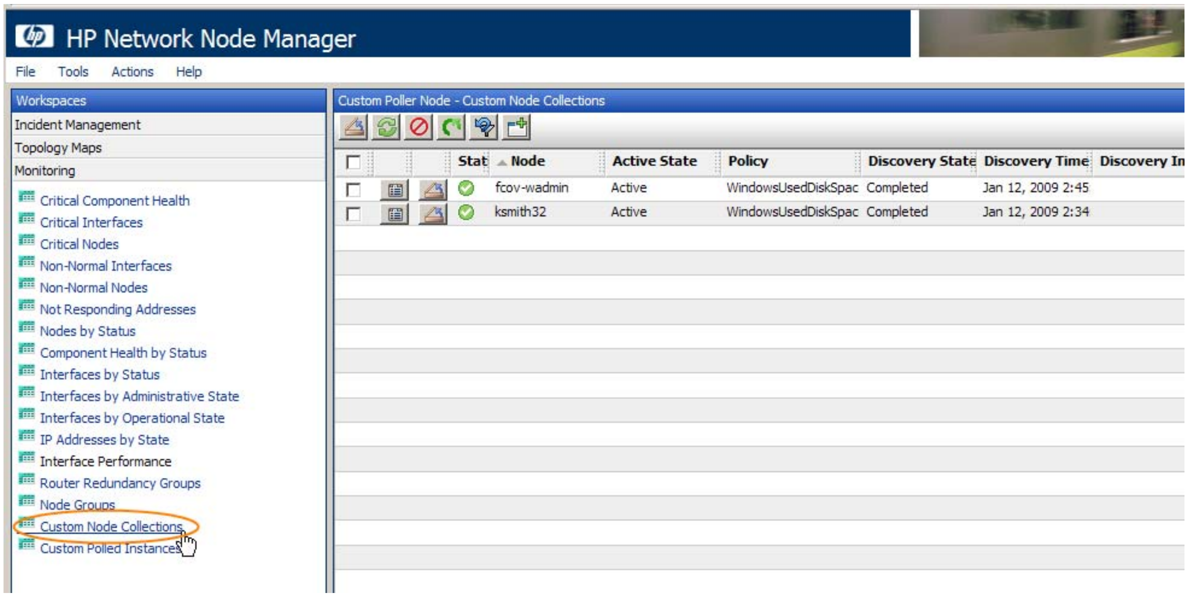
To view Custom Node Collections:

1. Navigate to the **Monitoring** workspace.
2. Select **Custom Node Collections**.

NNMi displays a table view of all Custom Node Collections that includes:

- The Custom Node Collection Status
- The topology node associated with the Custom Node Collection
- The Active State for the associated Policy
- The name of each Policy associated with each Custom Node Collection
- Discovery information regarding the MIB Poll Variable on each topology node, such as Discovery State, the Discovery Time Stamp, and Discovery State Information

As shown in the screen capture below, the NNMi database has two Windows PCs.



Custom Poller Node - Custom Node Collections

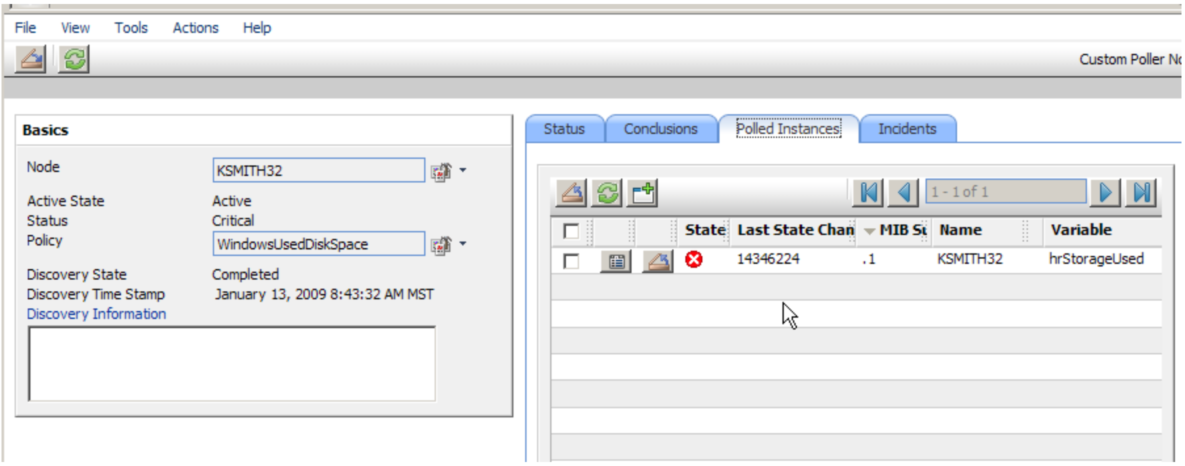
□			Stat	Node	Active State	Policy	Discovery State	Discovery Time	Discovery In
□			✓	fcov-wadmin	Active	WindowsUsedDiskSpac	Completed	Jan 12, 2009 2:45	
□			✓	ksmith32	Active	WindowsUsedDiskSpac	Completed	Jan 12, 2009 2:34	

Note:
The same node name can be listed in the Custom Node Collections view multiple times if it has multiple Custom Poller Collections applied to it. These are not “Nodes” but “Node Collections”.

Step 2: View Details of a Custom Node Collection

To view the details for a specific Custom Node Collection, click the Open icon that precedes the Node Collection of interest.

As shown in the example below, you can see the Status history, Conclusions, Polled Instances and any incidents that have been generated.



Basics

Node: KSMITH32
Active State: Active
Status: Critical
Policy: WindowsUsedDiskSpace
Discovery State: Completed
Discovery Time Stamp: January 13, 2009 8:43:32 AM MST
Discovery Information

State	Last State Chan	MIB Su	Name	Variable
⊗	14346224	.1	KSMITH32	hrStorageUsed

Step 3: View Details of a Polled Instance

Another useful view is the Custom Polled Instances. The first time the specified MIB variable is discovered, the results appear in a Polled Instance object. The Polled Instance object is updated whenever a change in the MIB Poll Variable's State is detected and includes the most recent polled value that caused the State to change.

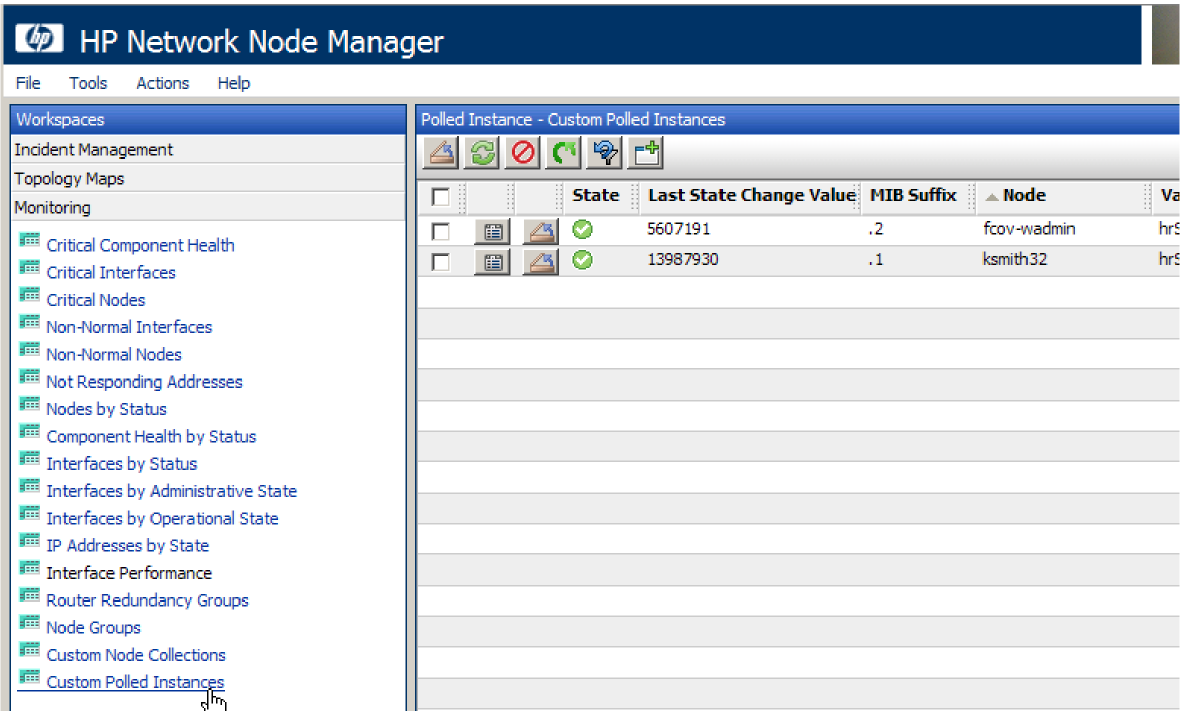
To view Custom Polled Instances:

1. Navigate to the **Monitoring** workspace.
2. Select **Custom Polled Instances**.

This view allows you to easily see all the Polled Instances that are polled by a specific Custom Poller Collection. For example, you can sort the view based on The MIB Variable to see all the Polled Instances for a particular Custom Poller Collection.

This table does not include the Custom Poller Collection Name. However, it lists the names of the MIB Variable being polled. Each collection has only one MIB Poll Variable. Therefore, if you use a unique name for your Custom Poller Collection variables, it is easy to associate the Custom Poller Collection with the MIB Poll Variable.

As shown in the example below, another server has the "C" drive mapped to MIB Suffix (or instance) .2 rather than .1.



□			State	Last State Change Value	MIB Suffix	Node	Va
□			✓	5607191	.2	fcov-wadmin	hrS
□			✓	13987930	.1	ksmith32	hrS

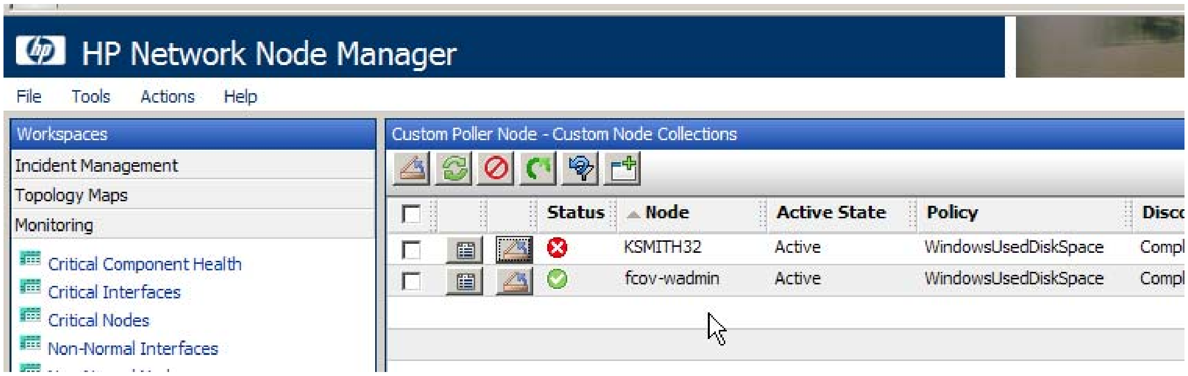
Note:
The value in **Last State Change Value** is not equivalent to the last measured value. Instead, this is the value that was measured when the State last changed.

Step 4: Evaluate the Results of the Custom Poll

To evaluate the results of our example Custom Poll, we trigger the threshold to see the changed State and the generated incident.

1. To do so, copy a few large files onto the disk of the PC to increase the disk usage.

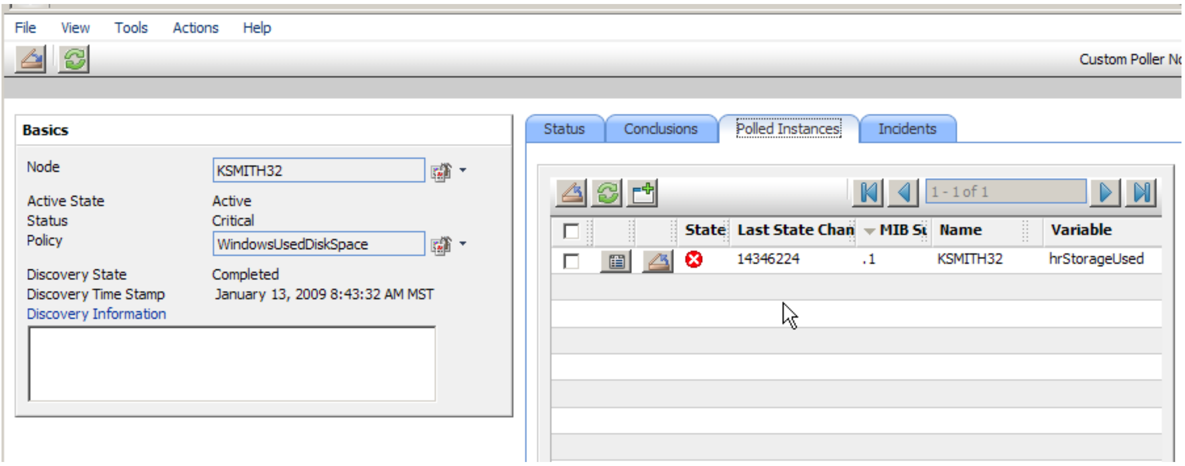
As shown in the example below, copying a few large files onto the disk of the PC causes the Status of the Custom Node Collection to change to Critical.



□			Status	Node	Active State	Policy	Disc
□			✖	KSMITH32	Active	WindowsUsedDiskSpace	Compl
□			✓	fcov-wadmin	Active	WindowsUsedDiskSpace	Compl

Note:
The Custom Node Collection Status is not necessarily equivalent to the Status of the source node.

2. Open the Custom Poller Collection to see that the value of hrStorageUsed is above the High Value threshold. (It may be even higher now, but this is the value that triggered the State change.)



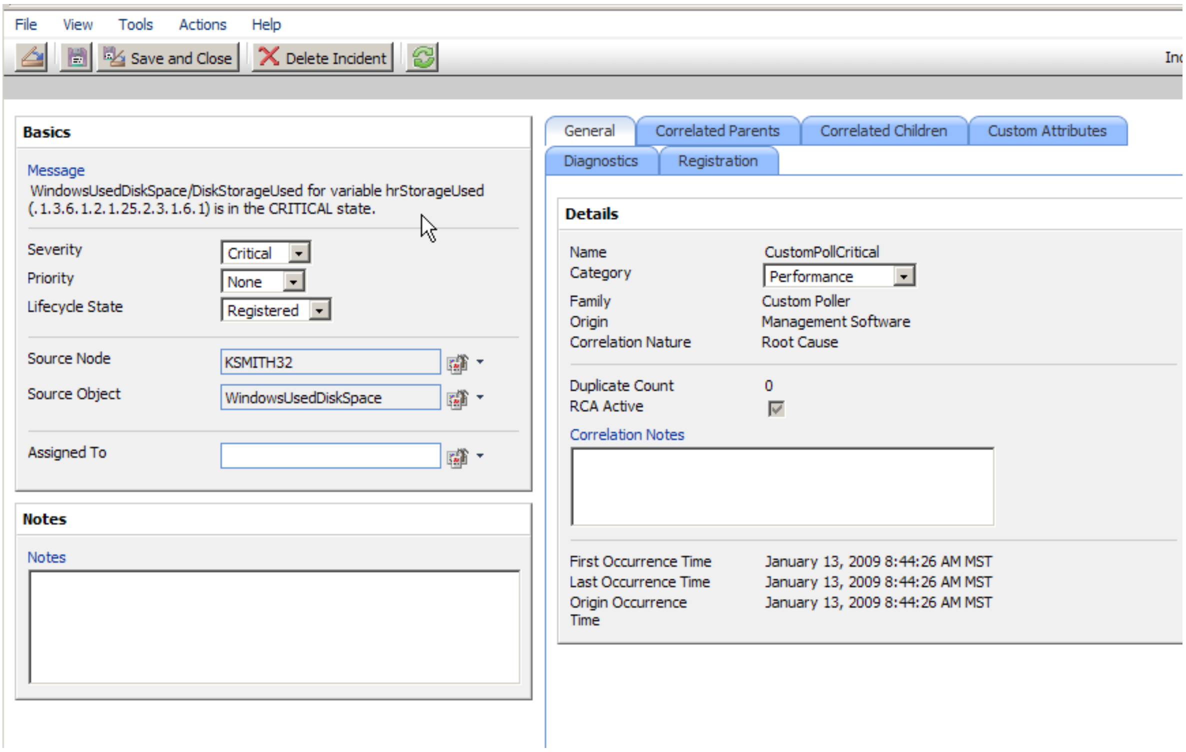
Basics

Node	KSMITH32
Active State	Active
Status	Critical
Policy	WindowsUsedDiskSpace
Discovery State	Completed
Discovery Time Stamp	January 13, 2009 8:43:32 AM MST

Discovery Information

State	Last State Chan	MIB Su	Name	Variable
☒	14346224	.1	KSMITH32	hrStorageUsed

3. Navigate to the Incidents tab to see that an incident was generated.



Basics

Message
WindowsUsedDiskSpace/DiskStorageUsed for variable hrStorageUsed (.1.3.6.1.2.1.25.2.3.1.6.1) is in the CRITICAL state.

Severity	Critical
Priority	None
Lifecycle State	Registered
Source Node	KSMITH32
Source Object	WindowsUsedDiskSpace
Assigned To	

Notes

Notes

Details

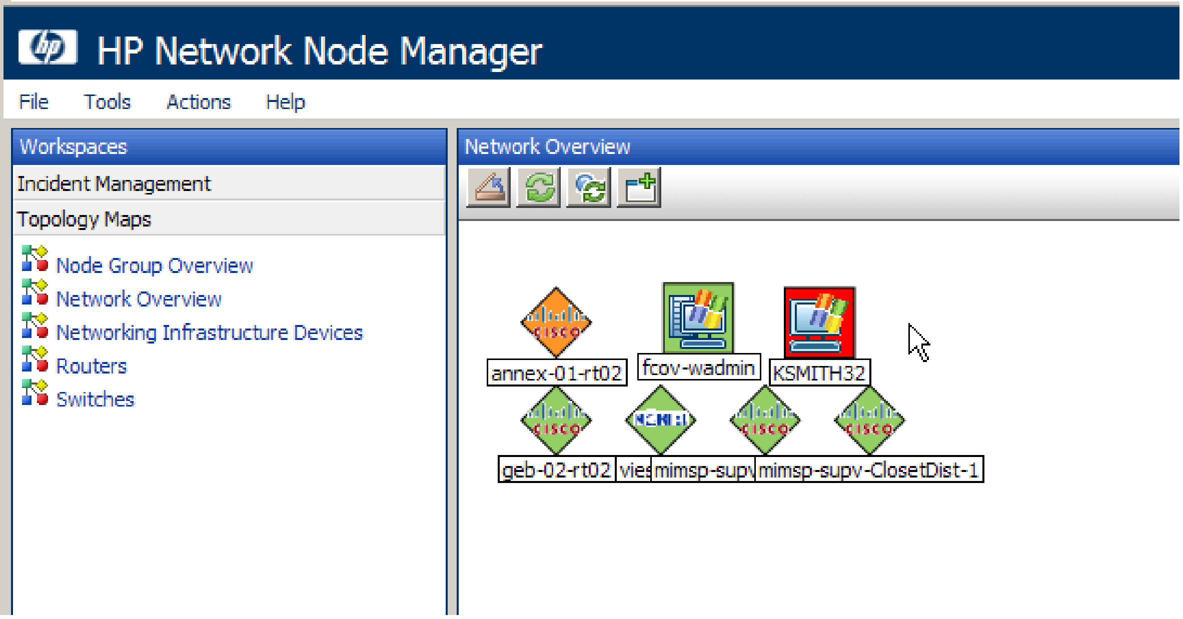
Name	CustomPollCritical
Category	Performance
Family	Custom Poller
Origin	Management Software
Correlation Nature	Root Cause
Duplicate Count	0
RCA Active	☑

Correlation Notes

First Occurrence Time	January 13, 2009 8:44:26 AM MST
Last Occurrence Time	January 13, 2009 8:44:26 AM MST
Origin Occurrence Time	January 13, 2009 8:44:26 AM MST

The Incident message presents the various names used in creating the Custom Poller Collection. You can change this message by editing the Incident Configuration. (From **Configuration→Incident Configuration,** select **Management Event Configuration** and then open the **CustomPollCritical** incident configuration.) For example, you might want to include the Custom Attribute "cia.LastValue" in your message to display the most recent value that caused the Custom Node Collection Status to change. To view the listing of possible Custom Attributes, open a Custom Poller incident and select the **Custom Attributes** tab.

4. Open the Source Node or select a Node View or Map, to verify that the Status of the source node has changed to Critical.

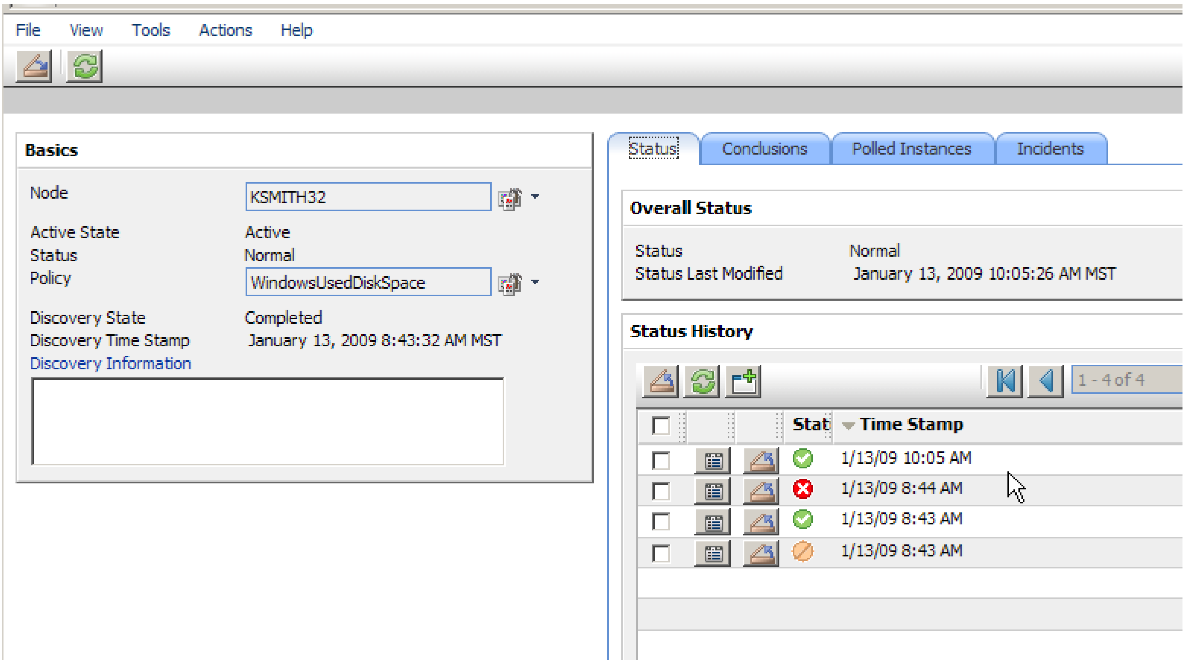


We have verified that the Custom Poll was successful and properly notified us that the disk space is Critical.

Next, we return the disk to its previous State.

5. Delete the large file from the PC.

6. Verify the Custom Node Collection's Status has returned to Normal, by opening the Custom Node Collection form and navigating to the **Status** tab.



Basics

Node	KSMITH32
Active State	Active
Status	Normal
Policy	WindowsUsedDiskSpace
Discovery State	Completed
Discovery Time Stamp	January 13, 2009 8:43:32 AM MST
Discovery Information	

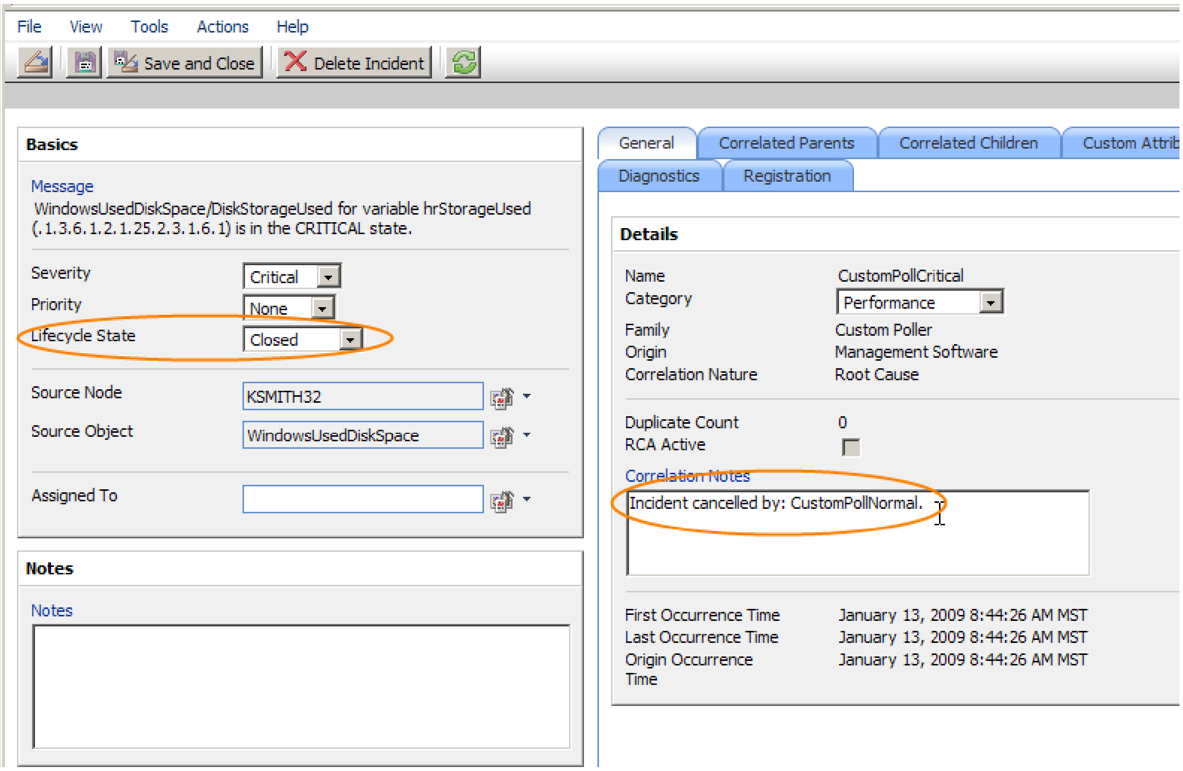
Overall Status

Status	Normal
Status Last Modified	January 13, 2009 10:05:26 AM MST

Status History

Stat	Time Stamp
✓	1/13/09 10:05 AM
✗	1/13/09 8:44 AM
✓	1/13/09 8:43 AM
⊘	1/13/09 8:43 AM

7. Verify that the incident has been closed by returning to the Incident form for the generated incident.



Basics

Message
WindowsUsedDiskSpace/DiskStorageUsed for variable hrStorageUsed (.1.3.6.1.2.1.25.2.3.1.6.1) is in the CRITICAL state.

Severity	Critical
Priority	None
Lifecycle State	Closed
Source Node	KSMITH32
Source Object	WindowsUsedDiskSpace
Assigned To	

Notes

Notes

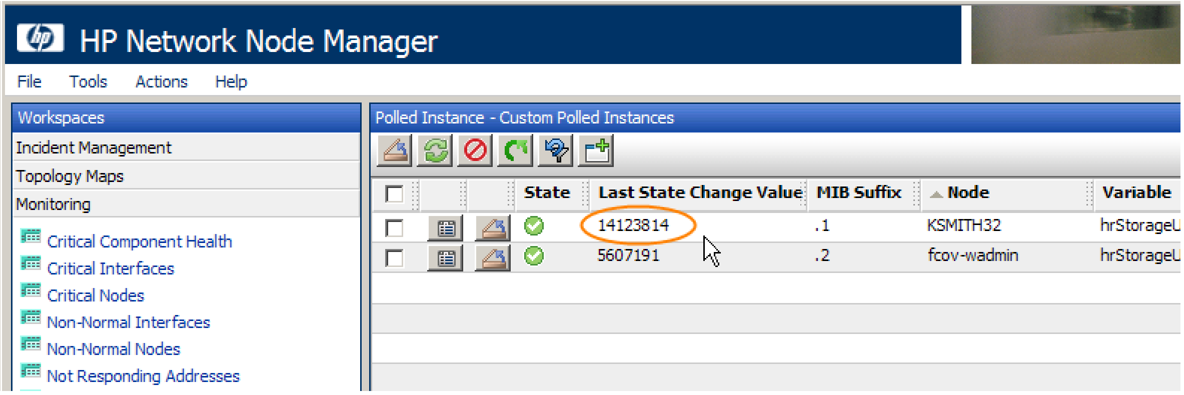
Details

Name	CustomPollCritical
Category	Performance
Family	Custom Poller
Origin	Management Software
Correlation Nature	Root Cause
Duplicate Count	0
RCA Active	□

Correlation Notes
Incident cancelled by: CustomPollNormal.

First Occurrence Time	January 13, 2009 8:44:26 AM MST
Last Occurrence Time	January 13, 2009 8:44:26 AM MST
Origin Occurrence Time	January 13, 2009 8:44:26 AM MST

8. Last, return to the Custom Polled Instances view to verify the value of the MIB Poll Variable is below the High State threshold.



Polled Instance - Custom Polled Instances

□			State	Last State Change Value	MIB Suffix	Node	Variable
□			✓	14123814	.1	KSMITH32	hrStorageU
□			✓	5607191	.2	fcov-wadmin	hrStorageU

The next section presents some tips for Troubleshooting the NNMi Custom Poller.

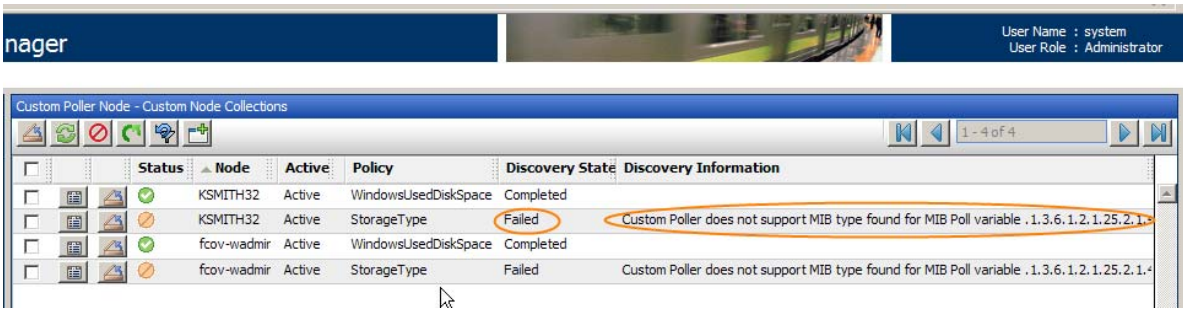
Troubleshooting Tips

NNMi provides feedback on common errors. This section describes some common types of feedback.

For Custom Poller Collections, NNMi supports queries of the following types (as defined in the MIB). Watch for possible aliases on the types.

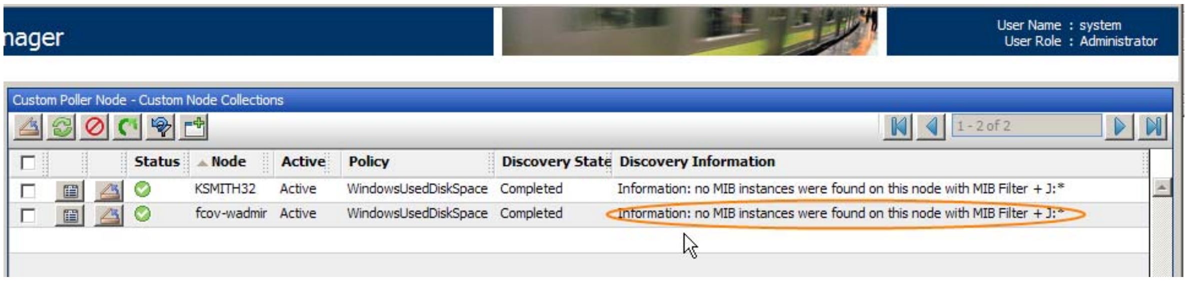
Supported MIB Poll Variable Type	Supported MIB Filter Variable Type
INTEGER, Integer32	INTEGER, Integer32
Unsigned32	Unsigned32
Counter, Counter32, Counter64	Gauge, Gauge32
Gauge, Gauge32	OCTET STRING
TimeTicks	IpAddress
OCTET STRING	

If you set up a Custom Poller Collection using a MIB Poll Variable or MIB Filter Variable of an unsupported type, NNMi displays an error in the Discovery State. NNMi also provides some additional information about the failure in the Discovery Information attribute.


nager
User Name : system
User Role : Administrator
Custom Poller Node - Custom Node Collections
1 - 4 of 4

☐			Status	Node	Active	Policy	Discovery State	Discovery Information
☐			✓	KSMITH32	Active	WindowsUsedDiskSpace	Completed	
☐			⊘	KSMITH32	Active	StorageType	Failed	Custom Poller does not support MIB type found for MIB Poll variable .1.3.6.1.2.1.25.2.1.
☐			✓	fcov-wadmir	Active	WindowsUsedDiskSpace	Completed	
☐			⊘	fcov-wadmir	Active	StorageType	Failed	Custom Poller does not support MIB type found for MIB Poll variable .1.3.6.1.2.1.25.2.1.

If no Polled Instances pass the defined filter, you also see a notification in the Discovery Information attribute.


nager
User Name : system
User Role : Administrator
Custom Poller Node - Custom Node Collections
1 - 2 of 2

☐			Status	Node	Active	Policy	Discovery State	Discovery Information
☐			✓	KSMITH32	Active	WindowsUsedDiskSpace	Completed	Information: no MIB instances were found on this node with MIB Filter + J:*
☐			✓	fcov-wadmir	Active	WindowsUsedDiskSpace	Completed	Information: no MIB instances were found on this node with MIB Filter + J:*

The Custom Poller log file in the following directory contains some useful debug information:
`/var/opt/OV/log/nnm`

January 2009

hp®
invent

Java™
COMPATIBLE