



HP Network Node Manager i Software

Step-by-Step Guide to Custom Poller

NNMi 9.1x Patch 2

This document includes two examples. The first example illustrates how to use Custom Poller to monitor disk space. The second example demonstrates the use of Custom Poller to detect reboots of devices. The second example builds on knowledge learned in the first example.

Note the following:

- The Custom Poller is only available with NNMi 8.11 or greater.
- This example uses a UNIX NNMi management server, but it can be followed on a Windows server.
- Some screen captures might be slightly different than those that appear in the NNMi graphical user interface.
- In NNMi 9.00, if you are logged in a “root” (UNIX) or “administrator” (Windows), then no user name or password is required for most command line tools.

CONTENTS

Monitoring Disk Space Example 3

 Setting Up Your MIB 3

 Step 1: Identify the MIB Variable You Want to Poll 3

 Step 2: Ensure the MIB Includes Supported Types 3

 Step 3: Load the Required MIB 4

 Step 4: Use the MIB Browser to View Current MIB Variable Values 5

 Setting Up a Custom Poll 6

 Step 1: Enable Custom Poller 6

 Step 2: Create a Custom Poller Collection 8

 Step 3: Create a Policy for a Custom Poller Collection 15

 Viewing the Results of Your Custom Poll 17

 Step 1: View the Node Collections Associated with Custom Poller Policies 17

 Step 2: View Details of a Polled Instance 18

 Step 3: Evaluate the Results of the Custom Poll 19

 Exporting the Custom Poller Collection 23

Detecting Reboots Example 25

 Step 1: Create a Custom Poller Collection 25

 Step 2: Create a Policy for a Custom Poller Collection 30

 Step 3: Test the Policy 32

Troubleshooting Tips 34

Monitoring Disk Space Example

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.

This example monitors the disk usage on Microsoft PCs using the `rfc2790-HOST-RESOURCES-MIB`. This MIB is shipped with NNMi under the following directory:

```
%NnmInstallDir%\misc\nnm\snmp-mibs\Standard
```

The `%NnmInstallDir%` location depends on your operating system:

Windows 2008

```
%NnmInstallDir% =  
<drive>\Program Files(x86)\HP\HP BTO Software\
```

UNIX

```
$NnmInstallDir = /opt/OV/
```

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

- 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 are using. This is especially important because the variables used must have a type that NNMi supports. See “Troubleshooting Tips,” later in this document, for a list of supported MIB variables.

1. First, check whether the MIB is loaded. From the workspace navigation panel, select the **Configuration** workspace, expand the **MIBs** folder, and then click **Loaded MIBs**. The `rfc2790-HOST-RESOURCES-MIB` should appear in the Loaded MIBs view. Because the MIB is loaded in NNMi, you can study the MIB using the Loaded MIBs view:
 - a. Click the MIB you want to view.
 - b. Click **Actions > Display MIB File**.
2. Next, check the MIBs that are available to load. If the MIB is available to load, you can study the view using the Unloaded MIBs tables.
 - a. Select **Tools > Load/Unload MIB...**
 - b. Look for the MIB in the Unloaded MIBs tables.
 - c. If the MIB is listed, click **Display** in the **Actions** column that appears next to the MIB name.

Note: To study the MIB, you can also use the HP tool `nmmsnmpwalk.ovpl` or read through the MIB with a text editor.

September, 2011

Consider an excerpt from the `rfc2790-HOST-RESOURCES-MIB`:

```
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`. `hrStorageUsed` and `hrStorageAllocationUnits` are of type `Integer32`. The NNMi Custom Poller supports both of these types.

According to the MIB definition, `hrStorageSize` is the size of the storage measured in `hrStorageAllocationUnits`. To show the amount of storage in kilobytes (KB) on the C drive, this example uses the following MIB expression:

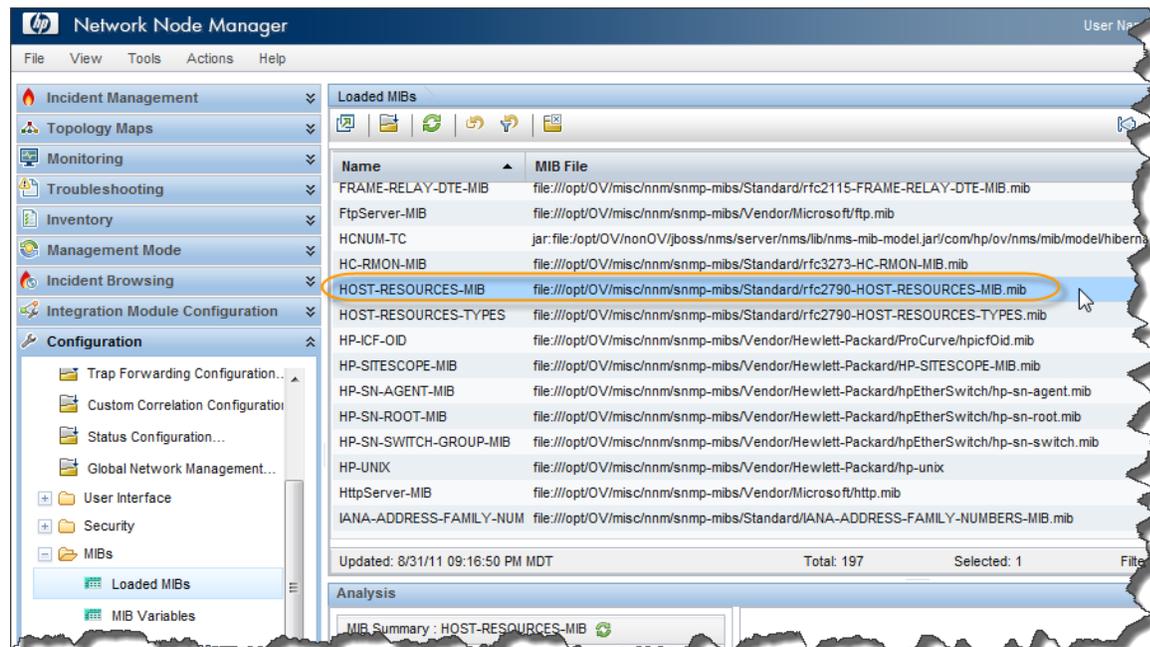
```
((hrStorageSize / 1000) * hrStorageAllocationUnits)
```

Step 3: Load the Required MIB

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

Using the Loaded MIBs view described in "Step 2: Ensure the MIB Includes Supported Types", we can determine that the `rfc2790-HOST-RESOURCES-MIB` is loaded in NNMi as shown in the following figure:

Figure 1: Loaded MIBs



Tip: To check whether the MIB is already loaded, you can also run the `nnmloadmib.ovpl -list` command. Look for the desired MIB in the results.

September, 2011

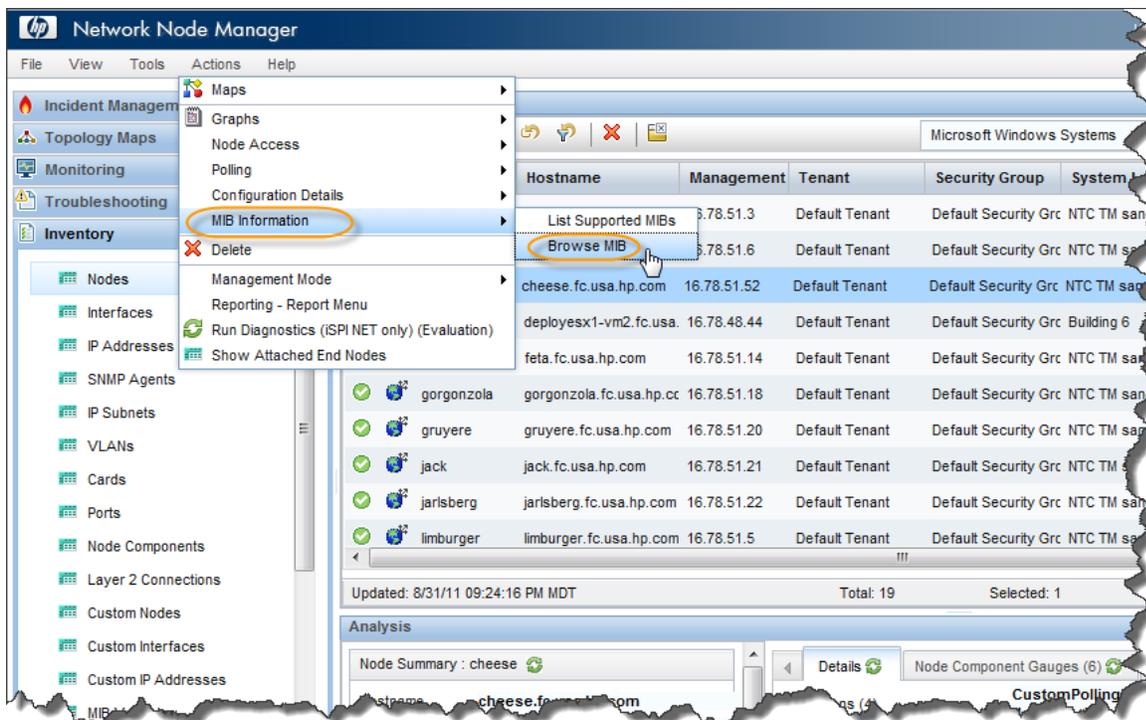
If the MIB had not been loaded, you can load it with the **Tools > Load /Unload MIB...** pull-down menu or the `nnmloadmib.ovpl -load` command.

Step 4: Use the MIB Browser to View Current MIB Variable Values

Next, use the MIB Browser to perform an SNMP query and become familiar with the MIB variable values returned from the node. In this example, the node is a Windows PC server.

1. From the workspace navigation panel, select the **Inventory** workspace and then click **Nodes**.
2. Select a node that has a Device Category of Server (🌐).
3. Click **Actions > MIB Information > Browse MIB**.

Figure 2: Browse MIB



In the **Node** attribute you should see the name of the node you selected.

4. In the **OID** attribute, enter `hrStorageTable`.
5. Click the ▶ Start SNMP Walk icon.

September, 2011

Figure 3: SNMP Walk

OID	Value
hrStorageTable.hrStorageEntry	
hrStorageIndex	
hrStorageType	
hrStorageDescr	
1	A:\
2	C:\ Label: Serial Number e03658ce
3	D:\
4	Virtual Memory
5	Physical Memory
hrStorageAllocationUnits	
1	0
2	4096
3	0
4	65536
5	65536
hrStorageSize	
hrStorageUsed	
1	0
2	8990618
3	0
4	83737
5	75806
hrStorageAllocationFailures	

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 previous figure, one of the strings in the Value column begins with C:\. You can see the value for hrStorageAllocationUnits is 4096 on this drive. The hrStorageUsed value is 8990618.

Next, you need to enable Custom Poller so that you can use it to specify the MIB Expression you want NNMI to poll.

Setting Up a Custom Poll

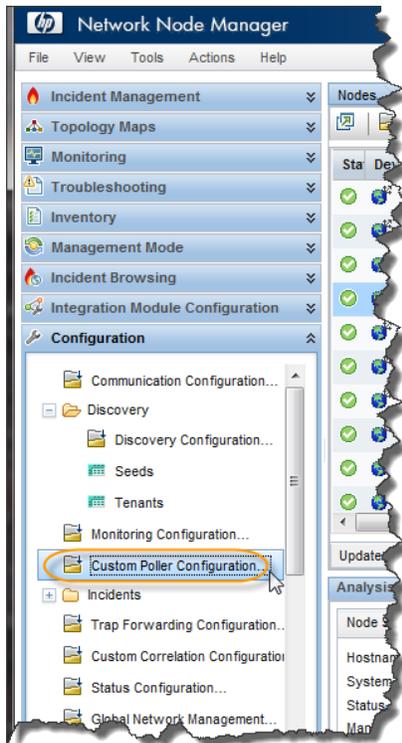
Step 1: Enable Custom Poller

Custom Poller is not enabled by default.

To enable Custom Poller:

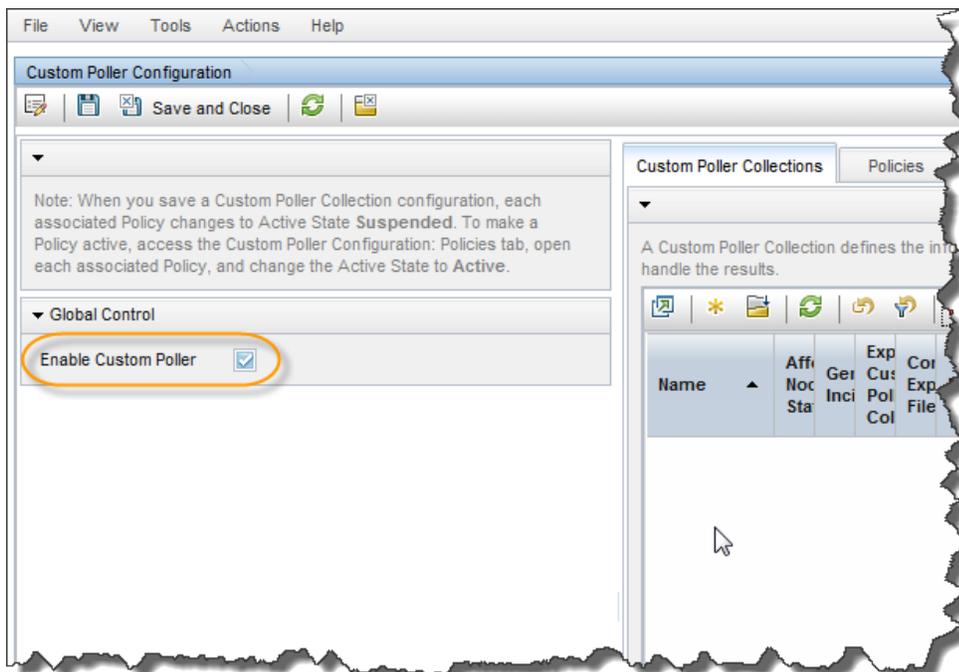
1. From the workspace navigation panel, select the **Configuration** workspace, expand the **Monitoring** folder, and then click **Custom Poller Configuration...**

Figure 4: Custom Poller Configuration



2. Select **Enable Custom Poller**.
3. Click  **Save and Close**.

Figure 5: Custom Poller Configuration: Enable Custom Poller

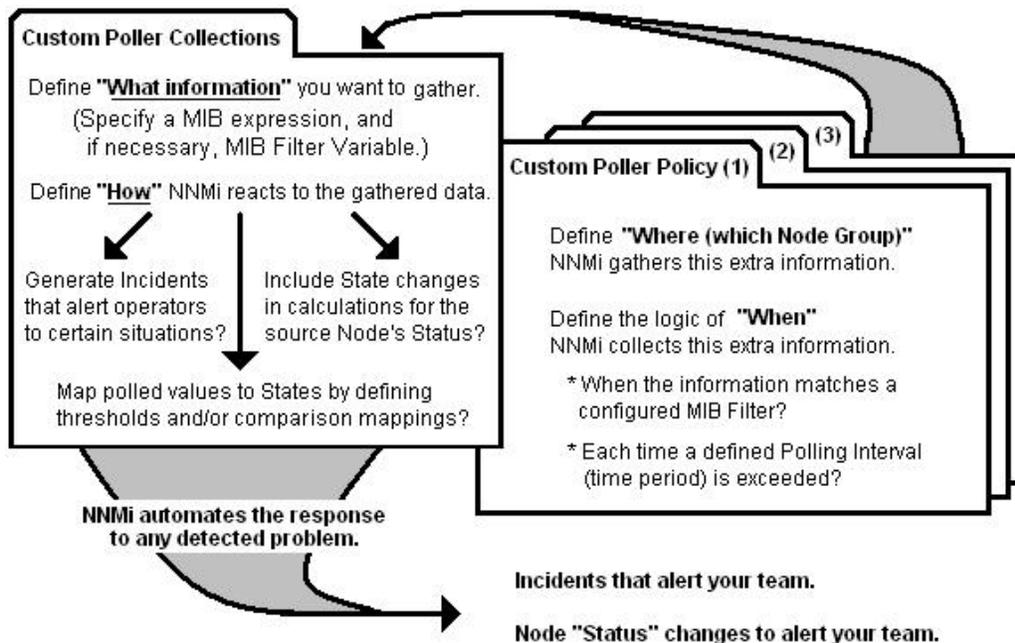


Step 2: Create a Custom Poller Collection

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. See the following diagram illustrating the hierarchy of Custom Poller Collections.

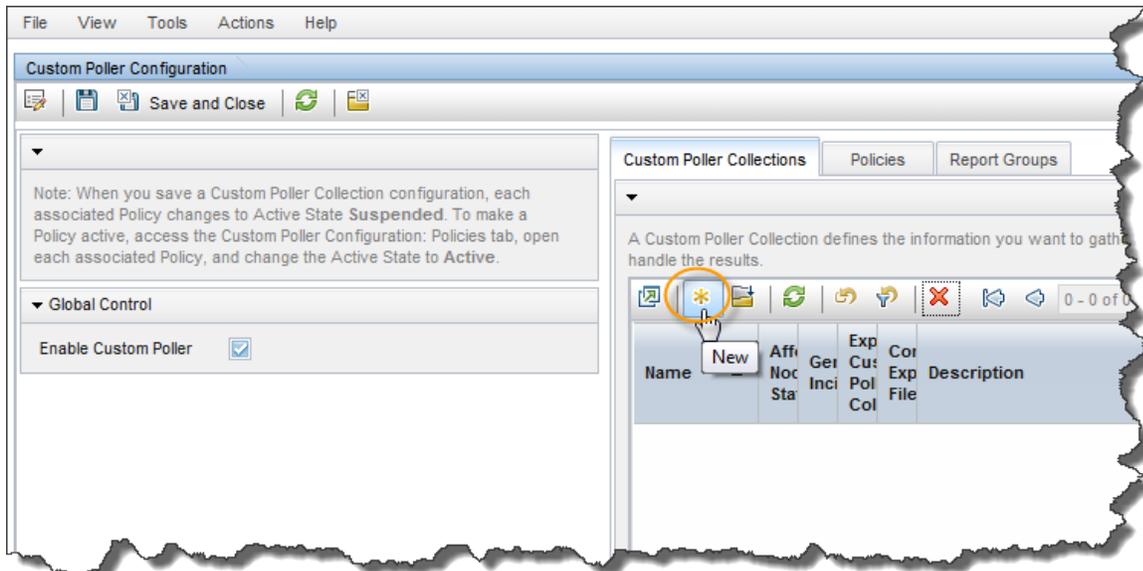
Figure 6: Custom Poller Collections and Policies



In this 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: Enable Custom Poller":
 - a. Navigate to the **Configuration** workspace.
 - b. Select **Custom Poller Configuration**.
2. Navigate to the **Custom Poller Collections** tab.
3. Click the * icon.

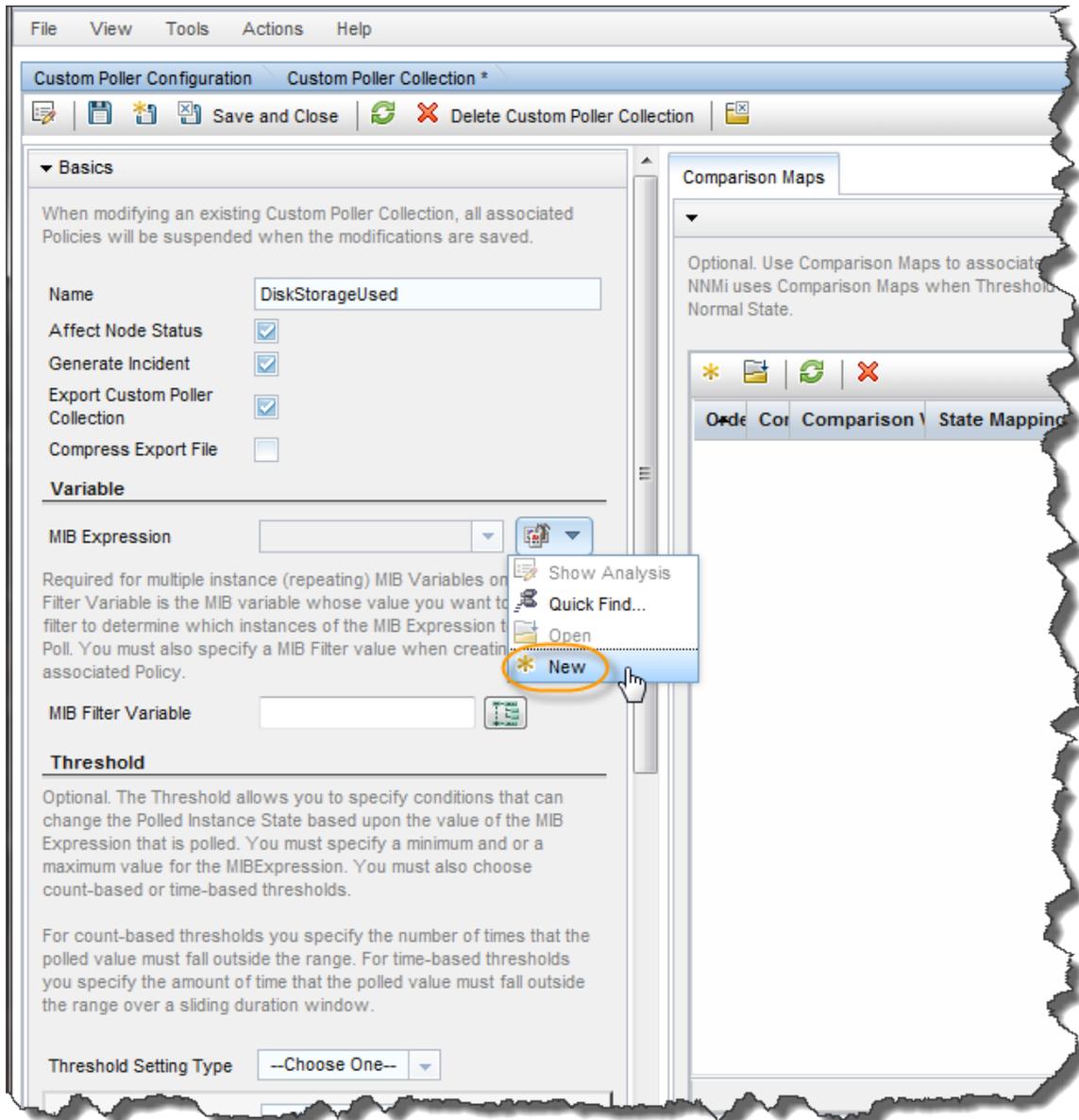
Figure 7: Custom Poller Configuration: New Custom Poller Collections

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

Next, we need to specify a MIB expression.

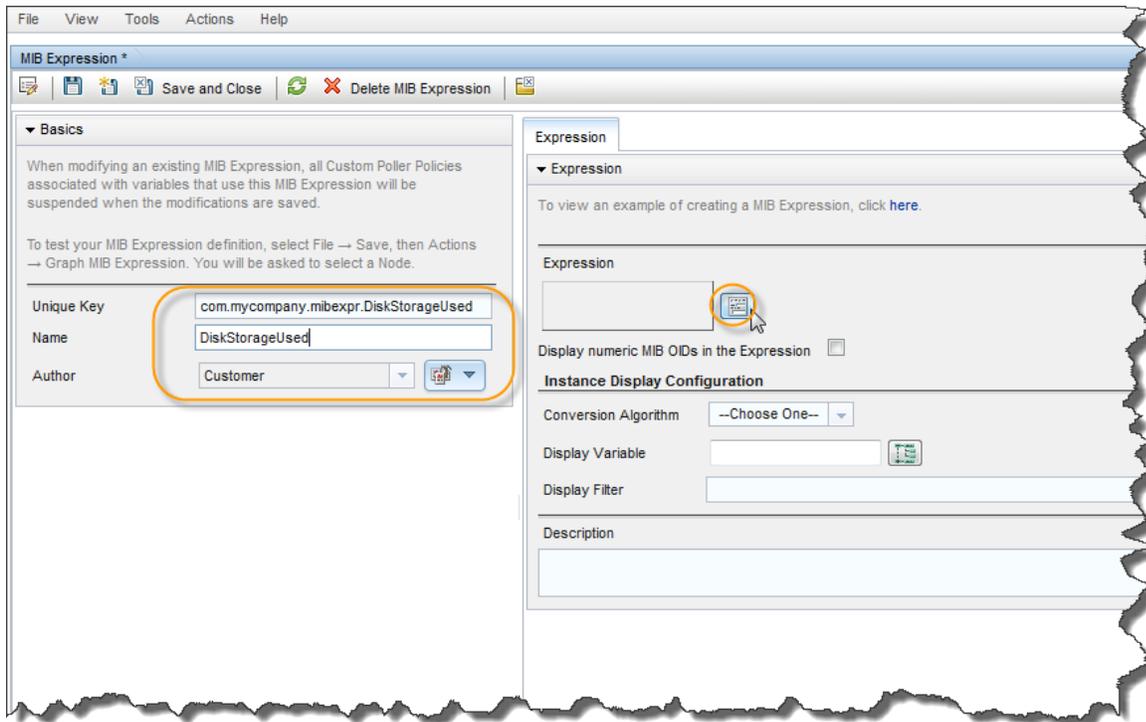
6. In the **MIB Expression** pull-down menu, click the * icon to open a New MIB Expression form.

September, 2011

Figure 8: Custom Poller Collection: New MIB Expression

7. Enter in a **Unique Key, Name** and **Author**. This example uses the author Customer, which is the default.

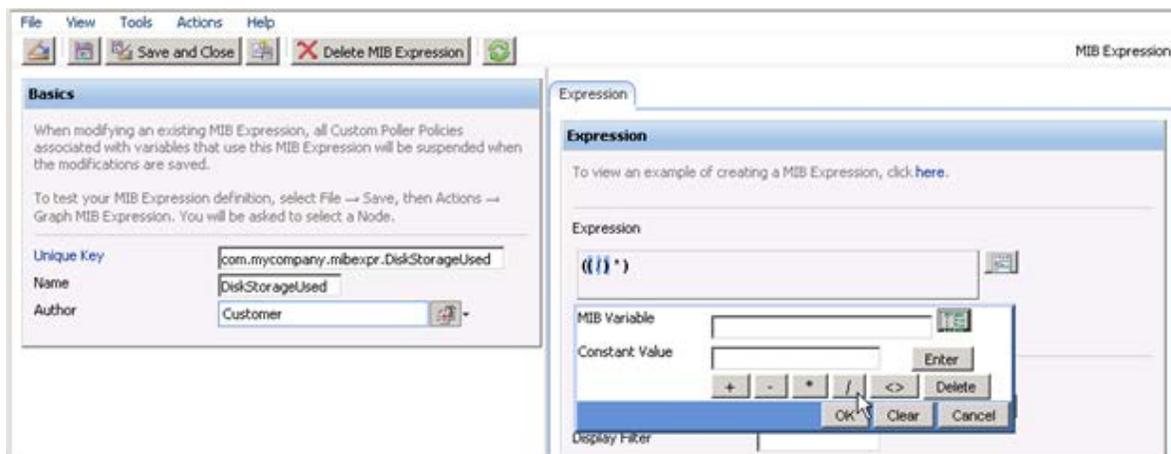
Next, create the MIB expression.

Figure 9: MIB Expression: Enter Basic Information

- Write out the expression before working with the user interface. In this example, the MIB expression is:

$$((hrStorageSize / 1000) * hrStorageAllocationUnits)$$

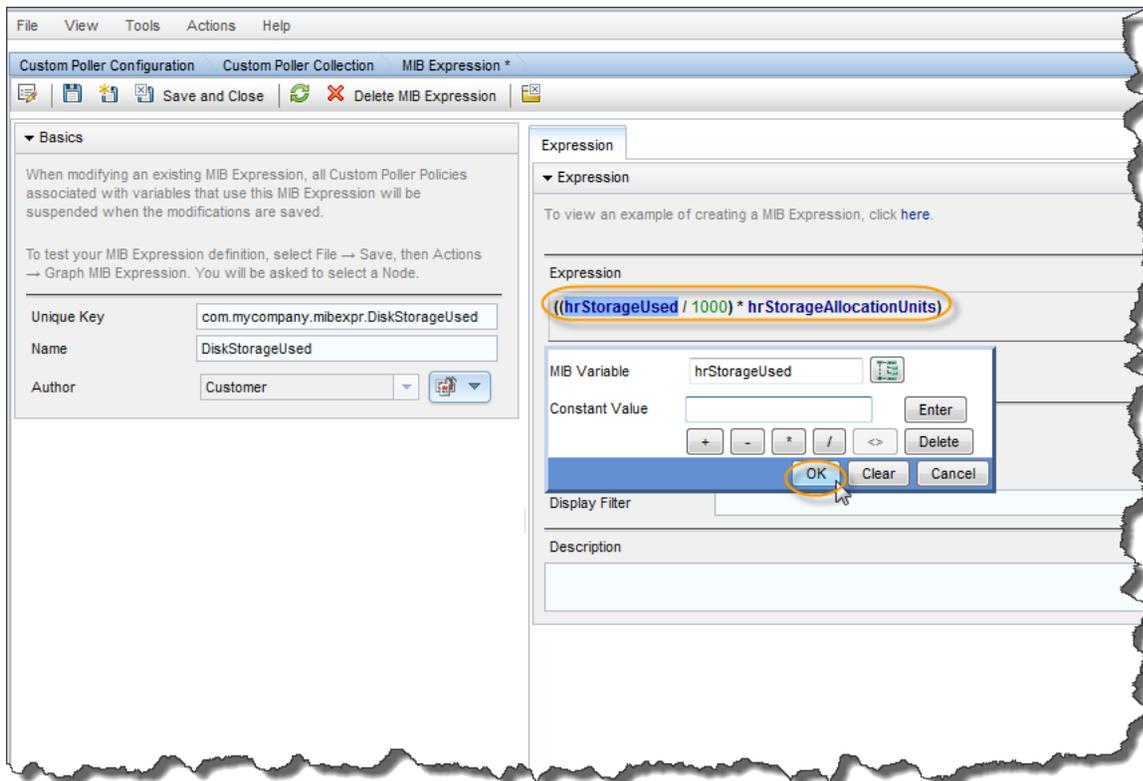
- When specifying the expression, you want to begin by inserting all of the operands. The operands in this expression include / and *.

Figure 10: MIB Expression: Define Expression

- After you select each operand, place the cursor in the location where you want to insert a MIB variable.

September, 2011

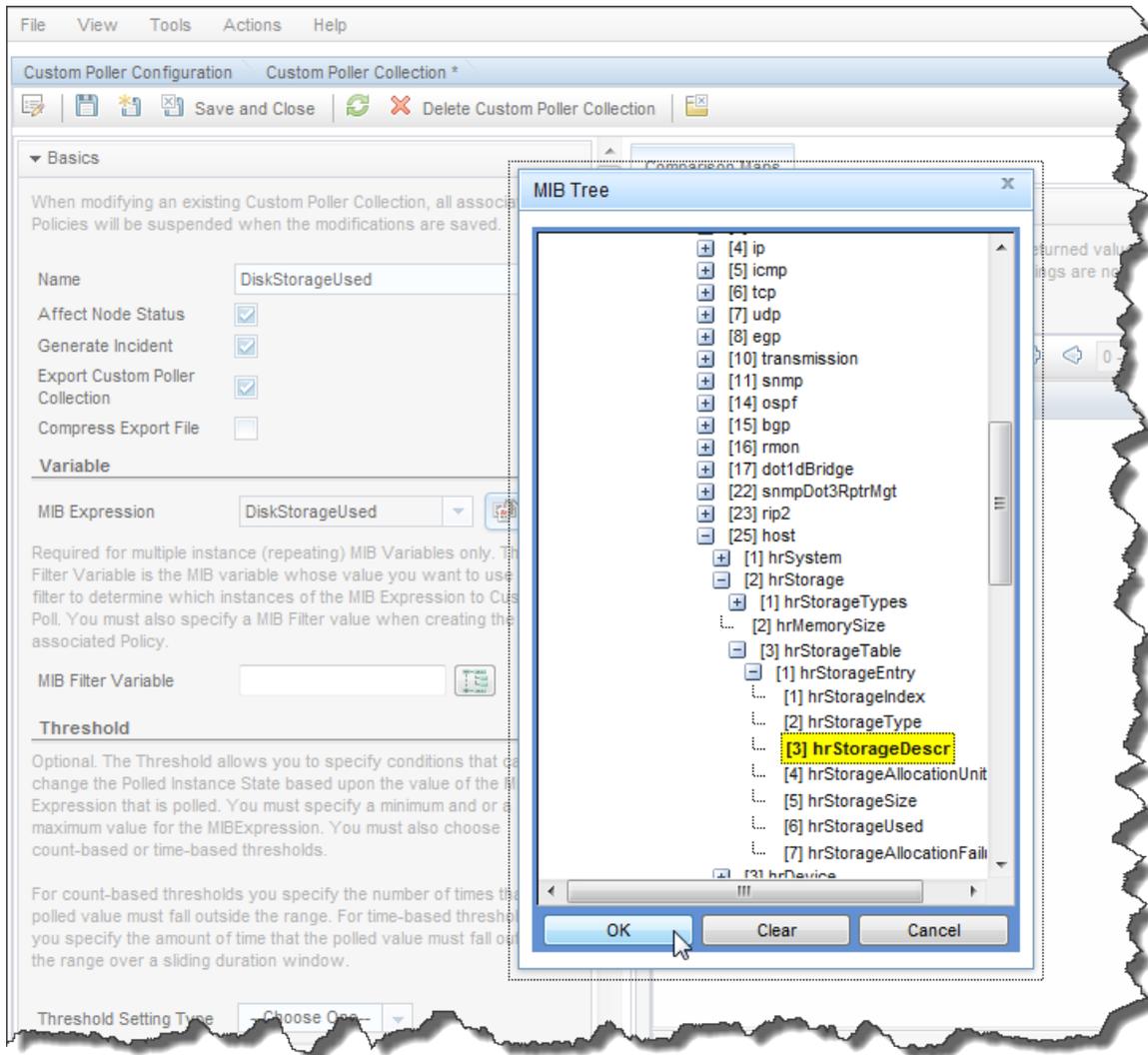
11. Navigate the MIB Variable tree to select the `hrStorageUsed` and `hrStorageAllocationUnits` variables.
12. Add the constant 1000.

Figure 11: MIB Expression: Expression Defined

13. Click **Save and Close**.

Next you must specify a MIB Filter Variable. This example uses `hrStorageDescr`.

14. From the MIB Filter Variable entry, navigate to the `hrStorageDescr` value.

Figure 12: Custom Poller Collection: Specify MIB Variable

Finally, set a threshold to trigger an alarm. NNMi 9.10 supports two types of threshold settings, one is Count based and the other is Time based. The Count based threshold means that NNMi must have a certain number of consecutive samples above the threshold to trigger it. The Time based threshold means that NNMi must have a certain number of samples over a sliding time window but they do not need to be consecutive. This example uses a Count based threshold with 36827000 as the threshold value with a rearm value of 36826000. The alarm triggers with just one sample above the threshold.

Figure 13: Custom Poller Collection: Specify Threshold

File View Tools Actions Help

Custom Poller Configuration Custom Poller Collection *

Save and Close Delete Custom Poller Collection

MIB Expression

Required for multiple instance (repeating) MIB Variables only. The MIB Filter Variable is the MIB variable whose value you want to use as a filter to determine which instances of the MIB Expression to Custom Poll. You must also specify a MIB Filter value when creating the associated Policy.

MIB Filter Variable

Threshold

Optional. The Threshold allows you to specify conditions that can change the Polled Instance State based upon the value of the MIB Expression that is polled. You must specify a minimum and or a maximum value for the MIBExpression. You must also choose count-based or time-based thresholds.

For count-based thresholds you specify the number of times that the polled value must fall outside the range. For time-based thresholds you specify the amount of time that the polled value must fall outside the range over a sliding duration window.

Threshold Setting Type

High State

High Value

High Value Rearm

High Trigger Count

High Duration

High Window Duration

Low State

Low Value

Comparison Maps

Optional. Use Comparison Maps to define the Normal State.

Code Color Compare

8. In the **High Value** attribute, enter 36827000.
9. In the **High Value Rearm** attribute, enter 36826000.
10. In the **High Trigger Count** attribute, enter 1.
11. Click  **Save and Close**.

You now have a Custom Poller Collection.

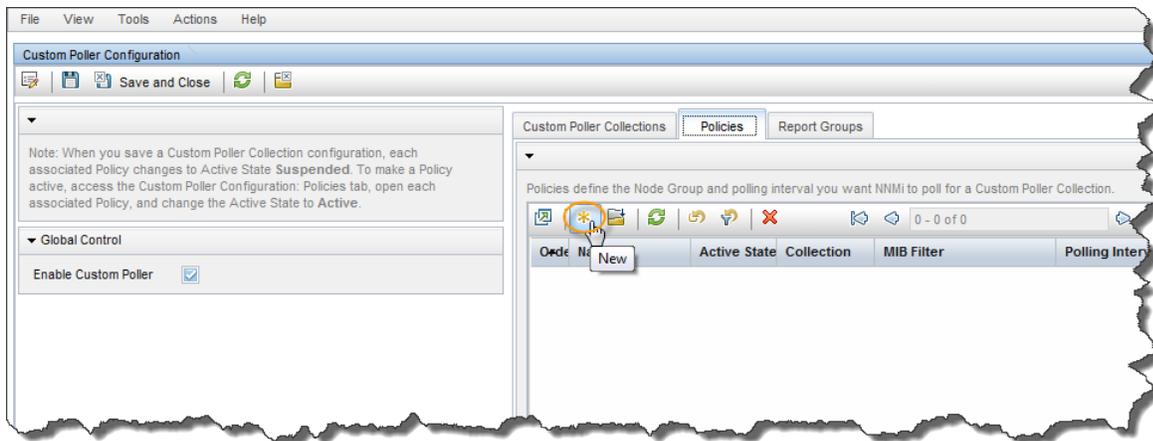
Next, you create a policy for the Custom Poller Collection that looks for C: at the start of the value.

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 used to define the MIB filter to select specific instances. You can create more than one Policy associated with a Custom Poller Collection.

1. To begin, navigate to the **Policies** tab and select the * icon.

Figure 14: Custom Poller Configuration: Policies Tab



2. 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 ordering.

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.

3. In the **Ordering** attribute, accept the default value of 1.
4. Use the **Quick Find** pull-down menu to select the Collection that we previously defined.

5. Change the **Active State** to *Active*.
6. Use the **Quick Find** pull-down menu to select the `Microsoft Windows Systems Node Group`.

Next, specify the MIB Filter.

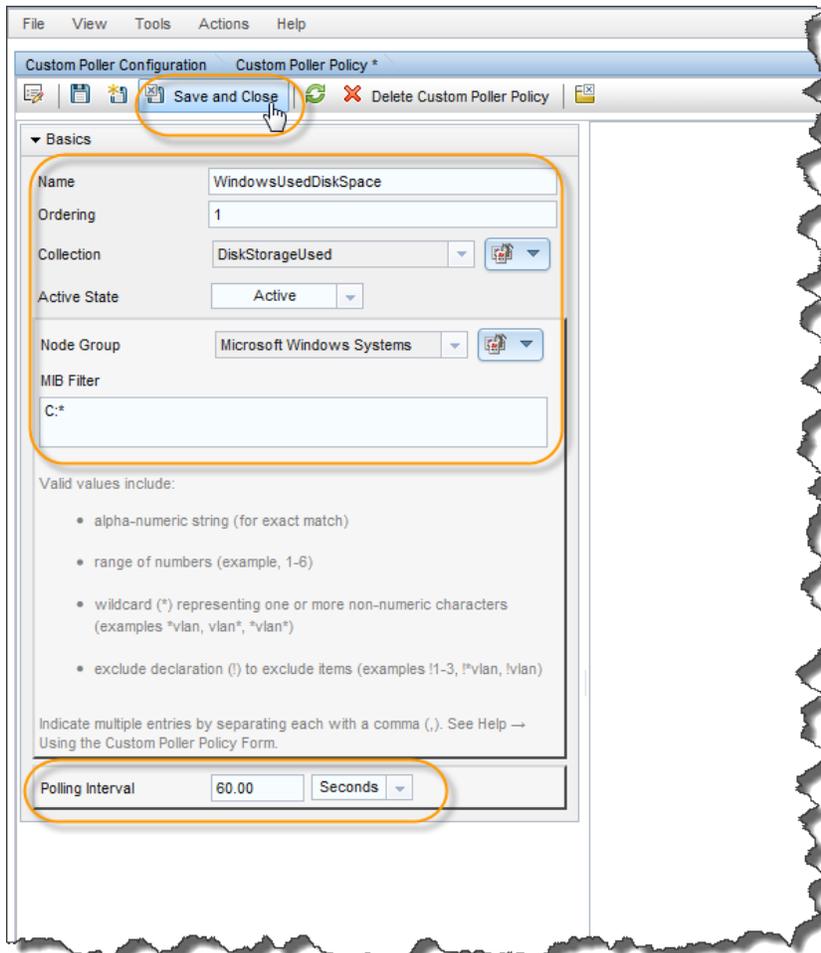
7. Recall that the values from our SNMP walk, the C drive always started with `C:`. Therefore, in the MIB Filter attribute, enter `C:*` as the filter. Avoid matching all instances by entering the asterisk (*) as the filter. This could lead to a large number of matches and adversely affect NNMi Custom Poller performance.

Finally, 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.

8. For the purposes of this example, set the Polling Interval attribute to 1 minute. 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.
9. Click  **Save and Close** to save your Policy.

Tip: 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*.

Figure 15: Custom Poller Policy

Viewing 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.

To view Custom Node Collections:

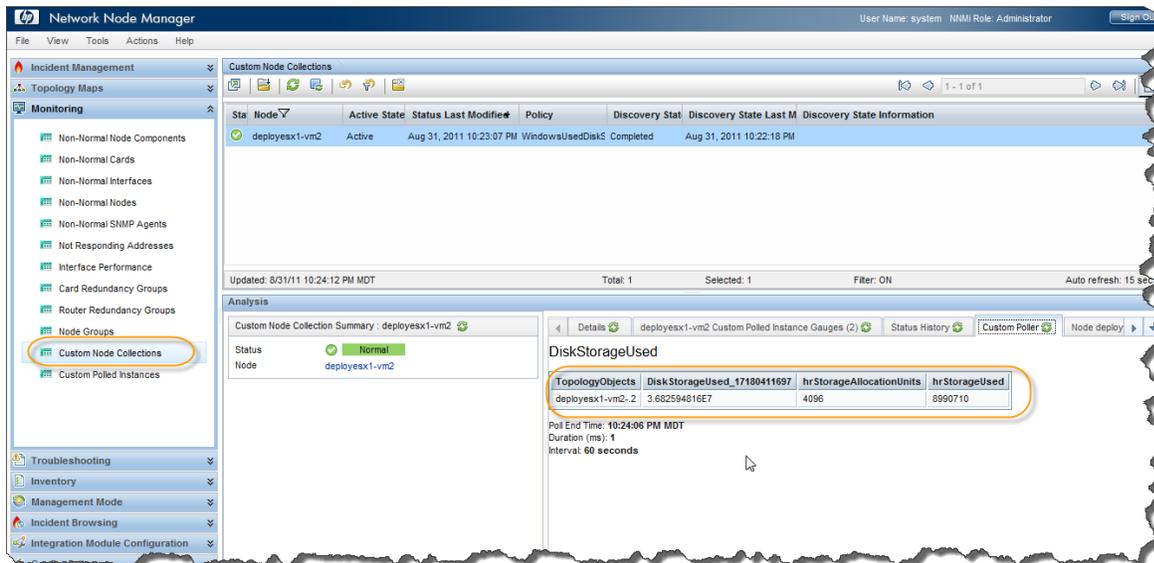
1. Navigate to the **Monitoring** workspace.
2. Click **Custom Node Collections**.

NNMi displays a table view of all Custom Node Collections, which includes the following:

- The Custom Node Collection status.
- The topology node associated with the Custom Node Collection.
- The Active State for the associated policy.
- The date and time the Status was last modified.

September, 2011

- 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 time stamp when the Discovery State was last modified, and Discovery State Information.
- You can also see the details of the collection by clicking the **Custom Poller** tab in the Analysis Pane.

Figure 16: Custom Node Collections: Custom Poller Tab

Tip: 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 Polled Instance

Another useful view is the Custom Polled Instances view. 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.

Note: The Custom Polled Instance value does not necessarily reflect the most recent polled value. It is the value that caused the State change, which might not be the subsequent value.

To view Custom Polled Instances:

1. Navigate to the **Monitoring** workspace.
2. Click **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, the table does list 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.

Figure 17: Custom Polled Instances

Sta	Last State Chan	MIB Expression	MIB Filtr	Disj	Node	Collection	Active State	State Last Modified
✓	36,825,948.16	DiskStorageUsed	.2	C:\L	deploysx1-vm2	DiskStorageUsed	Active	Aug 31, 2011 10:23:06 PM

TopologyObjects	DiskStorageUsed_17180411697	hr StorageAllocationUnits	hr StorageUsed
deploysx1-vm2-.2	3.6825899008E7	4096	8990698

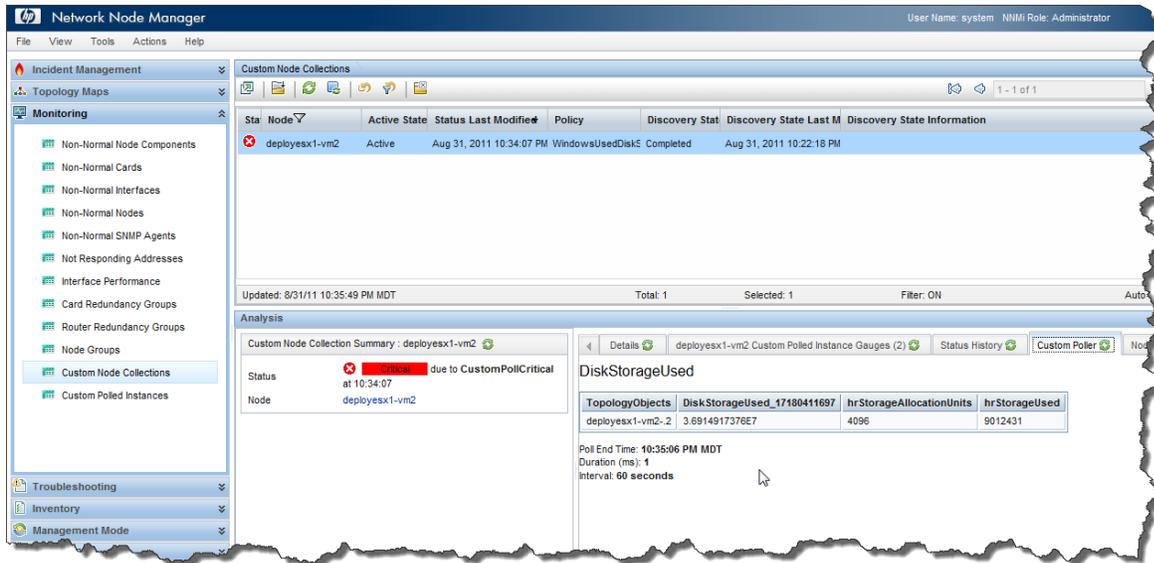
Step 3: Evaluate the Results of the Custom Poll

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

1. Copy a few large files onto the disk of the PC to increase the disk usage.

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

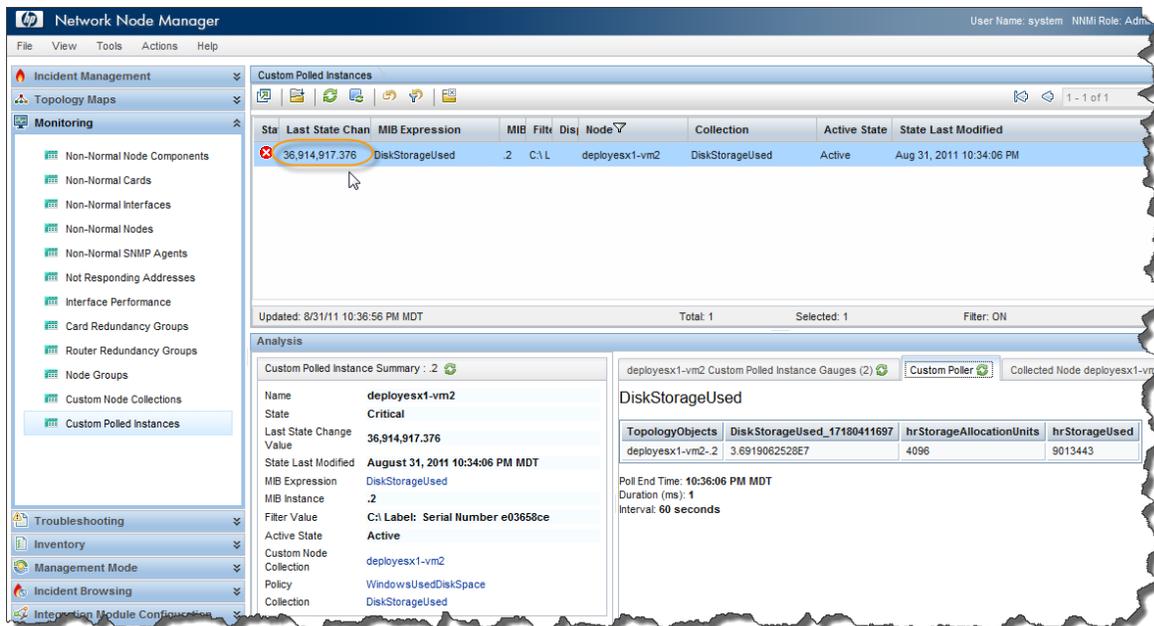
Figure 18: Custom Node Collections: Critical Status



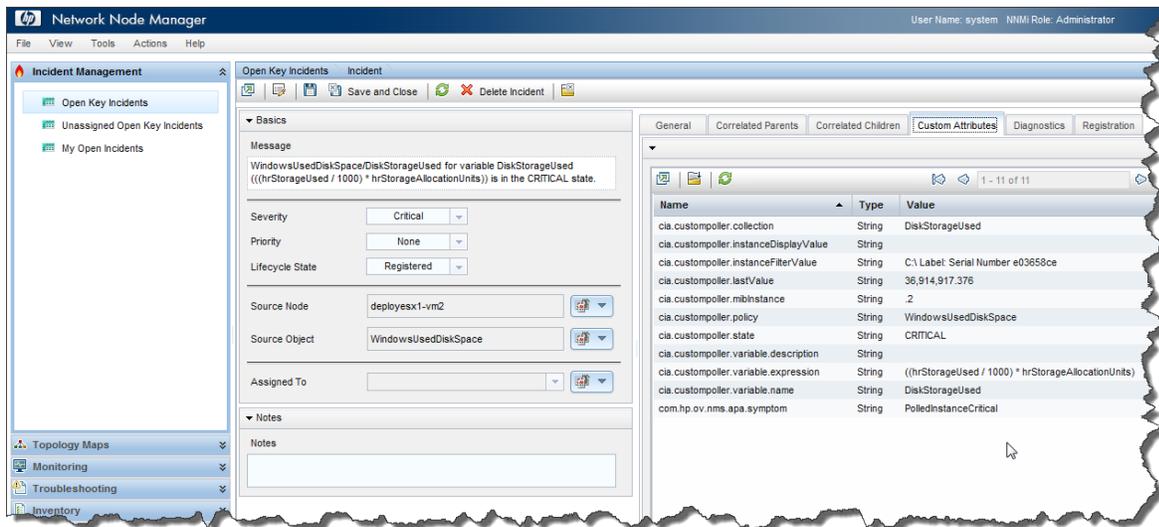
Note: The Custom Node Collection Status is not necessarily equivalent to the Status of the Source Node.

2. Open the Custom Poller Collection to see that the value of `hrStorageUsed` is above the **High Value** threshold. (This is the value that triggered the State change.)

Figure 19: Custom Polled Instances



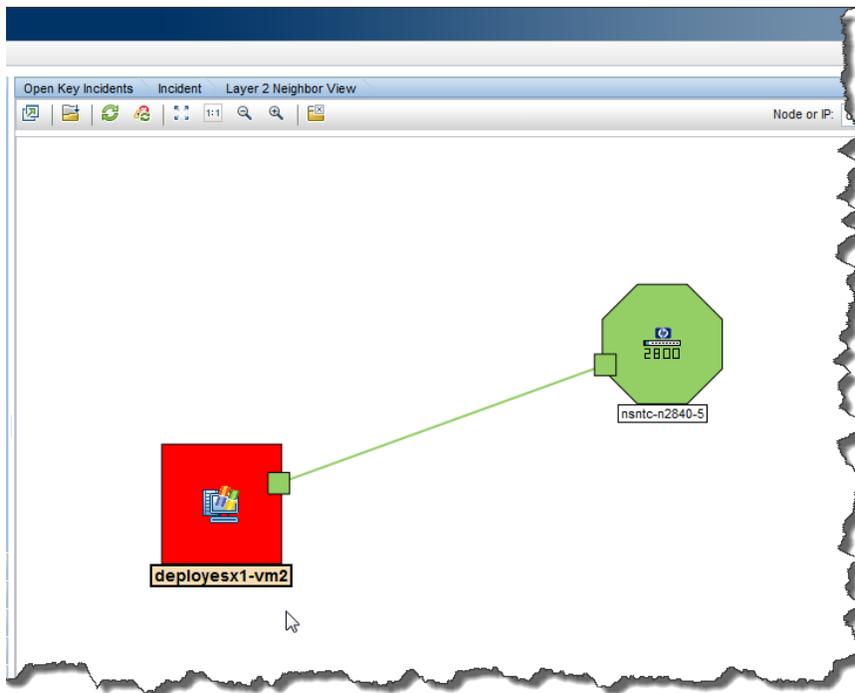
3. Navigate to the Incident form to see that an incident was generated.

Figure 20: Open Key Incidents

Note the possible Custom Attributes associated with this incident on the **Custom Attributes** tab.

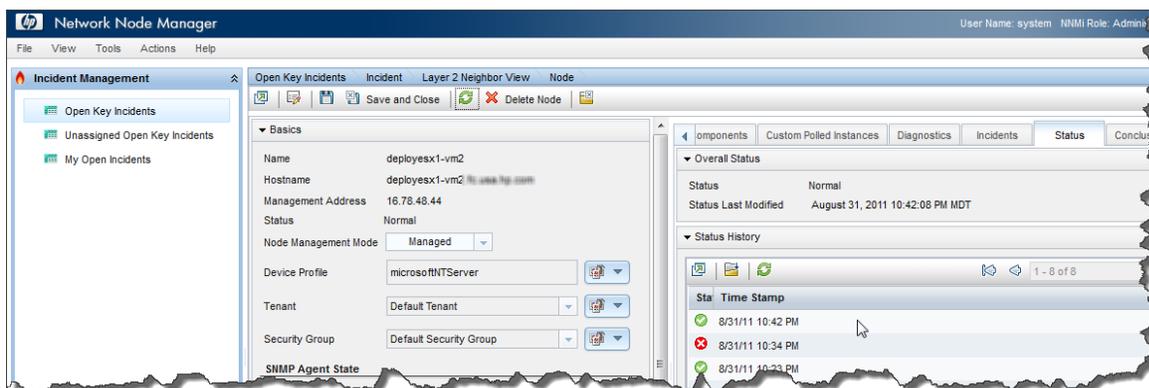
The Incident message presents the names used in creating the Custom Poller Collection. You can change this message by editing the Incident Configuration.

- From the workspace navigation panel, select the **Configuration** workspace, expand the **Incidents** folder, and then click **Management Event Configurations**.
- Double-click the CustomPollCritical incident configuration.
- For example, to display the most recent value that caused the Custom Node Collection Status to change, you might want to include the Custom Attribute `cia.LastValue` in your message.
- To verify that the Status of the Source Node has changed to `Critical`, open the Source Node or select a Node View or Map.

Figure 21: Layer 2 Neighbor View

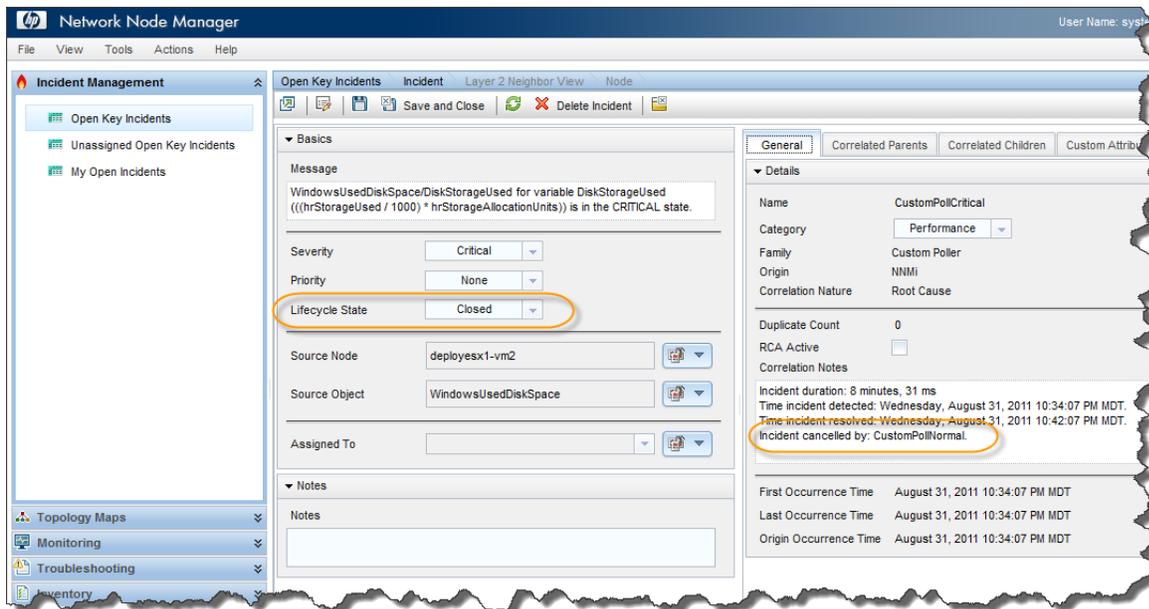
After you verify that the Custom Poll is successful and NNMi properly indicates that the disk space is Critical, return the disk to its previous State.

8. Delete the large file from the PC.
9. Verify the Custom Node Collection's Status has returned to Normal, by opening the Custom Node Collection form and clicking the Status tab.

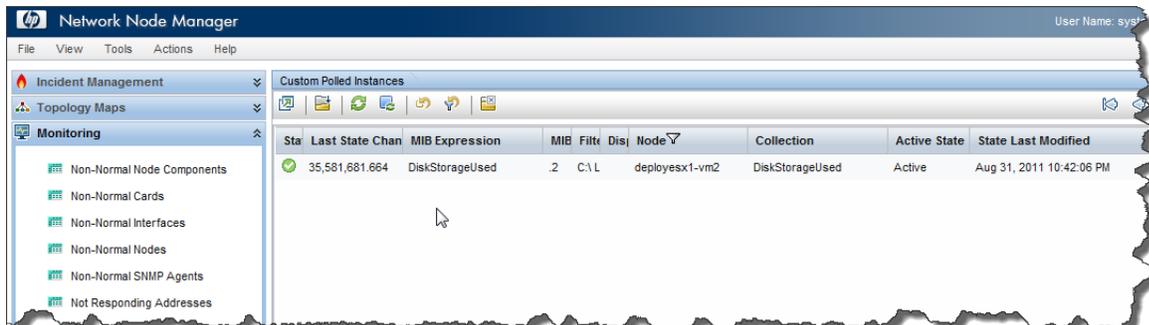
Figure 22: Layer 2 Neighbor View: Node Status

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

September, 2011

Figure 23: Incident Form: State: Closed

- Return to the Custom Polled Instances view to verify that the value of the MIB Poll Variable is below the **High State** threshold.

Figure 24: Custom Polled Instances

- After completing your initial testing, set the poll rate back to the desired value; for example, 5 minutes.

Exporting the Custom Poller Collection

NNMi enables you to store all the Custom Poller samples to a Comma-Separated Values (CSV) file.

To enable the export feature, on the Custom Poller Collection form, select the Export Custom Poller Collection check box. This selection causes NNMi to generate CSV files for each collection.

Note: The exported CSV file contains all samples, not just the samples that trigger a state change.

Detecting Reboots Example

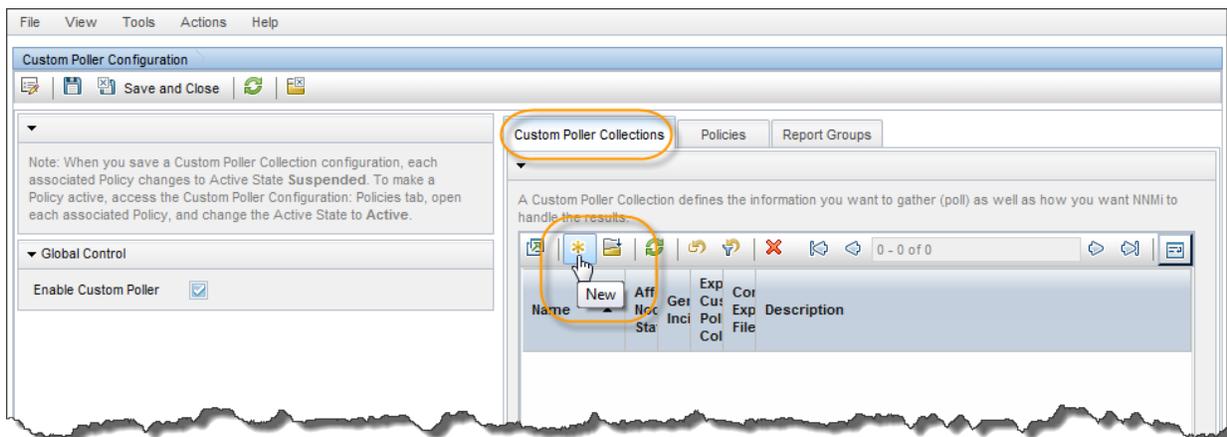
In this example, we are detecting reboots of devices by polling the sysUpTime value and looking for a low value indicating that the node has been rebooted. When the sampled value is smaller than a given threshold, the node changes status and an alarm is generated. Then, after a period of time, the sysUpTime increases above the threshold and the alarm and status should clear. We are setting a threshold on time ticks lower than 10 minutes. Since sysUpTime is defined as the time (in hundredths of a second) from when the network management portion of the system was last re-initialized, we are using the value $10 \times 60 \times 100$ (ten minutes times 60 seconds times 100 milliseconds) to arrive at the value of 60000. If a sample is lower than this, that means the device rebooted within the last 10 minutes. We are defining the alarm to clear after the device has been up for 2 hours. To define such an alarm, we use $120 \times 60 \times 100 = 720000$. This is our rearm value. When a sample is above this value, the state goes back to normal and the alarm is cleared.

Step 1: Create a Custom Poller Collection

To create the Custom Poller Collection:

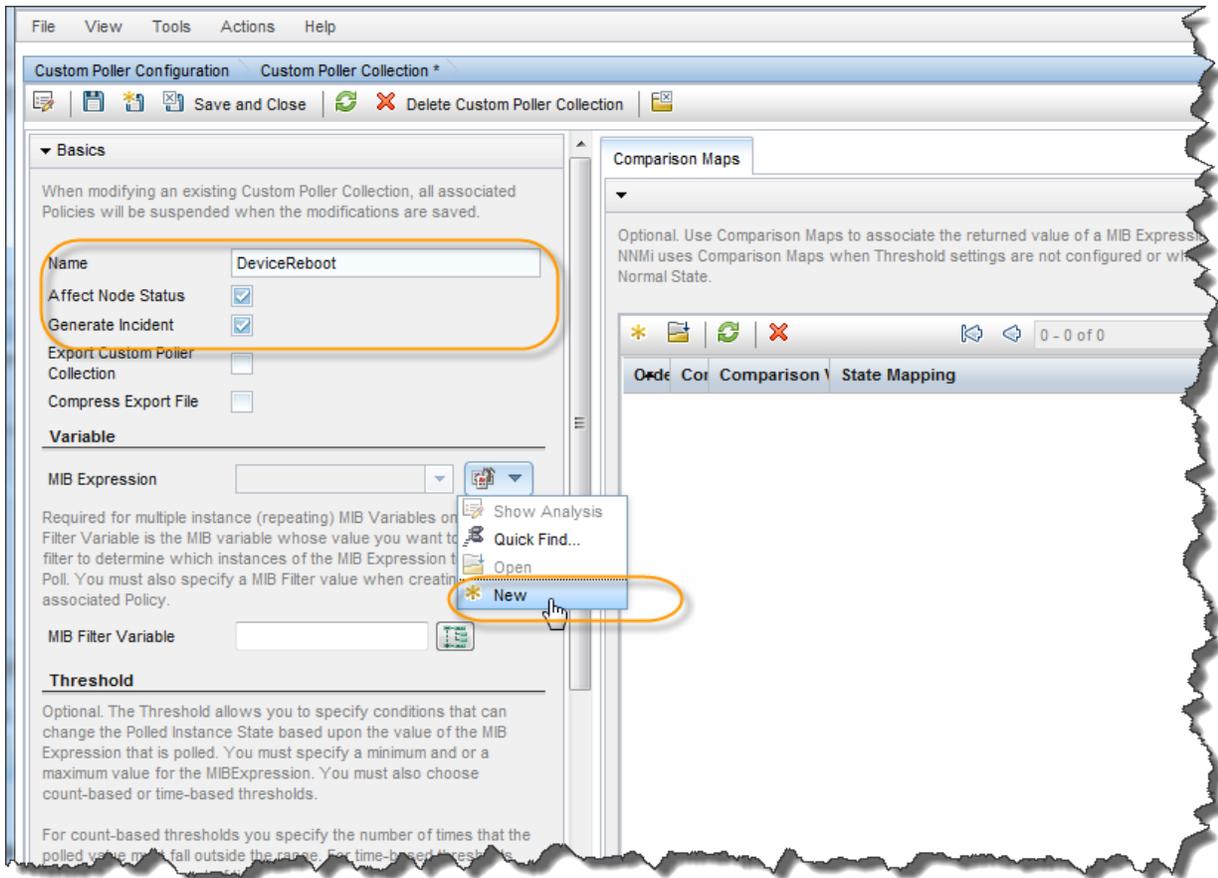
1. From the workspace navigation panel, select the **Configuration** workspace, expand the **Monitoring** folder, and then click **Custom Poller Configuration...**
2. Click the **Custom Poller Collections** tab and then click the * icon.

Figure 26: Custom Poller Collections: New



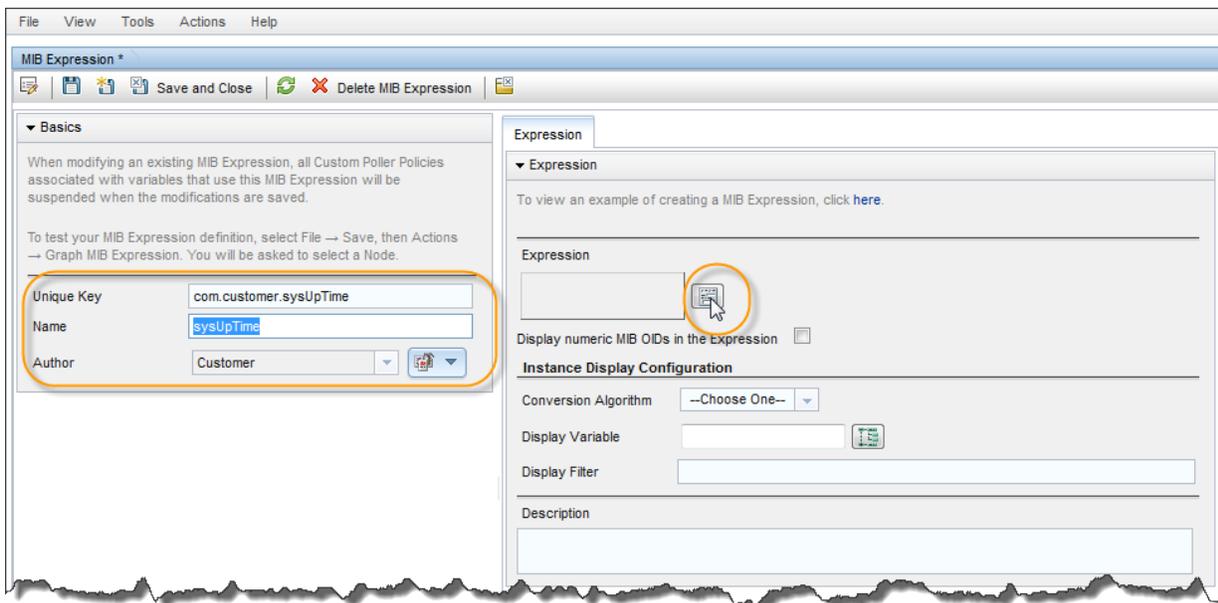
3. Fill in **Name** (DeviceReboot) and select the **Affect Node Status** and **Generate Incident** check boxes.
4. Then, click the drop-down menu to create a new **MIB Expression**.

Figure 27: Custom Poller Collection: New MIB Expression



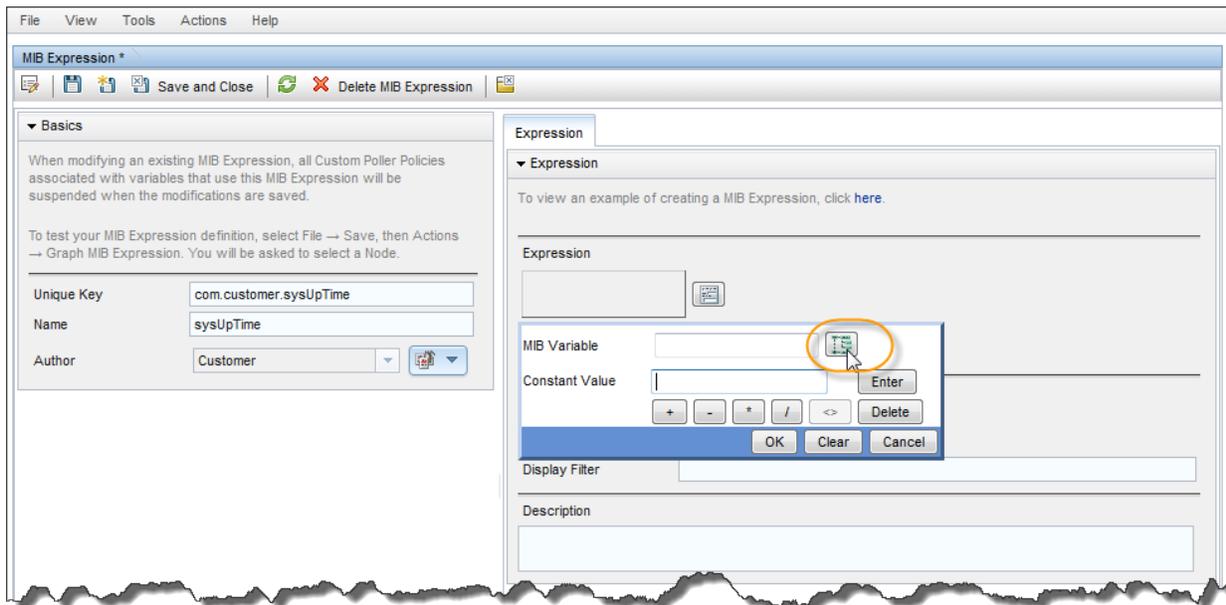
5. Enter a **Unique Key** (this can be anything, however, the general convention is shown in the following figure), **Name** and **Author**. Then click the **Expression** button.

Figure 28: MIB Expression



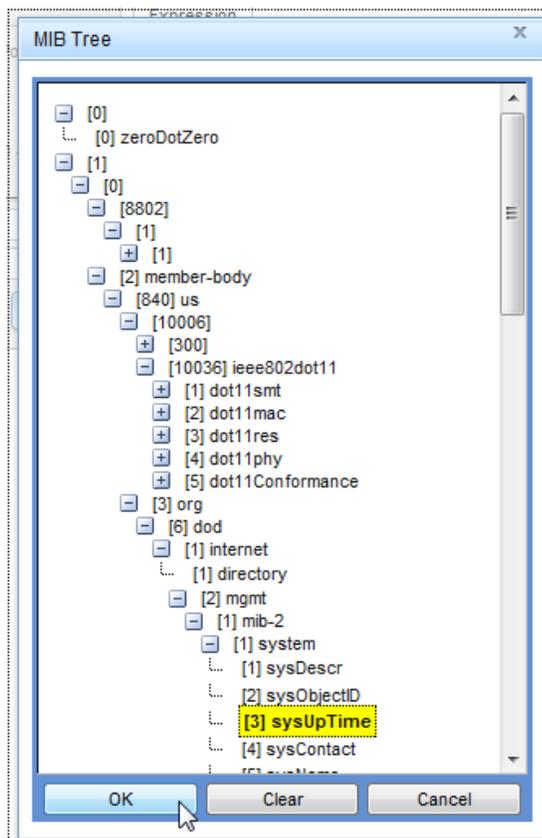
6. Click the **MIB Variable** tree button.

Figure 29: MIB Variable Tree



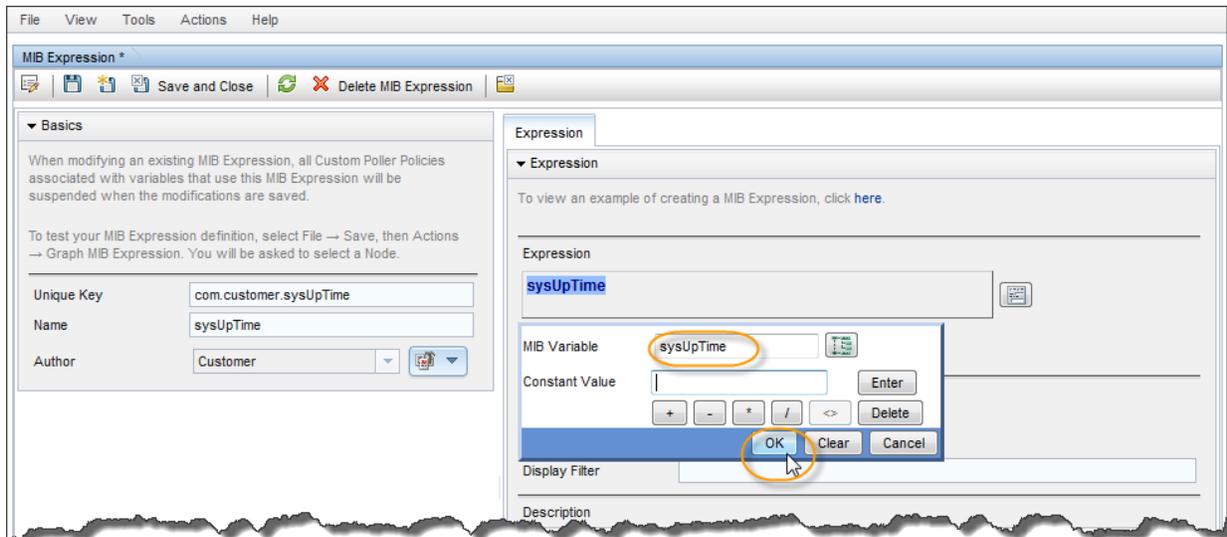
7. Navigate down to the `sysUpTime` variable and click **OK**.

Figure 30: MIB Tree: sysUpTime Variable

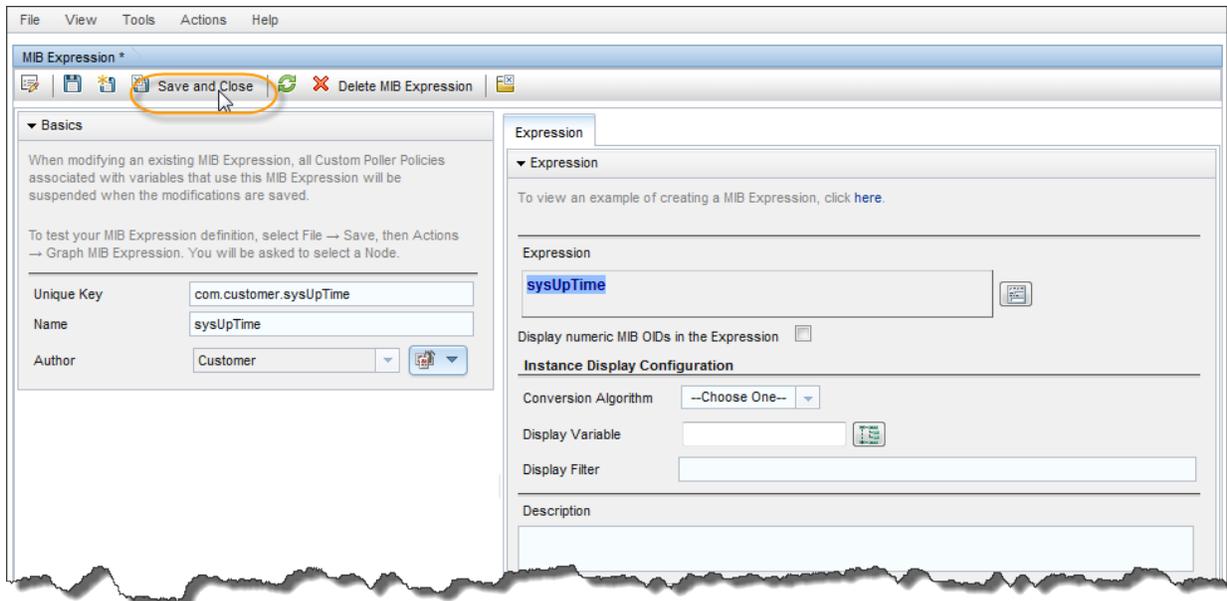


8. Click **OK**.

September, 2011

Figure 31: MIB Expression: OK

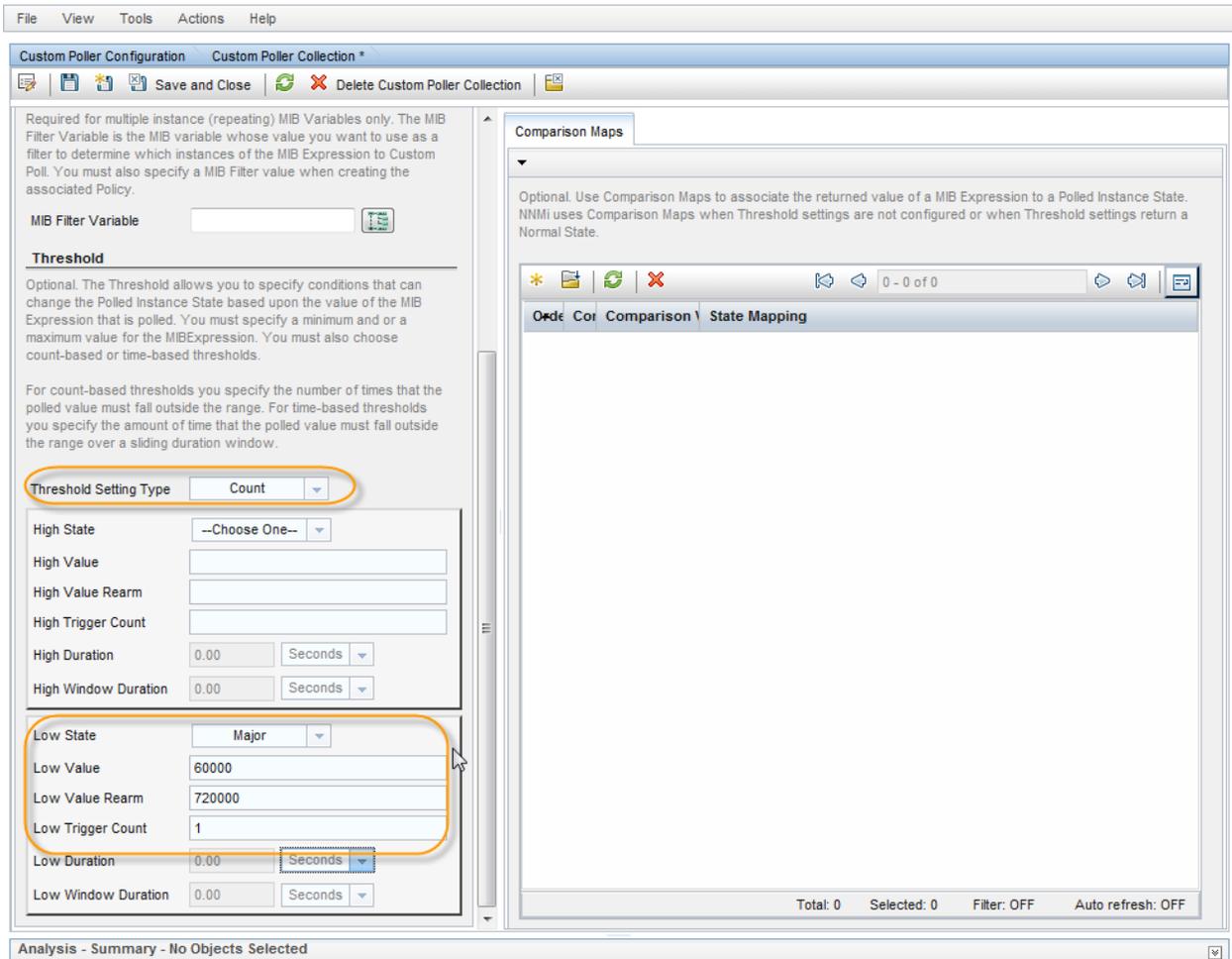
9. Click  **Save and Close** to complete the MIB Expression definition.

Figure 32: MIB Expression: Save and Close

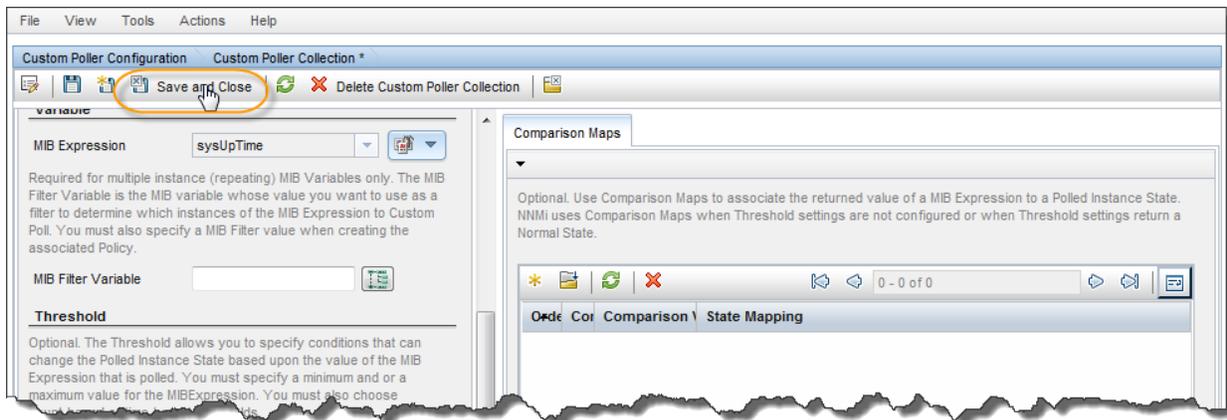
Note: In NNMi 9.00, if a MIB expression only contains an SNMP OID that is of type TimeTicks, then it is treated as a LONG and will report the raw TimeTicks value. NNMi 9.10 enhanced that logic a bit to include the case where if the MIB expression only had one OID in a mathematical expression, and that single OID was of type TimeTicks, then NNMi uses the raw TimeTicks value. In all other cases, NNMi treats TimeTicks as a counter and uses the delta value.

10. Now we need to set the threshold values. We use a `Count` threshold type and set the values as discussed previously. We also set the **Low Trigger Count** of 1 because we only need one sample below the threshold to trigger the state change and the alarm.

Figure 33: Custom Poller Collection: Threshold Settings

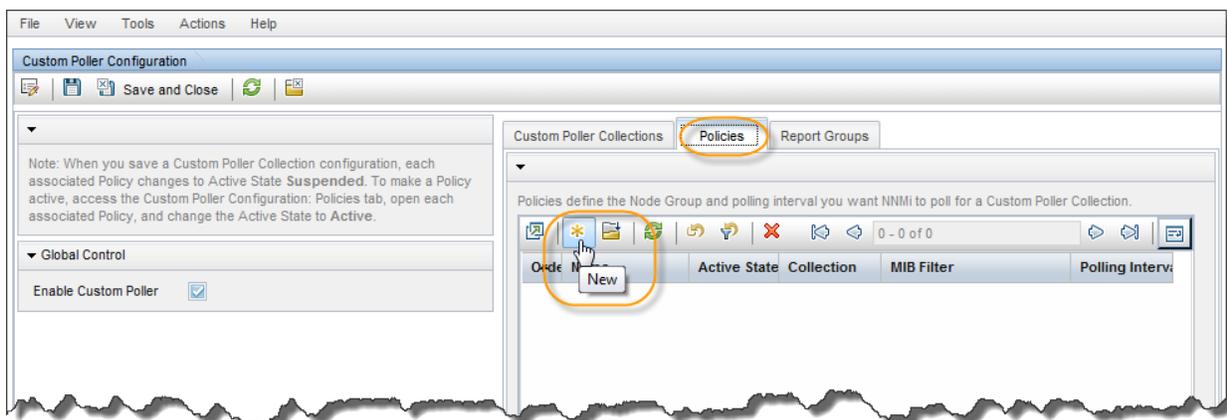


11. Click  **Save and Close** to complete the Custom Poller Collection.

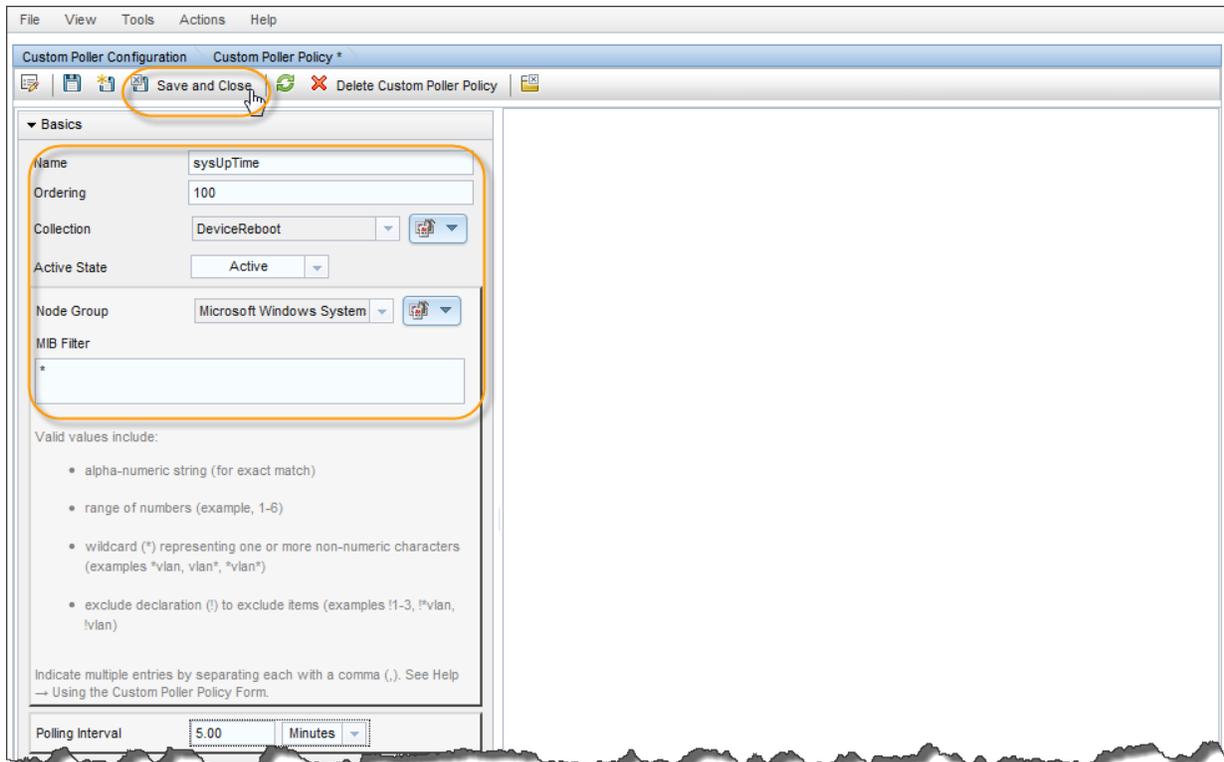
Figure 34: Custom Poller Collection: Save and Close**Step 2: Create a Policy for a Custom Poller Collection**

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

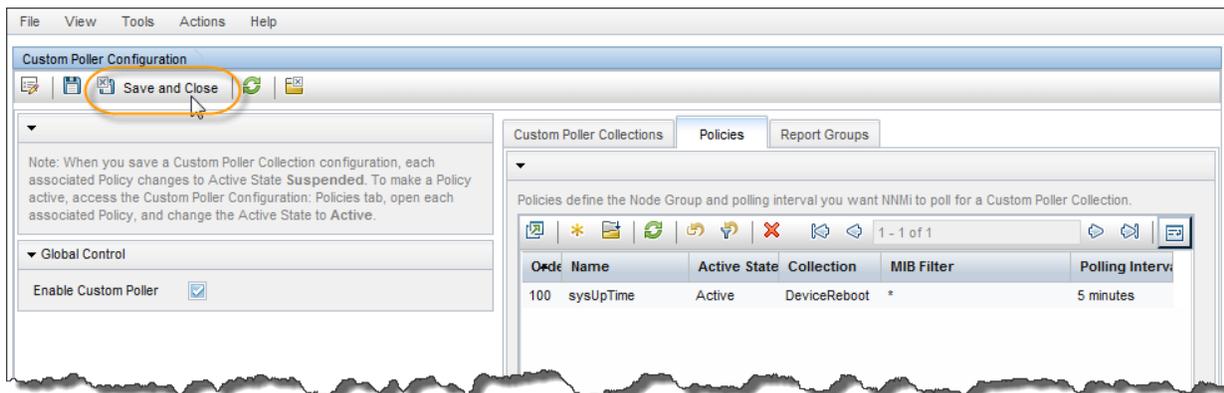
1. From the workspace navigation panel, select the **Configuration** workspace, expand the **Monitoring** folder, and then click **Custom Poller Configuration**.
2. Click the **Policies** tab and then click the * icon.

Figure 35: Custom Poller Configuration: New Policy

3. Fill in the values. Select the appropriate Node Group and MIB filter. In this example, we are rebooting a server so we only apply the collection to Microsoft Windows Systems. Use an asterisk (*) for the MIB filter since sysUpTime is a scalar value. Set the Active State to **Active**. Then click **Save and Close**.

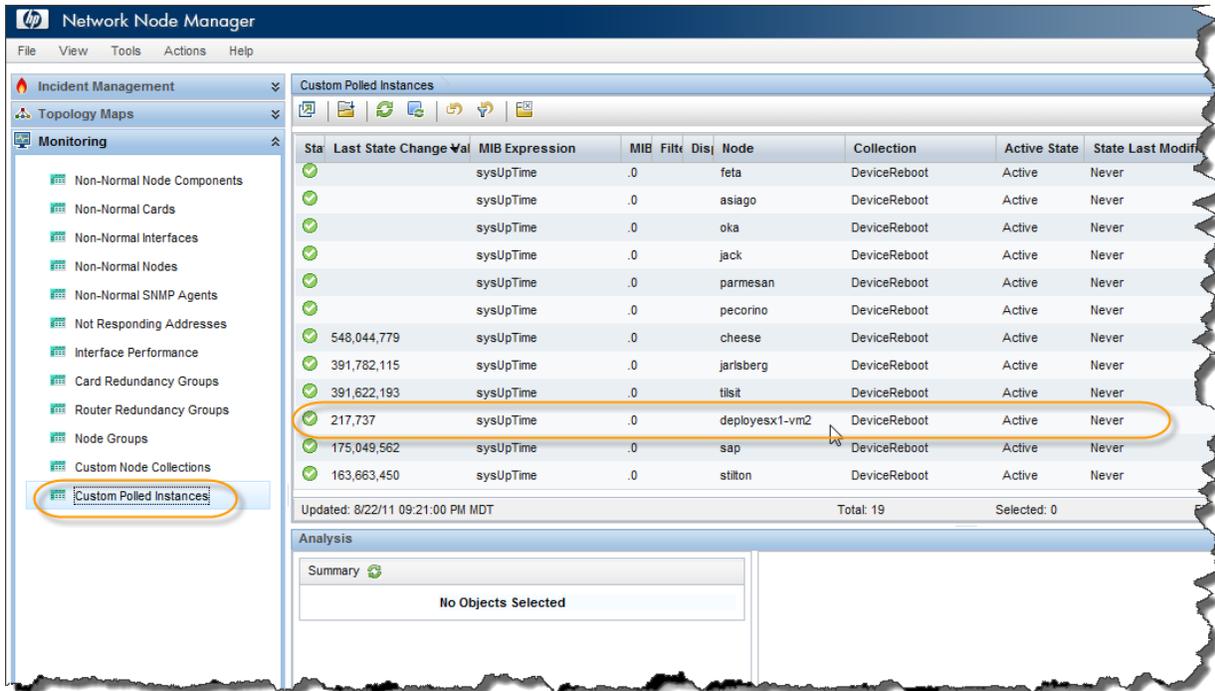
Figure 36: Custom Poller Policy: Settings, Save and Close

4. Finally, click  **Save and Close** to complete the Custom Poller Configuration.

Figure 37: Custom Poller Configuration: Save and Close

If you select the **Monitoring** workspace and click **Custom Polled Instances**, you can see the instances and their Last State Change Value. Since these instances went from Not Polled to Normal, you see the initial sample. Note that the Last State Change Value column is not updated until a state change occurs. The server we are interested in, `deploysx1-vm2`, is highlighted in the following figure.

Figure 38: Custom Polled Instances: Before Reboot



Step 3: Test the Policy

After rebooting the server, you can see that the `sysUpTime` value falls below the threshold. The node changes status and an incident is generated. This incident clears after the node has been up for two hours.

Figure 39: Custom Polled Instances: After Reboot

Network Node Manager

Custom Polled Instances

Id	Last State Change Value	MIB Expression	MIB Instance	Node	Collection	Active State	State Last Modified
548,044,779	sysUpTime	sysUpTime	.0	cheese	DeviceReboot	Active	Never
547,813,371	sysUpTime	sysUpTime	.0	feta	DeviceReboot	Active	Never
527,362,411	sysUpTime	sysUpTime	.0	brie	DeviceReboot	Active	Never
472,786,559	sysUpTime	sysUpTime	.0	taleggio	DeviceReboot	Active	Never
4,685	sysUpTime	sysUpTime	.0	deploysx1-vm2	DeviceReboot	Active	Aug 22, 2011 9:29:01 PM MDT
391,782,115	sysUpTime	sysUpTime	.0	jarlsberg	DeviceReboot	Active	Never
391,622,193	sysUpTime	sysUpTime	.0	tilsit	DeviceReboot	Active	Never
30,988,729	sysUpTime	sysUpTime	.0	limburger	DeviceReboot	Active	Never
175,049,562	sysUpTime	sysUpTime	.0	sap	DeviceReboot	Active	Never
163,663,450	sysUpTime	sysUpTime	.0	stilton	DeviceReboot	Active	Never
1,012,173,555	sysUpTime	sysUpTime	.0	mascarpone	DeviceReboot	Active	Never
1,012,114,009	sysUpTime	sysUpTime	.0	parmesan	DeviceReboot	Active	Never

Updated: 8/22/11 09:29:17 PM MDT Total: 19 Selected: 1 Filter: OFF

Analysis

Custom Polled Instance Summary : 0

Name: deploysx1-vm2
 State: Major
 Last State Change Value: 4,685
 State Last Modified: August 22, 2011 9:29:01 PM MDT
 MIB Expression: sysUpTime
 MIB Instance: .0
 Active State: Active
 Custom Node Collection: deploysx1-vm2
 Policy: sysUpTime
 Collection: DeviceReboot

Custom Poller: Collected Node deploysx1-vm2

DeviceReboot

sysUpTime : 4685

Topology Objects: deploysx1-vm2-.0

Poll End Time: 09:29:01 PM MDT
 Duration (ms): 3
 Interval: 5 minutes

Figure 40: Open Key Incidents

Network Node Manager

Open Key Incidents

Sev	Pri	Life	Last Occurrence	Assigned	Source Node	Source Object	Cat	Fan	Ori	Cor	Message
5	5	5	8/22/11 9:29:16 PM	deploysx1-vm2	sysUpTime	sysUpTime/DeviceReboot for variable sysUpTime (sysUpTime) is in the MAJOR state					
5	5	5	8/22/11 1:44:46 PM	ntc6kgw2	Chassis Fan Tra	Fan on ntc6kgw2 is malfunctioning					
5	5	5	8/22/11 8:45:38 AM	lab-1648-s1	lab-1648-s1	Node Down					
5	5	5	8/22/11 8:45:37 AM	lab-baystack411	lab-baystack410	Node Down					
5	5	5	8/22/11 8:43:42 AM	lab-baystack451	lab-baystack450	Node Down					
5	5	5	8/22/11 8:42:49 AM	nsntc-n3540-5	nsntc-n3540-5	Node or Connection Down					
5	5	5	8/22/11 8:42:23 AM	del3324	del3324	Node Down					
5	5	5	8/22/11 8:31:17 AM	lab-hp4000m-s1	Fan Sensor	Fan on lab-hp4000m-sw1 is malfunctioning					
5	5	5	8/20/11 8:03:24 PM	lab-hp4000m-s1	Fan Sensor	Fan on lab-hp4000m-sw2 is malfunctioning					
5	5	5	8/20/11 11:06:15 AM	lab-2821-lsr2	V122	Interface Down					
5	5	5	8/20/11 11:06:15 AM	lab-2821-lsr2	Po1	Interface Down					

Updated: 8/22/11 09:32:52 PM MDT Total: 34 Selected: 1 Filter: ON

Analysis

Incident Summary : CustomPollMajor

Message: sysUpTime/DeviceReboot for variable sysUpTime (sysUpTime) is in the MAJOR state.
 Severity: Major
 Lifecycle State: Registered
 RCA Active: true
 Source Object: sysUpTime (Custom Node Collection)
 Created/Opened: 8/22/11 09:29 PM (Open for 3.7 minutes)

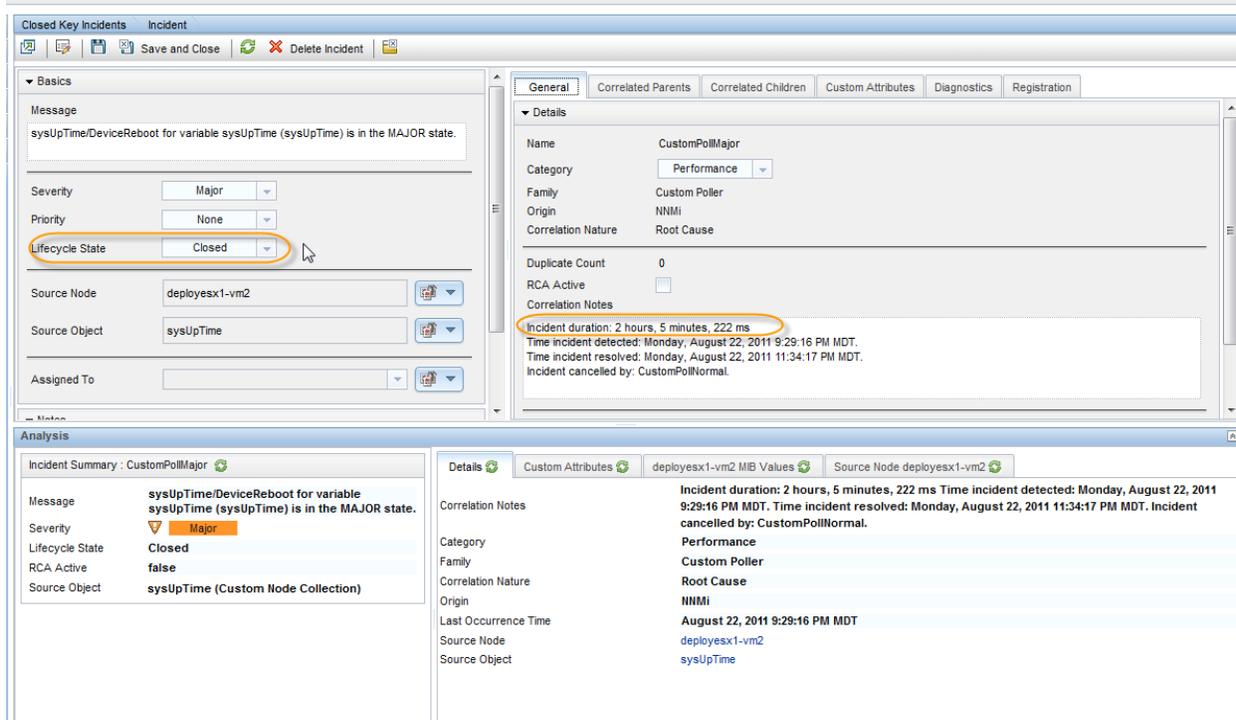
Details

Custom Attributes: deploysx1-vm2 MIB Values, Source Node deploysx1-vm2

Category: Performance
 Family: Custom Poller
 Correlation Nature: Root Cause
 Origin: NNMI
 Last Occurrence Time: August 22, 2011 9:29:16 PM MDT
 Source Node: deploysx1-vm2
 Source Object: sysUpTime

After two hours, you can see that the incident has closed.

Figure 41: Closed Key Incidents



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.

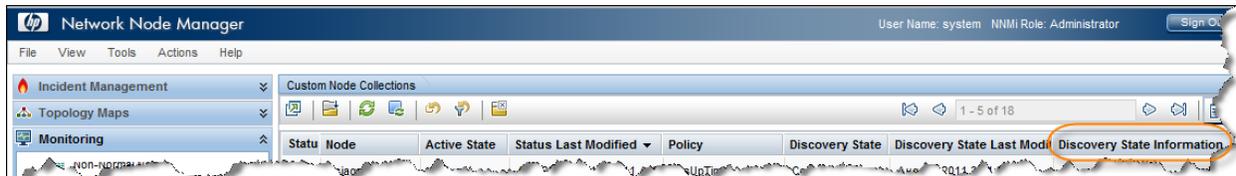
Supported MIB Expression Type	Supported MIB Filter Variable Type
INTEGER, Integer32	INTEGER, Integer32
Unsigned32	Unsigned32
Counter, Counter32, Counter64	Gauge, Gauge32
Gauge, Gauge32	OCTET STRING
TimeTicks	IpAddress
OCTET STRING	

If you set up a Custom Poller Collection using a MIB Expression 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 State Information attribute.

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

Custom Poller log messages are located in the `nnm.*.log` and `nnm-trace.*.log` files.

Figure 42: Custom Node Collections



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 2009–2011 Hewlett-Packard Development Company, L.P.

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

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

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

Oracle and Java are registered trademarks of Oracle 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 America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

Acknowledgements

This product includes software developed by the Apache Software Foundation.

(<http://www.apache.org>)

This product includes software developed by the Indiana University Extreme! Lab.

(<http://www.extreme.indiana.edu>)

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 log on. 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