

HPE Network Node Manager i Software

Managing Traps in NNMi

NNMi Version 10.20

White Paper

Contents

Introduction	
About SNMP Traps and NNMi	
The Flow of SNMP Traps on the NNMi Management Server	
Trap Storage and Logging Stage	7
Trap Storm Detection Stage	7
Trap Filter Stage	7
Trap Forwarding Stage	
Receiver Stage (Event Pipeline)	9
Resolver Stage	
Final Processing Stage	9
Troubleshooting Examples	
Step 1 - Determine why SNMP Traps are not appearing in the NNMi Console	
Step 2 - Manage the Number of Incoming SNMP Traps	
Step 3 - Periodically Trim the NNMi Database of SNMP Traps	
We appreciate your feedback!	

Introduction

Managing SNMP trap configuration within NNMi is a key task that can be used to improve NNMi efficiency. This document gives an overview on the flow of SNMP traps within NNMi, and then steps through the following three tasks for configuring and managing traps with NNMi:

- Determine why SNMP traps are not appearing in the NNMi Console
- Manage the Number of incoming SNMP traps
- · Periodically trim the NNMi database to remove SNMP traps

When following the procedures in this document, note the following:

- The example procedures in this document use a Linux NNMi management server, and can be followed on a Windows NNMi management server.
- The example commands assume the system user is assigned the following password: mypassword
- Some screen captures might be slightly different from those that appear in the most recent NNMi graphical user interface.

About SNMP Traps and NNMi

NNMi works with SNMP traps at the following three levels:

- The nnmtrapreceiver process (running as an operating system's service)
- The NNMi Trap Service (controlled by ov jboss)
- The NNMi Event Pipeline (controlled by ovjboss)

The nnmtrapreceiver process runs as an operating system service and is comprised of the nnmtrapreceivermd and nnmtrapreceiver processes as shown in the following example output:

🛃 root 🕼	😼 root@deploybx2:/var/opt/0V/log/nnm									
[root@	deploylx2	2 nnm]#	p	s -ef	grep	nnmtrap		_		
root	1957	19875	0	16:01	pts/3	00:00:00	grep nnmtrap			
root	12107	11816	0	Mar04	?	00:00:04	nnmtrapreceivermd -ovspmd			
root	12113	1	0	Mar04	?	00:10:29	/opt/OV/bin//nnmtrapreceiver -start			

Note the following:

- The nnmtrapreceivermd process integrates with ovspmd and responds to ovstart/ovstop messages
- During ovstart, the nnmtrapreceivermd process starts the nnmtrapreceiver process
- The ovstop command stops the nnmtrapreceivermd process, but leaves the nnmtrapreceiver process running
- The nnmtrapreceiver process sends traps into the NNMi Trap Service or queues traps until the ovjboss process is started. In both cases, the Trap Service is started.
- The nnmtrapreceiver process is especially useful in high availability (HA) and Application Failover clusters. It runs on BOTH the nodes (active and standby) in the cluster. This means that, during failover, **no trap** is lost.

The NNMi Trap Service maintains a separate data store of traps. This service is controlled by ov jboss and managed through the nnmtrapconfig. ovpl CLI.

The Event Pipeline is visible and controlled through the NNMi console.

As shown in the following diagram, incoming traps are received by the nnmtrapreceiver process. The process sends traps to a message queue, which is processed by the NNMi Trap Service. Each trap moves through various stages in NNMi Trap Service, and, if it passes various filters, eventually reaches the NNMi Event Pipeline.



Figure 1 – High-Level Diagram of SNMP Trap Flow on the NNMi Management Server

It is important to understand the flow of SNMP traps within NNMi Trap Service. The next section describes the details of how NNMi Trap Service and Event Service process SNMP traps at various stages.

The Flow of SNMP Traps on the NNMi Management Server

Figure 2 demonstrates the detailed flow of SNMP traps on the NNMi management server. It shows various checkpoints where a trap can potentially be dropped and where and how to configure filters.





Figure 2 - Diagram of SNMP Trap Flow on the NNMi Management Server

Trap Storage and Logging Stage

In the Trap Storage and Logging Stage, the following actions take place:

- 1. The nnmtrapreceiver process receives incoming traps.
- The nnmtrapreceiver process sends traps to the NNMi Trap Service, which is controlled by ovjboss. If the trap's source IP address, OID, or both are listed in the trapFilter. conf file, the trap is dropped. Otherwise, the trap is stored in the Trap Binary Store and is used in trap analysis (for example, to calculate the trap rate).

Note the following:

- By adding device IP addresses or OIDs to the trapFilter. conf file, you can block these high-volume incidents and avoid incident volume problems.
- The trapFilter.conf is located in <*NnmDataDir*>/shared/nnm/conf directory.
- The trap binary store consists of up to five binary files: traplog0 through traplog4. They are located in the <*NnmDataDir*>/shared/nnm/databases/traps directory.

3. If the trap log is enabled, traps are logged to the trap. csv, trap. log files, or both in the <NnmDataDir>/log/nnm directory.

You can enable/disable trap logging and change the trap logging format by using the nnmtrapconfig. ovpl command:

nnmtrapconfig.ovpl -setProp trapLoggingMode < TXT, CSV, BOTH, OFF>

Trap Storm Detection Stage

Note

This stage is also known as NarrowTrapAnalysis and WideTrapAnalysis.

In the Trap Storm Detection Stage the following events occur:

4. In NarrowTrapAnalysis, the NNMi Trap Service evaluates traps against the trap storm filter specified in the hosted-on-trapstorm. conf file. If the filter detects a hosted-on-trap storm, the trap is dropped.

Note

The hosted-on-trapstorm. conf file is located in the *<NnmDataDir*>/shared/nnm/conf directory.

5. In WideTrapAnalysis, if the NNMi Trap Service detects a trap storm based on the overall trap rate and threshold, the traps are dropped.

Note

To configure the trap storm definition, use the nnmtrapconfig. ovpl command.

Trap Filter Stage

In the Trap Filter Stage, the following events occur:

- 6. If the timestamp of the trap is older than 10 minutes, the trap is dropped.
- 7. If the trap's source IP address, OID, or both are listed in the nnmtrapd. conf, the trap is dropped.

Note the following:

- The nnmtrapd. conf file is located in the <*NnmDataDir*>/shared/nnm/conf directory.
- Traps that meet the following criteria are also evaluated against the nnmtrapd. conf filter:
 - Pass the trapFilter. conf filter
 - Stored in the Trap Binary Store
- If the trap's source IP address, OID, or both are listed in the nnmtrapd. conf file, the trap is dropped here. These traps are blocked from entering the NNMi event subsystem.
- NNMi still uses these traps to calculate the trap rate and to write to the trap binary store. By using the nnmtrapd. conf file approach, you can only stop incidents from being created or stored in the database.
- 8. If undefined traps are accepted, the trap is sent to the next stage—the Trap Forwarding Stage. If the trap is enabled, it also moves to the next stage.

Note the following:

- NNMi drops undefined traps silently by default. You can change the property com. hp. nnm. events. allowUndefinedTraps in the \$NNM_PROPS/nms-jboss.properties file.
- You can use the combination of the nnmloadmib. ovpl and "nnmincidentcfg. ovpl -loadTraps" CLIs to create new trap configuration in NNMi.

The article "Step- by-Step Guide to Incident Management" provides some examples.

9. If the trap is not enabled, it is dropped.

Note

You can enable and disable SNMP traps in the NNMi console.

Trap Forwarding Stage

The Trap Forward Stage is responsible for the following:

10. Traps are forwarded to the NNMi Event Pipeline and to the specified trap forwarding destinations, if any.

Note

You can configure trap forwarding destinations in the NNMi console.



Receiver Stage (Event Pipeline)

In the Receiver Stage, the following events occur:

11. This stage has two receivers. The SNMP Trap Receiver receives traps that are processed and forwarded by NNMi Trap Service. The Global or Regional Incident Receiver receives traps and incidents sent from a Global or a Regional Manager. Both the receivers send traps and incidents to the Resolver.

Note the following:

- Certain conditions might prevent an incident from being forwarded from the Regional Manager. Examples include:
 - If the number of messages in the JMS message queue between the Regional and Global Managers exceeds a configured threshold
 - If the rate of forwarding incidents exceeds a configured threshold.
 - In both cases, the idea of dropping traps is to be able to continue processing management events that are more important.
- If a trap from the Regional Manager is dropped, one of the following incidents is generated:

MessageQueueSizeExceeded incident with WARNING severity MessageQueueSizeExceeded incident with CRITICAL severity MessageQueueIncidentRateExceeded incident

Resolver Stage

In the Resolver Stage, the following events occur:

12. Various conditions are checked in this stage before traps are sent to the Event Pipeline's final processing stage. The evaluation of these conditions and the consequential trap flow are shown in the detailed flow diagrams provided in this section. Note that traps can be dropped in this stage under different configurations.

Note the following:

• In the NNMi console, you can configure whether to discard Unresolved Traps.



- You can configure whether to accept the Undefined Traps in the \$NNM_PROPS/nms-jboss.properties file. See Step 8.
- You can use the combination of the nnmincidentcfgdump. ovpl and nnmincidentcfgload. ovpl CLIs to export, modify, and import the trap and incident configurations.

Final Processing Stage

In the Final Processing Stage, the following events occur:

13. In this stage, first, traps are logged in the incident. csv file. Next, traps are evaluated, enriched, or dropped based on various configurations. These configurations include Interface Group Settings, Node Group Settings, action, enrichment, dampening, and suppression. For example, if the suppression configuration applies, the trap is dropped.

14. Finally, traps are persisted in the NNMi database and forwarded to the NNMi Northbound Interface.

Note the following:

- The incident.csv file is located in the <NnmDataDir>/log/nnm directory.
- NNMi allows 100,000 traps to be stored in the NNMi database. After this limit is reached, no additional traps are stored in the NNMi database.
- You can use the auto-trim oldest SNMP trap incidents feature to control the number of SNMP traps (and syslog messages) stored in the NNMi database and to retain important incoming SNMP traps. For more information, see the "Configuring the Auto-Trim Oldest SNMP Trap Incidents Feature" section of the "Maintaining NNMi" chapter in the HP Network Node Manager i Software Deployment Reference.

NNMi Trap Service also performs a trap analysis task in the background, which is not shown in Figure 2. It uses data from the Tarp Binary Store and generates Trap Analytics Reports the can be viewed from the NNMi console.

ncident Configu	Q Find Node Q Find Attached Switch Port	t	
3 🖻 🛛	mi MIB Browser		
 NNMi Trap Discard Upreso 	NNMi Status <u>S</u> tatus Distribution Graph:	5 🕨	
SNMP Traps an Syslog Message	NNMi Self-Monitoring <u>G</u> ra <u>T</u> rap Analytics	phs 🕨	Recent Top Trap Rate (by Node) (Unlicensed)
▼ Registratic	⊻isio Export ل	•	Recent Top Trap Rate (by SNMP OID) (Unlicensed) Total Traps Received (by Node) (Unlicensed)
Last Modified	Incident Actions Log Security Reports NNMi Audit Loa		Total Traps Received (by SNMP OID) (Unlicensed) Trap Analysis Log (Unlicensed)

The details of trap analytics are also logged in the trapanalytics. log file located in <NnmDataDir>/log/nnm directory.

Troubleshooting Examples

Step 1 - Determine why SNMP Traps are not appearing in the NNMi Console

SNMP traps are being sent to the NNMi management server, but are not displayed in the NNMi SNMP Trap views.

This example uses the following trap and node values:

- SNMP Trap: ciscoSyslogMIBNotification trap (.1.3.6.1.4.1.9.9.41.2.0.1)
- Node: 10. 2. 121. 254

To determine why SNMP traps are not appearing in the NNMi console:

1. Check that the trap service is up and running by using the ovstatus command:

ovstatus -v ovjboss

The output should include the following line to indicate the SNMP Trap Service is running:

NnmTrapService Service is started Check the SNMP Trap Service configuration using nnmtrapconfig.ovpl: nnmtrapconfig.ovpl -showProp -u system -p mypassword

2. Compare your defaults with the default values in the following example nnmtrapconfig.ovpl output:

Note

Many parameters can be tuned by using the nnmtrapconfig. ovpl tool. See the nnmtrapconfig. ovpl Reference Page (Help \rightarrow Documentation Library \rightarrow Reference Pages) for more information

tuanlataufaaa		All interfaces
trapInterTace	·	All interfaces
trapPort	:	162
recvSocketBufSize	:	2048 KBytes
blockTraps	:	true
thresholdRate	:	50 traps/sec
rearmRate	:	50 traps/sec
overalIThresholdRate	:	150 traps/sec
overallRearmRate	:	150 traps/sec
windowSize	:	300 secs
updateSourcesPeriod	:	30 secs
notifySourcesPeriod	:	300 secs
minTrapCount	:	25 traps
numSources	:	10
databaseQSize	:	300000 traps
pipelineQSize	:	50000 traps
databaseFileSize	:	100 MBytes
databaseFileCount	:	5
loopbackAddrOverride	:	Empty

- 3. After you determine that the service is running and configured, check the NNMi Trap Service storage (Trap Store) files to determine whether they are growing in size.
 - a. Navigate to the following directory:
 - /var/opt/OV/shared/nnm/databases/traps
 - b. List the directory contents. For example, on Linux:

|s -|

You should see a maximum of five files: traplog0 through traplog4. You can view the datestamps to see how fast these log files are being updated.

4. Use the nnmtrapdump. ovpl command to verify that the traps you are expecting to view in NNMi appear in the contents of the NNMi Trap Service storage (Trap Store).

Note

It is recommended that you redirect the output. Because the output can be quite large, you might want to limit the amount of traps in the output by using any of the following options: -source, -trapid, -from, and -to.

For our example, we use the -source option with a value of 10.2.121.254 to limit the amount of traps coming from the source device that has an IP address of 10.2.121.254.

nnmtrapdump. ovpl -u system -p mypassword -source¥ 10. 2. 121. 254 > /tmp/trapdump. txt Your output should appear similar to the following example nnmtrapdump. ovpl output. As shown in the example output, each entry begins with the Trap ID, and then provides additional information.

Trap .1.3.6.1.4.1.9.9.41.2.0.1 at October 31, 2008 3:20:30 PM MDT from cisco6509.cnd.hp.com

```
Version: SNMPv1
Enterprise 0ID: .1.3.6.1.4.1.9.9.41.2
Agent address: 10.2.121.254
Generic trap: 6
Specific trap: 1
Timeticks: 1.358,199.695
Varbinds:
<VB state='0' type='4' id='.1.3.6.1.4.1.9.9.41.1.2.3.1.2.2865238' value='SNMP' />
<VB state='0' type='2' id='.1.3.6.1.4.1.9.9.41.1.2.3.1.3.2865238' value='4' />
<VB state='0' type='4' id='.1.3.6.1.4.1.9.9.41.1.2.3.1.4.2865238' value='4' />
<VB state='0' type='4' id='.1.3.6.1.4.1.9.9.41.1.2.3.1.5.2865238' value='AUTHFAIL' />
<VB state='0' type='4' id='.1.3.6.1.4.1.9.9.41.1.2.3.1.5.2865238' value='AUTHFAIL' />
<VB state='0' type='4' id='.1.3.6.1.4.1.9.9.41.1.2.3.1.5.2865238' value='Authentication failure for SNMP req from
host 16.157.128.85' />
<VB state='0' type='67' id='.1.3.6.1.4.1.9.9.41.1.2.3.1.6.2865238'
value='1358199695' />
```

You can conclude that the traps of your interest are arriving at the NNMi Trap Service storage (Trap Store) if you can see them in the output.

Note

After arriving at the Trap Store, if an SNMP trap is filtered out, you cannot see the trap in the NNMi console.

5. Use the nnmtrapconfig. ovpl command to determine whether your expected trap is being blocked:

nnmtrapconfig.ovpl -dumpBlockList -u system -p mypassword

When examining the output, note the following:

- The allowed OIDs are from the SNMP traps configured in the NNMi incident configuration.
- The blocked OIDs are from the nnmtrapd. conf file.
- · Caches are dynamic.
- If a trap comes in and does not pass the filters, its OID is added to the blocked cache.

An abbreviated version of a typical output is shown here.

Note

An easy way to check if a trap is allowed is to look in the first section List of allowed trap OIDs as highlighted in the following example. In our particular case, the OID .1.3.6.1.4.1.9.9.41.2.0.1 is not listed.

Filter Configuration:

List of allowed trap OIDs:

. 1. 3. 6. 1. 4. 1. 9. 9. 13. 3. 0. 6

. 1. 3. 6. 1. 4. 1. 9. 10. 106. 2. 1

. 1. 3. 6. 1. 4. 1. 2272. 1. 21. 41

Blocking Caches:

Cache of blocked OIDs:

. 1. 3. 6. 1. 6. 3. 1. 1. 5. 5. 1. 3. 6. 1. 4. 1. 9. 1. 400

. 1. 3. 6. 1. 2. 1. 47. 2. 0. 1

. 1. 3. 6. 1. 4. 1. 9. 0. 1

Cache of trap OID's blocked from specific source addresses:

In our example, we determined the following:

- The SNMP trap is not blocked by the NNMi Trap Service.
- The SNMP trap is not in the allowed list.

This means that the SNMP trap of interest must be influenced by the NNMi incident configuration.

- 6. Navigate to the Incident Configuration form and look for the SNMP trap of interest:
 - a. From the workspace navigation panel, select the Configuration workspace.
 - b. Expand Incidents.
 - c. Select SNMP Trap Configurations.

Network Node Manager i	Eile View Iools	Agtions Help							User Name: system NNM Role A
lat. Dashboards	SNMP Trap Configuratio	ns X							
e Incident Management	C * * * 0	n 16 🖀	\cap						H 🗲 110-120
📥 Topology Maps	▲ Name	SNMP Object ID	Enabled	Root	Deduplica	Rate	Ser Ca	Fai Author	Message Format
P Monitoring			11	Cause	Enabled	Enabled			
▲ Troubleshooting	STPNewRoot	13.6.1.2.1.17.0.1	- 1	*.	~		* *	R HP Network	STP New Root
Inventory	STPTopologyChange	13.6.1.2.1.17.0.2	-	-	~		4	HP Network	STP Topology Change
Management Mode	SiteScopeAlertEventv1	13614111151401	*			.*	0 .	B HPE SiteSco	Alert "S.1.3.6.1.4.1.11.15.1.3.1.2" was friggered on monitor "S.1.3.6.1.4.1.2"
Q hardest barrens	SiteScopeAlertEventv7	1361411115141	-				0 +	IB HPE SiteSco	Alert 51301+11151317 was triggered on monitor 51301+1.
84 Incident Browsing	Systopressage	13.6141.9.941201						HD Reute A	SLSS S4 (aysing)
👂 Frame Relay	Traffichilabl inki itilization	13616180831112						HP Route Ar	 Traffic high link high high of the set S5 days SA are Tune S13 days Tune S14 +
💉 UI Extensions Test	TrafficLinkCoSUtilization	13614180831112					11	- HP Route Ar	Traffic high mixed in S4 for link src S5 dest S6 srcType S13 destType S
% Integration Module Configuration	TrafficLowLinkUtilization	13.6.1.4.1.8083.1.1.12	- 1					HP Route Ar	Traffic low linkUtil in \$4 for link: src \$5 dest \$6 srcType \$13 destType \$14 fra
F Configuration	TrafficQuantityAlert	13.6.1.4.1.8083.1.1.12	1- 1	-11		-	4	- HP Route Ar	r Traffic quantity alarm: trapInfo \$4 threshold \$5 clearThreshold \$6
Incident Configuration	defa dilocidenti opfinurati Updated: 7/14/16 03:24:0	1.2.3.4 SA77R 1234SA 05 AM	4/	1.62		Total: 132	132	R Customer Se	INCIDENT DISDLAY EMT lected: 1 Filter: OFF
SNMP Trap Configurations	- Analysia		V						
Syslog Message Configurations	* Analysis					_			
Management Event Configurations	SNMP Trap Configurati	on Summary : SyslogMessag	e	3	Details 2	:			
Pairwise Configurations	SNMP Object ID .1.	3.6.1.4.1.9.9.41.2.0.1			Enabled			1	alse
Custom Correlation Configuration	Message Format \$1	:\$3 \$4 (syslog)			Severity			1	Warning
					Family				nterface
Trap Forwarding Configuration					Root Cau	se		1	alse
Trap Logging Configuration				Author Description	on		Y	4P Network Node Manager When a syslog message is generated by the device this notification is sent.	

Note the following:

- If the trap is not loaded, it is filtered out by the NNMi Trap Service. You can load the trap by using the nnmincidentcfg.ovpl loadTraps command.
- If the trap is loaded, check whether the SNMP trap is enabled. If it is not enabled, it is filtered by the NNMi database.

In our case, we see that the trap is loaded.

- 7. Check whether the SNMP trap is enabled:
- 8. Select the SyslogMessage trap configuration.
- 9. Open the SNMP Trap Configuration form for the SyslogMessage trap configuration and verify that the Enable box is selected.

As shown in the following example, the Enable attribute indicates that the SNMP trap configuration is enabled.

Network Node Manager i	<u>File V</u> iew]	ools A <u>c</u> tions <u>H</u> elp		
📶 Dashboards	SNMP Trap Configur	ations × SNMP Trap Configuration * ×		
of Incident Management		1 🛛 2 💼		
🍰 Topology Maps	Name	SyslogMessage	-	< Interface Settings
Gamma Monitoring	The SNMP Object ID	(OID) attribute accepts one wildcard character	(*) that must	-
A Troubleshooting	appear at the end of beginning with .1.3.	the OID specified. NNMi permits wildcards only 6.1.4 (private MIBs). Click here for more inform	y in OIDs ation.	NNMi enables you to apply
S Inventory	SNMD Object ID	1741/100/1001	_	participation in an Interface
🧕 Management Mode	Enabled	.1.3.0.1.4.1.9.9.41.2.0.1	_	
ℜ Incident Browsing	Root Cause			▲ Interface Group Orderi
💋 Frame Relay	 Category 	Fault	▽ 👹 ▾	
💋 UI Extensions Test	Family	Interface	▼ 🗊 ▼	
𝗞 Integration Module Configuration	* Severity	Warning 🗢		
🗲 Configuration	Specify how the loci	dent message appears in the Incident view. To i	include Incident	
Incident Configuration	information in the m	essage use \$(variable_name). Select these varia	ables from a set of	
SNMP Trap Configurations	 Analysis 			

- 10.Save and close the SNMP Trap Configuration form.
- 11.Save and close the Incident Configuration form.
- 12. To confirm that the changes have taken effect, use the nnmtrapconfig. ovpl tool to view the list of allowed SNMP trap OIDs: nnmtrapconfig. ovpl -dumpBlockList -u system -p mypassword

You should see the OID for the SNMP trap that you just enabled.

- List of allowed trap OIDs:
- . 1. 3. 6. 1. 4. 1. 9. 9. 13. 3. 0. 6
- . 1. 3. 6. 1. 4. 1. 9. 10. 106. 2. 1
- . 1. 3. 6. 1. 4. 1. 2272. 1. 21. 41
- . 1. 3. 6. 1. 4. 1. 9. 9. 41. 2. 0. 1

13. To make sure that the traps are arriving, navigate to the SNMP Traps view:

- a. From the workspace navigation panel, select the Incident Browsing workspace.
- b. Select SNMP Traps.

You should begin to see the traps.

As shown in the following example, the top of the **SNMP Traps** view includes a "time" filter and a Node Group filter. Make sure that these are set appropriately for your testing. Sometimes these filters are set to show only the last hour or a certain group of nodes that does not include the node of interest.

Network Node Manager i	<u>F</u> ile <u>V</u> iew <u>T</u> ools)	A <u>c</u> tions <u>H</u> elp			User Name: system NI
ull Dashboards	SNMP Traps ×				
ه» Incident Management	ଟା 🗃 ଟା 🦘 🖒	Î			All 🗢 <empty filter="" group=""> 🗸</empty>
📥 Topology Maps	Severity Lifecycle State	▼ Last Occurrenc Source Node	Source Obj	ect Cat Fan Cor Message	Notes
Monitoring	ن 🔺	7/14/16 11:16:4 Somerset7200	none	🔹 🗸 🌬 System:11424 This is the text	that make
A Troubleshooting	ى 🕹	7/14/16 11:16:4 joOmcswd	none	👙 📰 🐜 System:11424 This is the text	that make
—	ے ا	7/14/16 11:16:4 hqcndlr-06-sw07	none	🜞 🌆 🐜 Authentication failure	
E Inventory	ය 🔺	7/14/16 11:16:4 MA02BFSH	none	🜞 釋 🎭 OSPF Virtual Interface State C	hange: Roi
🧕 Management Mode	. 4	7/14/16 11:16:4 Chesilhurst-0409-SW1	none	🗯 🚟 🐜 System:11424 This is the text	that make
Q Incident Browsing	ى 🕹	7/14/16 11:16:4 MA02BGSA	none	🗯 🌆 🌬 Authentication failure	
Open Key Incidents	بة 🔺	7/14/16 11:16:4 ny-elpp-dhp01	none	👙 🛱 🐜 System:11424 This is the text	that make
	S 8	7/14/16 11:16:4 MA02BFSH	none	🗯 📇 🐜 RMON Falling Alarm: 23 fell be	elow threst
	S 🗢	7/14/16 11:16:4 MA02BFSH	none	🍁 🚟 🍻 BGP Established: State .1.3.6.	1.2.1.15.0
Open Root Cause Incidents	ē,	7/14/16 11:16:4 MA02BFSH	none	🜞 📰 🐜 BGP Backward Transition: Stat	te .1.3.6.1.
Service Impact Incidents	ے ا	7/14/16 11:16:4 ei0sdfsl	none	🔹 📇 🐜 System:11424 This is the text	that make
III All Incidents	_ة <u>ا</u>	7/14/16 11:16:4 go04afsg	none	🜟 🌆 🐜 Authentication failure	
Custom Open Incidents	.⊿ ē,	7/14/16 11:16:4 ITO-Warehouse	none	🔹 📇 🐜 System:11424 This is the text	that make
Custom Incidents	ى 🕹	7/14/16 11:16:4 ma18cfsh	none	🔹 🚟 🍻 System:11424 This is the text	that make
NNM 6 x/7 x Events	L 2 L	7/14/16 11:16:4 NPID31686	none	🗯 鐸 🐜 OSPF Virtual Interface State C	hange: Rou
Europe Manager	L	7/14/16 11:16:4 Wall_TWP_1352	none	🗯 🌆 🐜 Authentication failure	
	S 📀	7/14/16 11:16:4 NPID31686	none	🗯 🚟 🐜 BGP Established: State .1.3.6.	1.2.1.15.0
III SNMP Traps	<u>5</u>	7/14/16 11:16:4 Perth-Amboy-SW1-basement	none	🔹 📇 🎭 System:11424 This is the text	that make
	Updated: 7/15/16 12:26:21 AM			Total: 1611 Selected: 0) Filter: ON

Step 2 - Manage the Number of Incoming SNMP Traps

This example steps through troubleshooting the following scenario:

You are receiving too many traps into the NNMi console and you want to reduce the number of incoming traps.

Analysis done against the SNMP Trap Service storage (Trap Store) can be different than what is shown in the NNMi console, which reflects the NNMi database contents. This scenario uses both the SNMP Trap Service storage (Trap Store) and the NNMi database. Therefore, carefully check whether the step described is using data from the NNMi database or from the SNMP Trap Service storage (Trap Store).

When working with SNMP trap limits, note the following:

- NNMi allows 100,000 traps to be stored in the NNMi database. The 100,000 trap limit applies only to the NNMi database. After this limit is reached, no additional traps are stored in the NNMi database.
- The NNMi Trap Service storage (Trap Store) continues to receive traps after the 100,000 SNMP Trap limit is reached. When the NNMi Trap Service storage has "filled up", it uses the FIFO (First In, First Out) model, deleting the oldest traps and saving the newest ones.

When using results from the NNMi database, note the following:

- Although the NNMi console provides a visual inspection of the traps stored in the database, use nnmtrapdump. ovpl and nnmtrapconfig. ovpl to analyze the data as described in this section of the document.
- Use caution when viewing the SNMP trap information in the NNMi console. The Source Node displayed in the NNMi console is not necessarily the SNMP trap source address. Instead, the trap could be coming from any address on the Source Node.

NNMi generates an incident when you are near or at the limit of total SNMP traps allowed in the NNMi database as shown in the following example:

当 💼 🧏 Save and	Close K Delete Incident		<u>A</u>		Incide
3asics			General Correlated P	arents Correlated Children Custom Attributes	1
Message Number of snmp traps p exceeded the maximum	persisted in the database has reached or allowed limit of 100000, Current number:		Diagnostics Registrat		-
100000. Snmp traps will Please reduce the numb nnmtrimincidents.ovpl.	no longer be accepted by the event system. er of snmp traps within the database using		Name Category	SnmpTrapLimitCritical Application Status	
Severity Priority	Critical V		Family Origin Correlation Nature	Trap Analysis Management Software Root Cause	
5ource Node	Registered none		Duplicate Count RCA Active	0	
Source Object	none 📰 🖈		Correlation Notes		
Assigned To	- 10 ·				
lotes			First Occurrence Time Last Occurrence Time	02 November 2008 20:37:56 o'clock EET 02 November 2008 20:37:56 o'clock EET 03 November 2008 20:37:56 o'clock EET	
Notes		-		UZ NOVEMBER ZUUB ZU:37:56 O'Clock EET	

See Step 3: Periodically Trim the NNMi Database of SNMP Traps for information about how to remove traps from the database.

To manage the number of SNMP traps, start by determining which nodes are generating most of the traps.

To determine which nodes are generating most of the traps, begin by viewing information in the Custom Attributes tab for the SNMP Trap configuration of interest. The Custom Attributes tab is an excellent source of information about a particular trap in the NNMi console. This tab shows all of the information about a selected trap, including the Source Address as shown in the following example:

<u>F</u> ile <u>V</u> iew	Tools Actions	Help									User N	ame: system NN	IMi Role: Administ	rator	Log OL
SNMP Traps 🕷 📕	ncident ×														
	R 🕄										-				
 Basics 				Genera	Cor	rrelated	Parents	Cor	related Children Cus	tom Att	ributes Diagnos	tics Registratio	on		
Message				-	_										
System:11424 Thi	s is the text that mal	kes up the syslog message. (syslog)		NNMi lis Attribut	sts the (e table,	Custom . click th	Attribute e Restore	s for ir Defau	ncidents in the order in v Ilt Settings icon to resto	vhich th re the C	ey are received from Sustom Attribute ord	n the SNMP trap. Ier for the selecte	lf you sort or filter d incident.	the C	ustom
* Severity	Warning \bigtriangledown			C.		C	•					М	← 1-9 of 9	> 1	
* Priority	None 🔝			Name			Ту	pe	Value						
 Lifecycle State 	Registered \bigtriangledown			.1.3.6	1.4.1.9	9.9.41.1	.2.3.1 as	n_octe	System						
				.1.3.6	1.4.1.9	9.9.41.1	2.3.1 as	n_inte	4						
Source Node	Somerset7200		🗊 🔻	.1.3.6	1.4.1.9	9.9.41.1	2.3.1 ası	n_octe	11424						
Source Object	none			.1.3.6	1.4.1.9	9.9.41.1	2.3.1 as	n_octe	This is the text that ma	kes up	the syslog message.				
				.1.3.6	1.4.1.9	9.9.41.1	2.3.1 as	n_tim	112459						
Assigned To			▽ 📸 ▼	.1.3.6	1.4.1.1	1.2.17	2.2.0 as	n_ipac	44.1.4.35						
				cia.snr	npoid		Str	ring	.1.3.6.1.4.1.9.9.41.2.0	0.1					
 Notes 				cia.ado	lress		Str	ring	44.1.4.35						
Notes				cia.ori	ginalado	dress	Str	ring	127.0.0.1						
				Updat	ed: 7/1	.5/16 12	2:17:20 A	M	Tota	1:9	Selected: 0	Filter: OFF	Au	ito refr	esh: OFF

Tip: When looking at the SNMP Traps view, it can be difficult to get a sense of which nodes are generating most of the SNMP traps. You can try the filters provided to see which Source Node is generating most of the SNMP traps.



You can also use the nnmtrimincidents. ovpl command to export the trap information from the NNMi console to a comma-separated file for easier analysis. This file can then be imported into a tool such as Microsoft Excel.

The following example generates a file with all traps greater than one day old that are in the NNMi database (not the NNMi Trap Service storage (Trap Store)):

nnmtrimincidents.ovpl -age 1 -incr days -origin SnmpTrap -archiveOnly -u system -p mypassword

Note

The -archive0nly option does NOT trim incidents from the database.

When importing the file into Excel, note the following:

- Before importing the file, delete the first few lines of the output file that describe the file contents. (These lines begin with #.)
- Specify that the file is comma-separated.

An example output file that can be used for analysis is shown below:

	A	B	C	D	E	F	G	H		J	K	L	M
23	8 Wed Feb 25 04:49:15 MST 2009	192.25.203.67	none	Cisco incorrect community name (SNMP authenticationFail	Registere	WARNING	None	null	null	Status	Node	Symptom	null
23	9 Tue Feb 24 23:34:14 MST 2009	192.25.203.67	none	Cisco incorrect community name (SNMP authenticationFail	Registere	WARNING	None	null	null	Status	Node	Symptom	null
24	0 Tue Feb 24 23:34:16 MST 2009	192.25.203.67	none	Cisco incorrect community name (SNMP authenticationFail	Registere	WARNING	None	null	null	Status	Node	Symptom	null
24	1 Tue Feb 24 23:54:17 MST 2009	192.25.203.67	none	Cisco Agent Interface Up (linkUp Trap) on interface 4	Registere	NORMAL	None	null	null	Fault	Interface	Symptom	null
24	2 Tue Feb 24 23:54:19 MST 2009	192.25.203.67	none	Cisco Agent Interface Up (linkUp Trap) on interface 4	Registered	NORMAL	None	null	null	Fault	Interface	Symptom	null
24	3 Tue Feb 24 23:54:25 MST 2009	192.25.203.67	none	Cisco Agent Interface Up (linkUp Trap) on interface 4	Registere	NORMAL	None	null	null	Fault	Interface	Symptom	null
24	4 Tue Feb 24 23:54:45 MST 2009	192.25.203.67	none	Cisco Agent Interface Down (linkDown Trap) on interface 4	Registered	MINOR	None	null	null	Fault	Interface	Symptom	null
24	5 Wed Feb 25 03:28:26 MST 2009	192.25.203.67	none	Cisco incorrect community name (SNMP authenticationFail	Registere	WARNING	None	null	null	Status	Node	Symptom	null
04	C MILL ELL OF 02-20-24 MOT 2000	100 05 000 07		Cines in second company the second (CNIMD authorities Fail	Desisters	MALA DAUNIC	Mana	and the second s	and the second s	Chatting	Mada	Constant	and the second s

Alternatively, you can work with the archive file from the command line. For example, to get a list of the trap OIDs:

cut -f 20 -d ',' incidentArchive.txt | sort -rn | uniq -c

In the example output below, the left column represents the count.

 $254 \ . \ 1. \ 3. \ 6. \ 1. \ 6. \ 3. \ 1. \ 1. \ 5. \ 5. \ 1. \ 3. \ 6. \ 1. \ 4. \ 1. \ 9$

- 3 . 1. 3. 6. 1. 6. 3. 1. 1. 5. 4. 1. 3. 6. 1. 4. 1. 9
- 8 . 1. 3. 6. 1. 6. 3. 1. 1. 5. 3. 1. 3. 6. 1. 4. 1. 9
- 4 . 1. 3. 6. 1. 6. 3. 1. 1. 5. 1. 1. 3. 6. 1. 4. 1. 9

From this example, you can see that you might want to disable the Cisco Authentication Failure trap (.1.3.6.1.6.3.1.1.5.5.1.3.6.1.4.1.9).

To get a list of the source addresses and occurrence count:

cut -f 23 -d ',' incidentArchive.txt | sort -rn | uniq -c

From the example output below, you can see that you might want to block traps from the node 10.25.203.67.

269 10. 25. 203. 67

10 10. 18. 10. 2

The next two sections — "Blocking Traps Using the Node IP Address" and "Blocking Traps Using the Trap OID" — describe how to block traps using the nnmtrapd. conf file.

For the remainder of this example, instead of working with the SNMP trap information in the NNMi database, we analyze the SNMP traps in the NNMi Trap Service storage (Trap Store).

Advantages of analyzing the SNMP traps in the NNMi Trap Service storage include the following:

- The tools provide information that you cannot obtain using the NNMi database.
- You can determine which nodes and traps are the most verbose (even when they are filtered out at the NNMi Incident level).
- If you can discard the trap at the NNMi Trap Service level, NNMi performs more efficiently.

A disadvatage of using the NNMi Trap Service data is that you might get information about SNMP traps that are dropped before arriving at the NNMi console.

Tip: When using the NNMi Trap Service data, you can analyze your existing SNMP Trap information by using the node's IP Address or the SNMP Traps OID.

Blocking Traps Using the Node IP Address

- 1. Obtain a list of all the Source Addresses for SNMP traps in the NNMi Trap Service storage (Trap Store).
- 2. Send the contents of the NNMi Trap Service storage (Trap Store) to a file by using the nnmtrapdump. ovpl tool:

nnmtrapdump.ovpl -u system -p mypassword > /tmp/trapdump.txt

3. Find all occurrences of "Agent address" and delete extra words so that only the IP addresses are included. For example, on Linux:

grep "Agent address" /tmp/trapdump.txt | cut -f 3 -d ' ' > /tmp/a

4. Sort the IP addresses, obtain a unique count for each IP address, and then sort the results in order of the most occurrences to least occurrences. For example, on Linux:

sort -rn /tmp/a | uniq -c | sort -rn > /tmp/trapcount.txt

The file /tmp/trapcount.txt should look similar to the following.

Tip: The column on the left is the unique count. The column on the right is the source address.

60109 10. 168. 201. 16

5064 10. 98. 112. 2 5061 10. 98. 110. 2 1089 10. 130. 252. 90 63 10. 168. 129. 19 52 10. 104. 252. 7 36 10. 106. 252. 19 36 10. 103. 252. 18 16 10. 108. 130. 37 16 10. 102. 252. 14 12 10. 105. 252. 8 12 10. 102. 130. 12

From this output, you can see that most traps are coming from the node 10.168.201.16.

5. Block the traps from node 10.168.201.16 by editing the blocking filters in the file nnmtrapd. conf. The location of this file is:

Linux:

/var/opt/OV/shared/nnm/conf/nnmtrapd.conf

Windows:

data_dir¥shared¥nnm¥conf¥nnmtrapd.conf

This file contains examples and documentation.

Tip: When working with your own data, before blocking traps from the node, first evaluate the node and its traps and determine whether it is useful to retain the traps.

6. To block all traps using the node's address, add the following line to the end of the nnmtrapd. conf file.

<10. 168. 201. 16, . *>

Tip: Notice the use of wildcards (.*) to catch all traps from this source address.

7. Use the nnmtrapconfig. ovpl command to indicate the NNMi Trap Service should read the filter file again:

nnmtrapconfig.ovpl -readFilter -u system -p mypassword

8. Confirm that the change took effect:

nnmtrapconfig.ovpl -dumpBlockList -u system -p mypassword

You should see the following entry to verify that the traps are blocked from this source address:

List of blocked source addresses:

10.168.201.16

Blocking Traps Using the Trap OID

You can perform a similar task with trap OIDs rather than source addresses.

1. Run the following commands to determine the highest number of SNMP traps per OID:

nnmtrapdump.ovpl -u system -p mypassword > /tmp/trapdump.txt

grep Trap /tmp/trapdump.txt | cut -f 2 -d ' ' > /tmp/b

sort -rn /tmp/b | uniq -c | sort -rn > trapOID.txt

For example, we determine that the trap OID .1.3.6.1.2.1.10.21.2.0.1 is occurring frequently and is filling the NNMi console with traps, and therefore, we do not want to receive this trap.

You can block the trap in one of the following ways:

• Use the NNMi console

Assuming you have this trap defined in NNMi, navigate to the incident configuration view, find the definition of this trap, and then uncheck the Enable box as described in **Step 1: Determine why SNMP Traps are not appearing in the NNMi console**.

• Use the nnmtrapd. conf file

The NNMi Trap Service storage (Trap Store) continues to receive the trap even after blocking the trap by using this method.

An advantage of this method is that you could consider only blocking this trap when it originates from a specific IP Address. This is not possible in the NNMi console.

For example, if you wish to block this trap when it is received from any address in the subnet 10.6.3.*, edit the nnmtrapd. conf file to include the following line:

<10. 6. 3. *, . 1. 3. 6. 1. 2. 1. 10. 21. 2. 0. 1>

Tip: You can also mix and match IP address ranges and wildcards as well as OID ranges and wildcards.

Additional Tips for Managing Your SNMP Trap Numbers

A common error when working with the nnmtrapd. conf file is to have two lines that match the same IP address. The parser works from the top to the bottom. The parser stops when it finds an IP address match.

For example, only the first line is effective if the nnmtrapd. conf file contains the following two lines:

<10. 6. 3. *, . 1. 3. 6. 1. 2. 1. 14. 16. 0. 6>

<10. 6. 3. *, . 1. 3. 6. 1. 2. 1. 10. 21. 2. 0. 1>

Because the node address matches this first line, the second line is never matched.

To enable nnmtrapd. conf to match each specified filter, use a syntax similar to the following:

<10. 6. 3. *, . 1. 3. 6. 1. 2. 1. 14. 16. 0. 6, . 1. 3. 6. 1. 2. 1. 10. 21. 2. 0. 1>

As a general rule, use the NNMi console to block traps unless you only want to block traps from specific IP addresses.

To ensure all of the specific filters are enabled, include all the specific filters above a general "catch all" filter at the bottom of the nnmtrapd. conf file. For example, if the trapd. conf file included an OID filter in our previous example, and it occurred below the IP address filters in the trapd. conf file, the OID filter might never be matched.

If you have the MIB loaded in the NNMi database for the traps that you wish to block, you could load the trap definition, and then block the trap by simply disabling the trap in the NNMi console. Using the NNMi console does not reduce performance and you might find it easier to manage.

The benefit of using the nnmtrapd. conf file is that you get finer granularity for controlling traps because you can use the wildcard character and specify nodes and SNMP trap OIDs together.

Note

SNMP traps appear in the NNMi Trap Service storage even if they are blocked. Therefore, ignore statistics on traps that you know you are blocking.

Step 3 - Periodically Trim the NNMi Database of SNMP Traps

This section explains how to use the nmtrimincidents. ovpl command to trim the database of traps that are no longer needed. This practice helps you avoid reaching the trap limit in the NNMi database.

Note

The nmtrimincidents. ovpl command applies to the NNMi database and not the NNMi Trap Service storage (Trap Store).

The nnmtrimincidents.ovpl command has a rich set of options to help you trim the database.

Run the nnmtrimincidents.ovpl command regularly in a cron job (Linux) or a scheduled task (Windows).

The following command deletes any traps that are older than one week:

nnmtrimincidents.ovpl -u system -p mypassword -age 1 -incr weeks -origin SnmpTrap -trimOnly -quiet

You can adjust the example to fit your trap rate.

When using nnmtrimincidents.ovpl, note the following:

- The example options do not keep an archive of the trimmed traps. You can easily do this by using the -archiveOnly option. See the nnmtrimincidents.ovpl reference page (Help --> Documentation Library --> Reference Pages) for more information.
- The -quiet mode does not prompt for user confirmation. (This option is ideal for a cron job.)

We appreciate your feedback!

If an email client is configured on this system, by default an email window opens when you click here.

If no email client is available, copy the information below to a new message in a web mail client, and then send this message to **network-management-doc-feedback@hpe.com**.

Product name and version: NNMi 10.20

Document title: Managing Traps in NNMi

Feedback:



[©] Copyright 2016 Hewlett Packard Enterprise Development LP. The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise 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. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.