How to Monitor Windows Event Logs Using SNMP Traps

Purpose

This document describes how to monitor Windows Event Logs via SNMP within Nagios XI using the eTrap agent from the Nagios Exchange. This allows users to quickly and easily be alerted to real-time network events and incidents taking place on their network, specifically their Windows machines.

Prerequisites

This document assumes you have the **snmptt** add-on already running and installed on your Nagios XI server. If you are not currently processing SNMP traps within Nagios XI please refer to the document below for instructions on how to integrate SNMP Traps with Nagios XI.

Editing Files

In many steps of this documentation, you will be required to edit files. This documentation will use the vi text editor. When using the vi editor:

- To make changes press `i` on the keyboard first to enter insert mode
- Press **Esc** to exit insert mode
- When you have finished, save the changes in vi by typing `:wq` and press **Enter**

Download eTrap Application

The eTrap agent is a community contribution, and can be found on the Nagios Exchange Site.

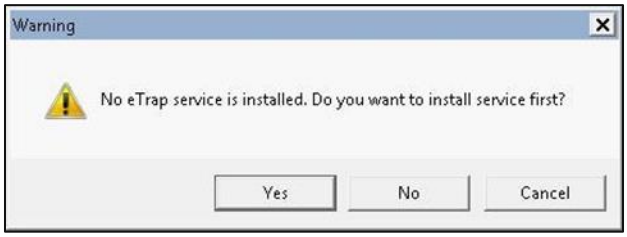
You will need to log into your Windows machine as an administrator and download the etrap_configurator.exe file to your Windows machine.

Install eTrap

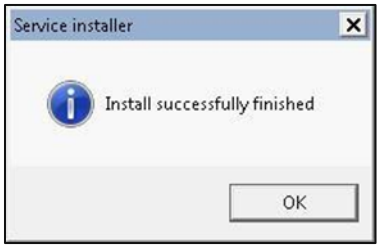
Double click the **etrap_configurator.exe** file to begin the installation.

You may be prompted by User Account Control for permission, click **Yes** to proceed.

The installer will detect that the eTrap service is not installed, click **Yes** to perform the installation.



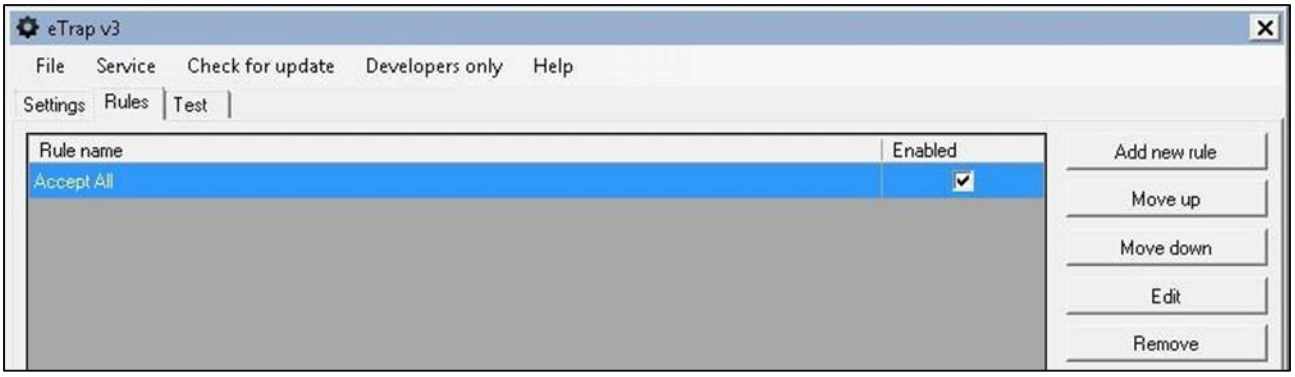
The installation will take no time, and you will be presented with a message saying it was successful. Click **OK**.



The eTrap application will now open ready to be configured.

ETrap Initial Configuration

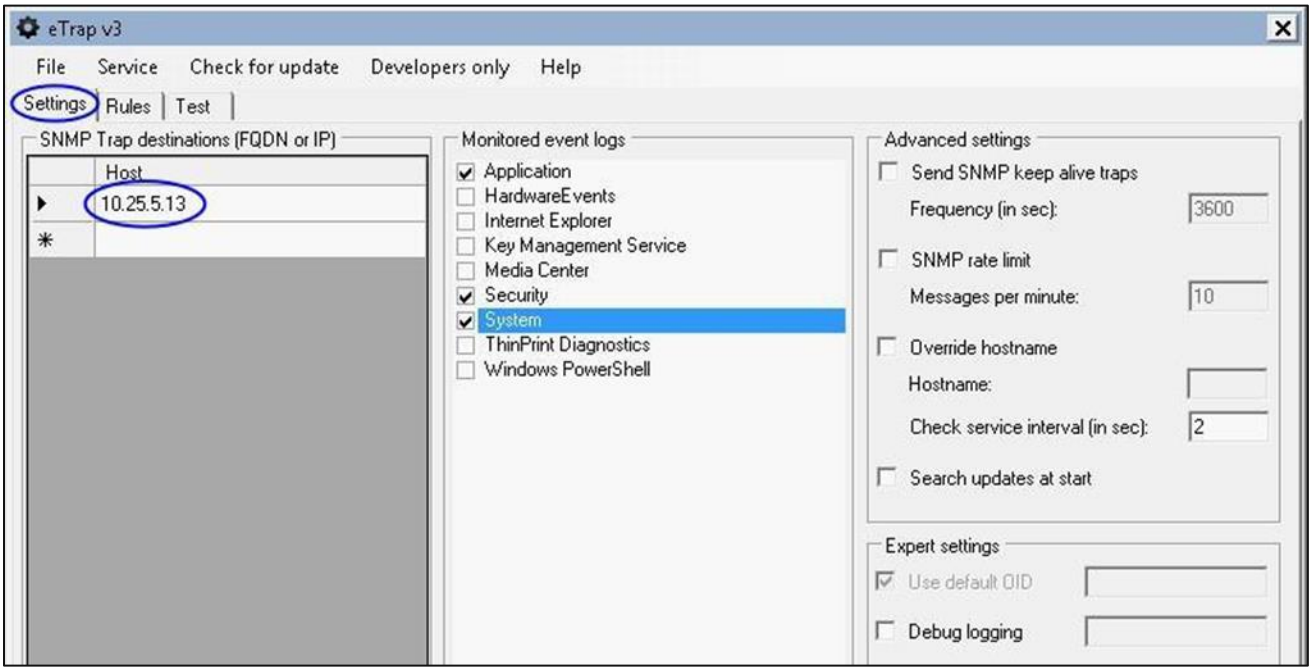
When the eTrap application is opened you will be presented with the default configuration on the Rules tab.



Click the **Settings** tab, here you can add the address of your Nagios XI server that you want to send traps to.

You can also see that the **Application**, **Security** and **System** logs have been checked (required).

Click **File** > **Save and restart service** after making the required changes and then click **OK** on the two success messages.



Now that you have eTrap configured to send to Nagios XI you can send a test trap to ensure that it can communicate. Click the **Test** tab and click the **Send** button. You will be informed that the message was successfully sent.

Establish a terminal session to your Nagios XI server and execute the following command:

```
tail /var/log/snmptt/snmpttunknown.log -n 20
```

You should see the following information about the test trap:

```
Thu Jun 1 13:20:28 2017: Unknown trap (.1.3.6.1.4.1.29037.8.9.0.1) received
from win7-02 at:
Value 0: win7-02
Value 1: 10.25.14.3
Value 2: 0:0:00:29.59
Value 3: .1.3.6.1.4.1.29037.8.9.0.1
Value 4: 10.25.14.3
Value 5: public
Value 6: .1.3.6.1.4.1.29037.8.9
Value 7:
Value 8:
```

```
Value 9:
Value 10:
Ent Value 0: .1.3.6.1.4.1.29037.9.1=win7-02
Ent Value 1: .1.3.6.1.4.1.29037.9.2=WindowsEvent
Ent Value 2: .1.3.6.1.4.1.29037.9.3=1
Ent Value 3: .1.3.6.1.4.1.29037.9.4=2017-6-01,13:20:28.2 Ent Value 4:
.1.3.6.1.4.1.29037.9.5=For Advanced license visit
http://www.smartoservice.com/licenses - TrapEvent:19100:Test message
This confirms that eTrap was able to sent a trap to your Nagios XI server.
The Ent Value 0 - 4 entries are what contains the important information,
from the documentation they are:
1:Source – The hostname of the sender host (ex.: winserver.contoso.com)
2:Service – The name of the service the trap is sent (ex.: WindowsEvent)
3:Severity – The severity of the windows event (OK or Critical or Error)
4:TimeStamp – The time stamp of the windows event
5:Info – A concatenated string. (Containing the windows event source, event
ID and event message if Advanced mode is used).
```

For purposes of clarity, the Ent Value 0 - 4 entries are tied to these in the documentation:

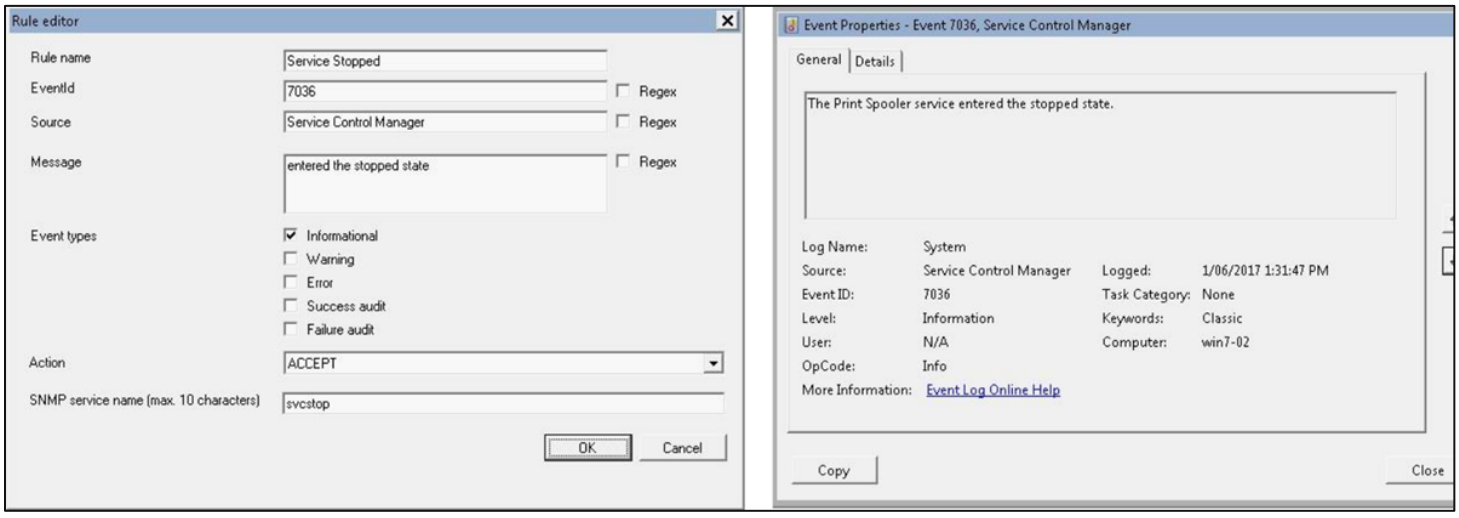
```
$1 = Ent Value 0 = 1:Source
$2 = Ent Value 1 = 2:Service
$3 = Ent Value 2 = 3:Severity
$4 = Ent Value 3 = 4:TimeStamp
$5 = Ent Value 4 = 5:Info
```

The $1, $2, etc values are how they are referenced in the snmptt configuration, which will be explained later.

At this point the Nagios XI server is not doing anything with the trap as snmptt has not been configured to send the trap to Nagios XI.

eTrap Rules

Rules are what make eTrap send SNMP Traps to your Nagios XI server. If you click the **Rules** tab in eTrap you can see there is a default Accept All rule. Select it and click the **Edit** button. You can see that this rule is basic, it will only send traps for Warning and Error logs in the event logs (Application, Security and System as this was what was defined on the Settings tab). The following example will demonstrate how to create a rule that will send a trap when a service is stopped, the Print Spooler service will be used as an example. Click the **Add new rule** button and populate the rule as per the following screenshot on the left. The screenshot to the right shows an example entry of what information is logged when a service is stopped.



You can see from the rule being created that the event type is **Informational**. Windows logs a service stop event as informational, so this needs to be accounted for. The **EventId** is defined as well as **Source** and **Message**. The action is ACCEPT and the SNMP service name is svcstop.

Click the **OK** button to create the rule and then click **File > Save and restart service** to make the rule active.

When you manually stop the Print Spooler service you'll see the following logged in `/var/log/snmptt/snmpttunknown.log` on your Nagios XI server:

```
Ent Value 0: .1.3.6.1.4.1.29037.9.1=win7-02
Ent Value 1: .1.3.6.1.4.1.29037.9.2=svcstop
Ent Value 2: .1.3.6.1.4.1.29037.9.3=0
Ent Value 3: .1.3.6.1.4.1.29037.9.4=2017-6-01,13:43:48.6 Ent Value 4:
.1.3.6.1.4.1.29037.9.5=For Advanced license visit
```

```
http://www.smartoservice.com/licenses - Service Control Manager:7036:The
Print Spooler service entered the stopped state.
```

Additionally, if you manually start the Print Spooler service you'll see that nothing gets logged in `/var/log/snmptt/snmpttunknown.log` on your Nagios XI server, this is because no rule exists that matches a service start event. Now that the Nagios XI server is receiving the trap, the next step is to configure snmptt to send the trap to Nagios XI.

SNMPTT Trap EVENT

An EVENT in the snmptt.conf file is how a trap is sent to Nagios XI. On your Nagios XI server open the `/etc/snmp/snmptt.conf` file in vi using the following command:

```
vi /etc/snmp/snmptt.conf
```

Type (or paste) the following lines (it doesn't matter if they are at the top or bottom):

```
EVENT event .1.3.6.1.4.1.29037.8.9.0.1 "Status Events" CRITICAL
FORMAT Service stopped event: $-* EXEC /usr/local/bin/snmptraphandling.py
"$r" "SNMP Traps" "$s" "$@" "$-*" "Service Stopped: $5"
MATCH $2: (svcstop)
```

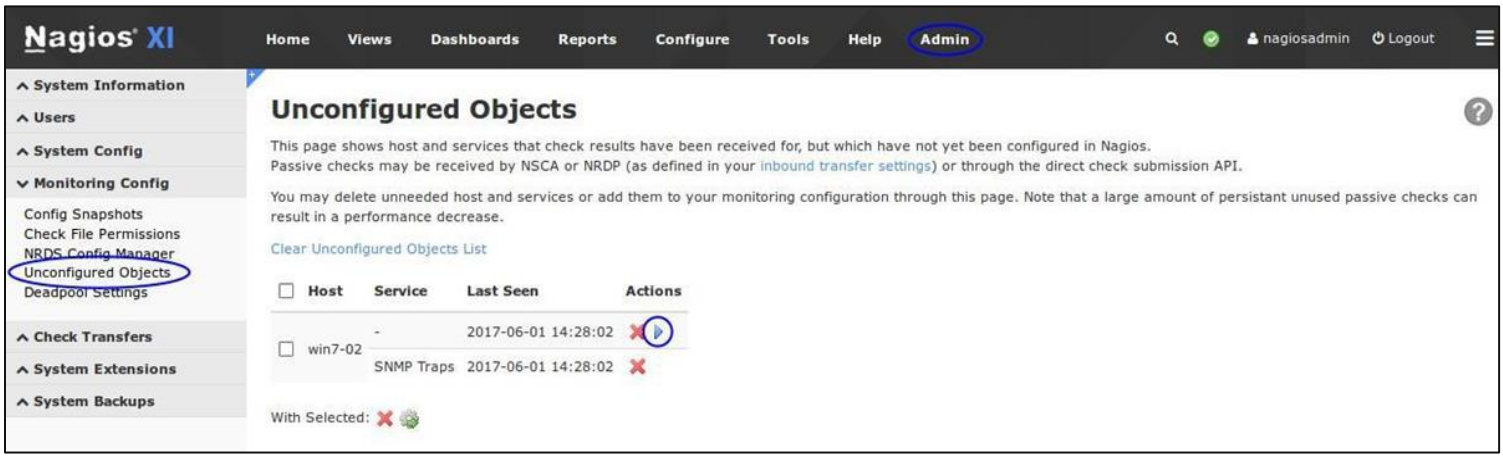
All traps come in on the .1.3.6.1.4.1.29037.8.9.0.1 OID. To differentiate between different trap types a MATCH line is used, which in this case is looking for the value of svcstop in the $2 variable. This allows the trap to be defined as critical, so it gets sent through to Nagios XI as a critical state. Save the snmptt.conf file and restart the snmptt service:

```
service snmptt restart
```

Now you can go back to your Windows machine and start / stop the Print Spooler service. You will notice that this time nothing is logged in the `/var/log/snmptt/snmpttunknown.log` file, instead it is logged in `/var/log/snmptt/snmptt.log`, it will be a lot of information like:

```
Thu Jun 1 14:27:51 2017 .1.3.6.1.4.1.29037.8.9.0.1 CRITICAL "Status Events"
win7-02 - Service stopped event: enterprises.29037.9.1 ():win7-02
enterprises.29037.9.2 ():svcstop enterprises.29037.9.3 ():0
enterprises.29037.9.4 ():2017-6-01,14:27:51.0 enterprises.29037.9.5 ():For
Advanced license visit http://www.smartoservice.com/licenses - Service
Control Manager:7036:The Print Spooler service entered the stopped state.
```

Open Nagios XI and navigate to **Admin > Monitoring Config > Unconfigured Objects**.



You will see the trap received for the SNMP Traps service it was sent to. It exists under Unconfigured Objects because Nagios XI does not yet know about this service. To create it click the play icon to start the

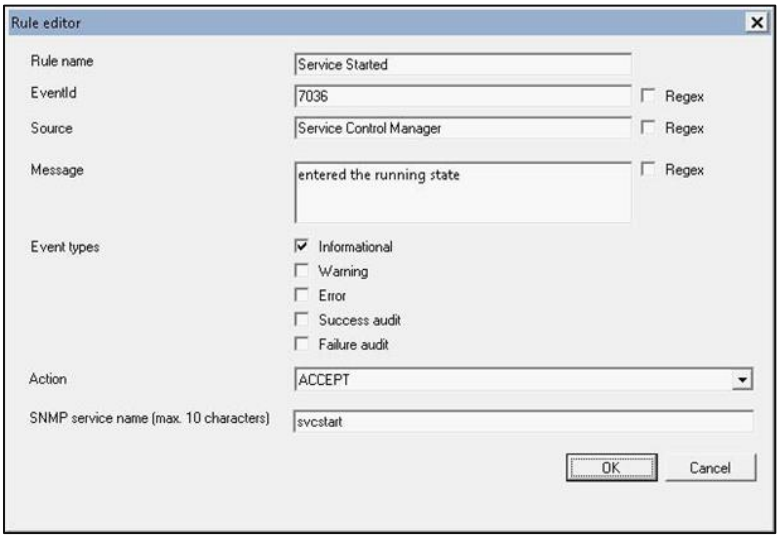
Unconfigured Passive Object wizard. When the wizard opens click **Next** and **Finish** to create the service.

Once it's created click the **View status details for xxxx** link to see the new service.

There's nothing to report now as a trap will need to be received before the status updates. Go back to your Windows machine and start / stop the Print Spooler service. Within a brief time frame the service should be updated with a critical state:

You will also notice that if you go back to your Windows machine and start the Print Spooler service the service will stay in a critical state. This is because eTrap has not been configured to send a trap when a service has started (and a trap EVENT needs to be defined). The next section will show you how to do this.

Service Start Rule / SNMPTT EVENT



Create another eTrap rule like the following.

Click the **OK** button to create the rule and then click **File > Save and restart service** to make the rule active

Add the following EVENT in the `/etc/snmp/snmptt.conf` file:

```
EVENT event
.1.3.6.1.4.1.29037.8.9.0.1 "Status
Events" NORMAL
FORMAT Service started event: $-*
EXEC /usr/local/bin/snmptraphandling.py "$r" "SNMP Traps" "$s" "$@" "$-*"
"Service Started: $5"
MATCH $2: (svcstart)
```

Save the snmptt.conf file and restart the snmptt service:

```
service snmptt restart
```

Go back to your Windows machine and start the Print Spooler service, the SNMP Traps service will return to an OK state.

Host	Service	Status	Duration	Attempt	Last Check	Status Information
win7-02	SNMP Traps	Ok	16s	1/1	2017-06-01 14:59:22	Service Started: For Advanced license visit http://www.smartoservice.com/licenses - Service Control Manager:7036:The Print Spooler service entered the running state. / enterprises.29037.9.1 ():win7-02 enterprises.29037.9.2 ():svcstart enterprises.29037.9.

Now you have a service in Nagios XI that will receive Windows Event Logs via SNMP traps.

Further Reading

There eTrap program allows you to create more complicated rules, please refer to the official documentation on this page.

SNMPTT is extremely powerful, the examples in this documentation only scratch the surface of what it is capable. You may find the following documentation useful:

SNMP Trap Tutorial

Understanding Trap Variables

SNMPTT Documentation

Configuring Passive Services With Nagios XI

Finishing Up

This completes the documentation on how to monitor Windows Event Logs via SNMP within Nagios XI using the eTrap agent from the Nagios Exchange. If you have additional questions or other support-related questions, please visit us at our Nagios Support Forum, Nagios Knowledge Base, or Nagios Library:

Visit Nagios Support Forum Visit Nagios Knowledge Base Visit Nagios Library