

Configuring opEvents to send SNMP Traps

This page will describe the steps to configure opEvents to send SNMP traps as a working proof of concept, using the SNMPTRAP command line tool.

- [Pre-requisites](#)
- [Configuring opEvents to Send SNMP Traps](#)
 - [Configure the Script Action in Event Action](#)
 - [Before Configuring the Script Action](#)
 - [Adding the Script Action](#)
 - [After Configuring the Script Action](#)
 - [Configure an Action to Send SNMP Traps](#)
 - [Before Configuring the Policy](#)
 - [Adding the Policy](#)
 - [After Configuring the Policy](#)
 - [Test the Event Actions](#)
- [Testing opEvents Sending the TRAP](#)
 - [Installing the OPMANTEK-MIB](#)
 - [Generate an Event in opEvents](#)

Pre-requisites

A copy of the OPMANTEK-MIB.mib has been obtained, this is currently in DRAFT state and is available in /usr/local/nmis9/mibs/traps. You can also download it from [our GitHub](#).

opEvents has been installed and licensed.

opEvents is already processing events and "working".

NET-SNMP is installed on the target server and the snmptrap command should be /usr/bin/snmptrap.

Configuring opEvents to Send SNMP Traps

Configure the Script Action in Event Action

Before Configuring the Script Action

Where opEvents is installed, edit the file /usr/local/omk/conf/EventActions.json, locate the section called script. The default configuration file from /usr/local/omk/install includes this section:

```
{
  "script" : {
    "traceroute_node" : {
      "output" : "save",
      "arguments" : "--max-hops=20 node.configuration.host",
      "exec" : "traceroute"
    },
    "ping_node" : {
      "exec" : "/bin/ping",
      "arguments" : "-c 5 node.configuration.host",
      "output" : "save"
    },
    "ping_neighbor" : {
      "exec" : "/bin/ping",
      "output" : "save",
      "arguments" : "-c 5 event.element"
    }
  }
},
```

Alternatively you can also use the Event Actions Editor GUI which is available in opEvents by clicking on System then Edit Event Actions.

Adding the Script Action

We are going to be inserting the content below into that section.

In the output below IP_ADDRESS_OF_TEMIP needs to be replaced with the IP Address of the actual server and COMMUNITY_STRING with a string of your choice.

```

        "send_snmptrap" : {
            "arguments" : "-v 2c -Ci -c COMMUNITY_STRING IP_ADDRESS_OF_TEMPIP ' '
1.3.6.1.4.1.4818.1.1 1.3.6.1.4.1.4818.2.1.1 s event._id 1.3.6.1.4.1.4818.2.1.2 s event.time
1.3.6.1.4.1.4818.2.1.3 s event.date 1.3.6.1.4.1.4818.2.1.4 s event.node 1.3.6.1.4.1.4818.2.1.5 s event.host
1.3.6.1.4.1.4818.2.1.6 s event.event 1.3.6.1.4.1.4818.2.1.7 s event.element 1.3.6.1.4.1.4818.2.1.8 s event.
state 1.3.6.1.4.1.4818.2.1.9 s event.stateful 1.3.6.1.4.1.4818.2.1.10 s event.details 1.3.6.1.4.1.4818.2.1.11 s
event.type 1.3.6.1.4.1.4818.2.1.12 s event.priority 1.3.6.1.4.1.4818.2.1.13 s event.level",
            "exec" : "/usr/bin/snmptrap",
            "output" : "save"
        },
    },

```

After Configuring the Script Action

Once finished you will have a script section which looks like:

```

    "script" : {
        "send_snmptrap" : {
            "arguments" : "-v 2c -Ci -c COMMUNITY_STRING IP_ADDRESS_OF_TEMPIP ' ' 1.3.6.1.4.1.4818.1.1
1.3.6.1.4.1.4818.2.1.1 s event._id 1.3.6.1.4.1.4818.2.1.2 s event.time 1.3.6.1.4.1.4818.2.1.3 s event.date
1.3.6.1.4.1.4818.2.1.4 s event.node 1.3.6.1.4.1.4818.2.1.5 s event.host 1.3.6.1.4.1.4818.2.1.6 s event.event
1.3.6.1.4.1.4818.2.1.7 s event.element 1.3.6.1.4.1.4818.2.1.8 s event.state 1.3.6.1.4.1.4818.2.1.9 s event.
stateful 1.3.6.1.4.1.4818.2.1.10 s event.details 1.3.6.1.4.1.4818.2.1.11 s event.type 1.3.6.1.4.1.4818.2.1.12 s
event.priority 1.3.6.1.4.1.4818.2.1.13 s event.level",
            "exec" : "/usr/bin/snmptrap",
            "output" : "save"
        },
        "traceroute_node" : {
            "output" : "save",
            "arguments" : "--max-hops=20 node.configuration.host",
            "exec" : "traceroute"
        },
        "ping_node" : {
            "exec" : "/bin/ping",
            "arguments" : "-c 5 node.configuration.host",
            "output" : "save"
        },
        "ping_neighbor" : {
            "exec" : "/bin/ping",
            "output" : "save",
            "arguments" : "-c 5 event.element"
        }
    },

```

Configure an Action to Send SNMP Traps

Before Configuring the Policy

The default policy starts with a section 1, which is going to match any event.

```

"policy" : {
  "1" : {
    "BREAK" : "false",
    "IF" : "event.any",
    "THEN" : {
      "2" : {
        "THEN" : "tag.outageCurrent(FALSE)",
        "BREAK" : "false",
        "IF" : "event.details =~ \"outage_current=false\""
      },
      "20" : {
        "BREAK" : "false",
        "IF" : "node.configuration.roleType eq \"distribution\" and event.event =~ qr{Down}",
        "THEN" : "priority(+2)"
      },
      "10" : {
        "BREAK" : "false",
        "IF" : "node.configuration.roleType eq \"core\" and event.event =~ qr{Down}",
        "THEN" : "priority(+3)"
      },
      "1" : {
        "THEN" : "tag.outageCurrent(TRUE)",
        "IF" : "event.details =~ \"outage_current=true\"",
        "BREAK" : "false"
      }
    }
  }
},

```

Adding the Policy

The if statement here is going to be TRUE if the contents of the event name (event.event) do NOT contain OPMANTEK-MIB:

```

      "5" : {
        "IF" : "event.event ne \"OPMANTEK-MIB\"",
        "THEN" : "script.send_snmptrap()",
        "BREAK" : "false"
      },

```

After Configuring the Policy

The configuration will look like below.

```

"policy" : {
  "1" : {
    "IF" : "node.any and event.any",
    "THEN" : {
      "5" : {
        "IF" : "event.event ne \"OPMANTEK-MIB\"",
        "THEN" : "script.send_snmptrap()",
        "BREAK" : "false"
      },
      "10" : {
        "IF" : "node.configuration.roleType eq \"core\" and event.event =~ qr
{Down}",
        "THEN" : "priority(+3)",
        "BREAK" : "false"
      },
    }
  },

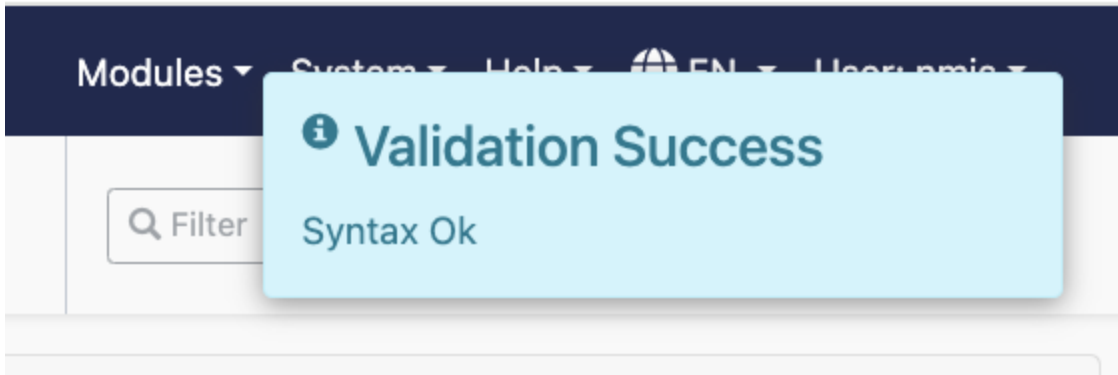
```

Test the Event Actions

When you have finished editing the file you can check the file in the Event Actions GUI in opEvents.

At the top of the screen there are three buttons, click on Validate.

If the Syntax is correct, you will see the following message display on screen:



Click on the Save button and restart the opEvents Daemon (opeventsd).

Testing opEvents Sending the TRAP

Installing the OPMANTEK-MIB

Copy the file OPMANTEK-MIB.mib to the target system which will be receiving the SNMP TRAPS. If this system is another vendors system, they will need to process the file into their system and confirm that it is done.

To load it into an FirstWave VM copy the file to /usr/local/nmis9/mibs/traps and restart the SNMP Trap daemon (snmptrapd). There may be other distribution-dependant items that may need to be configured, [for example on Ubuntu](#).

Generate an Event in opEvents

The easiest way to generate a test event to prove this is working is to use opevents-cli.pl.

You can invoke the following command which will create an event with some needed data:

```
/usr/local/omk/bin/opevents-cli.pl act=create-event node=YOUR_NODE event="Test Trap" state="open" stateful="Test Trap" details="This is a test trap event" element="SNMPTRAP" priority=7
```

This will create an stateful "open" event, so it is also best you close the event to prove statefulness is working end-to-end:

```
/usr/local/omk/bin/opevents-cli.pl act=create-event node=YOUR_NODE event="Test Trap Closed" state="closed" stateful="Test Trap" details="This is a test trap event" element="SNMPTRAP" priority=2
```