

3 SNMP traps

Overview

Receiving SNMP traps is the opposite to querying SNMP-enabled devices.

In this case the information is sent from a SNMP-enabled device and is collected or “trapped” by Zabbix.

Usually traps are sent upon some condition change and the agent connects to the server on port 162 (as opposed to port 161 on the agent side that is used for queries). Using traps may detect some short problems that occur amidst the query interval and may be missed by the query data.

Receiving SNMP traps in Zabbix is designed to work with **snmptrapd** and one of the built-in mechanisms for passing the traps to Zabbix - either a perl script or SNMPPTT.

The workflow of receiving a trap:

1. **snmptrapd** receives a trap
2. snmptrapd passes the trap to SNMPPTT or calls Perl trap receiver
3. SNMPPTT or Perl trap receiver parses, formats and writes the trap to a file
4. Zabbix SNMP trapper reads and parses the trap file
5. For each trap Zabbix finds all corresponding SNMP interfaces on hosts for the received IP or DNS address
6. For each found SNMP interface, the trap is compared to all regexes in “snmptrap[regex]” items. If found, the trap is set as the value of **all** matching items. If no match is found but there exists an “snmptrap.fallback” item, the trap is set as the value of that item.
7. If no match has been found for any of the corresponding SNMP interfaces, Zabbix by default logs the unmatched trap. (This is configured by “Log unmatched SNMP traps” in Administration → General → Other).

3.1 Configuring SNMP traps

Configuring the following fields in the frontend is specific for this item type:

- Your host must have an SNMP interface

In *Configuration* → *Hosts*, in the **Host interface** field set an SNMP interface with the correct IP or DNS address. The address from each received trap is compared to the IP and DNS addresses of all SNMP interfaces to find the corresponding hosts.

- Configure the item

In the **Key** field use one of the SNMP trap keys:

Key		
Description	Return value	Comments
snmptrap[regex]		

Key		
Description	Return value	Comments
Catches all SNMP traps from a corresponding address that match regex	SNMP trap	This item can be set only for SNMP interfaces. This item is supported starting from version 2.0.0. Note: Starting with Zabbix 2.0.5, user macros and global regular expressions are supported in the parameter of this item key.
snmptrap.fallback		
Catches all SNMP traps from a corresponding address that were not caught by any of the snmptrap[] items for that interface	SNMP trap	This item can be set only for SNMP interfaces. This item is supported starting from version 2.0.0.

Set the **Type of information** to be 'Log' for the timestamps to be parsed. Note that other formats such as 'Numeric' are also acceptable but might require a custom trap handler.

For SNMP trap monitoring to work, it must first be correctly set up.

3.2 Setting up SNMP trap monitoring

Configuring Zabbix server/proxy

To read the traps, Zabbix server or proxy must be configured to start the SNMP trapper process and point to the trap file that is being written by SNMPTT or a perl trap receiver. To do that, edit the configuration file ([zabbix_server.conf](#) or [zabbix_proxy.conf](#)):

1. StartSNMPTrapper=1
2. SNMPTrapperFile=[TRAP FILE]

Configuring SNMPTT

At first, snmptrapd should be configured to use SNMPTT.

For the best performance, SNMPTT should be configured as a daemon using **snmptthandler-embedded** to pass the traps to it. See instructions for configuring SNMPTT in its homepage: <http://snmptt.sourceforge.net/docs/snmptt.shtml>

When SNMPTT is configured to receive the traps, configure SNMPTT to log the traps:

1. log traps to the trap file which will be read by Zabbix:
 - log_enable = 1
 - log_file = [TRAP FILE]
2. set the date-time format:
 - date_time_format = %H:%M:%S %Y/%m/%d = [DATE TIME FORMAT]

Now format the traps for Zabbix to recognise them (edit `snmptt.conf`):

1. Each `FORMAT` statement should start with `"ZBXTRAP [address]"`, where `[address]` will be compared to IP and DNS addresses of SNMP interfaces on Zabbix. E.g.:
EVENT coldStart .1.3.6.1.6.3.1.1.5.1 "Status Events" Normal
FORMAT ZBXTRAP \$aA Device reinitialized (coldStart)
2. See more about SNMP trap format below.

Do not use unknown traps - Zabbix will not be able to recognise them. Unknown traps can be handled by defining a general event in `snmptt.conf`:

```
EVENT general .* "General event" Normal
```

Configuring Perl trap receiver

Requirements: Perl, Net-SNMP compiled with `--enable-embedded-perl` (done by default since Net-SNMP 5.4)

Perl trap receiver (look for `misc/snmptrap/zabbix_trap_receiver.pl`) can be used to pass traps to Zabbix server directly from `snmptrapd`. To configure it:

- add the perl script to `snmptrapd` configuration file (`snmptrapd.conf`), e.g.:
`perl do "[FULL PATH TO PERL RECEIVER SCRIPT]";`
- configure the receiver, e.g:
`$SNMPTrapperFile = '[TRAP FILE]';`
`$DateTimeFormat = '[DATE TIME FORMAT]';`

If script name is not quoted, `snmptrapd` will refuse to start up with messages, similar to these:

```
Regex modifiers "/l" and "/a" are mutually exclusive at (eval 2) line 1, at  
end of line  
Regex modifier "/l" may not appear twice at (eval 2) line 1, at end of line
```

SNMP trap format

All customised perl trap receivers and `SNMPTT` trap configuration must format the trap in the following way:

```
[timestamp] [the trap, part 1] ZBXTRAP [address] [the trap, part 2]
```

where

- `[timestamp]` - timestamp used for log items
- `ZBXTRAP` - header that indicates that a new trap starts in this line
- `[address]` - IP address used to find the host for this trap

Note that `"ZBXTRAP"` and `"[address]"` will be cut out from the message during processing. If the trap is formatted otherwise, Zabbix might parse the traps unexpectedly.

Example trap line in the file:

```
11:30:15 2011/07/27 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events" localhost - ZBXTRAP 192.168.1.1 Link down on interface 2. Admin state: 1. Operational state: 2
```

This will result in the following trap for SNMP interface with IP=192.168.1.1:

```
11:30:15 2011/07/27 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events" localhost - Link down on interface 2. Admin state: 1. Operational state: 2
```

3.3 System requirements

Log rotation

Zabbix does not provide any log rotation system - that should be handled by the user. The log rotation should first rename the old file and only later delete it so that no traps are lost:

1. Zabbix opens the trap file at the last known location and goes to step 3
2. Zabbix checks if the currently opened file has been rotated by comparing the inode number to the define trap file's inode number. If there is no opened file, Zabbix resets the last location and goes to step 1.
3. Zabbix reads the data from the currently opened file and sets the new location.
4. The new data are parsed. If this was the rotated file, the file is closed and goes back to step 2.
5. If there was no new data, Zabbix sleeps for 1 second and goes back to step 2.

File system

Because of the trap file implementation, Zabbix needs the file system to support inodes to differentiate files (the information is acquired by a `stat()` call).

3.4 Setup example

This example uses `snmptrapd` + `SNMPTT` to pass traps to Zabbix server. Setup:

1. **zabbix_server.conf** - configure Zabbix to start SNMP trapper and set the trap file:
StartSNMPTrapper=1
SNMPTrapperFile=/tmp/my_zabbix_traps.tmp
2. **snmptrapd.conf** - add `SNMPTT` as the trap handler:
traphandle default snmptt
3. **snmptt.ini** - configure output file and time format:
log_file = /tmp/my_zabbix_traps.tmp
date_time_format = %H:%M:%S %Y/%m/%d
4. **snmptt.conf** - define a default trap format:
EVENT general .* "General event" Normal
FORMAT ZBXTRAP \$aA \$ar
5. Create an SNMP item TEST:

```
Host's SNMP interface IP: 127.0.0.1
Key: snmptrap["General"]
Log time format: hh:mm:ss yyyy/MM/dd
```

This results in:

1. Command used to send a trap:
`snmptrap -v 1 -c public 127.0.0.1 '.1.3.6.1.6.3.1.1.5.3' '0.0.0.0' 6 33 '55' .1.3.6.1.6.3.1.1.5.3 s "teststring000"`
2. The received trap:
`15:48:18 2011/07/26 .1.3.6.1.6.3.1.1.5.3.0.33 Normal "General event" localhost - ZBXTRAP 127.0.0.1 127.0.0.1`
3. Value for item TEST:
`15:48:18 2011/07/26 .1.3.6.1.6.3.1.1.5.3.0.33 Normal "General event" localhost - 127.0.0.1`

This simple example uses SNMPPTT as **traphandle**. For better performance on production systems, use embedded Perl to pass traps from snmptrapd to SNMPPTT or directly to Zabbix.

Normal

FORMAT ZBXTRAP \$aA \$ar

1. Create an SNMP item TEST:
Host's SNMP interface IP: 127.0.0.1
Key: snmptrap[

From:
<https://www.zabbix.com/documentation/2.0/> - **Zabbix Documentation 2.0**

Permanent link:
<https://www.zabbix.com/documentation/2.0/manual/config/items/itemtypes/snmptrap?rev=1371209267>

Last update: **2014/09/25 13:38**

