

# 1 Configuring a trigger

## Overview

To configure a trigger, do the following:

- Go to: *Configuration* → *Hosts*
- Click on *Triggers* in the row of the host
- Click on *Create trigger* to the right (or on the trigger name to edit an existing trigger)
- Enter parameters of the trigger in the form

## Configuration

The **Trigger** tab contains all the essential trigger attributes.

Trigger
Tags
Dependencies

\* Name

Operational data

Severity Not classified Information Warning Average High Di

\* Expression  Add

Expression constructor

OK event generation Expression Recovery expression None

PROBLEM event generation mode Single Multiple

OK event closes All problems All problems if tag values match

\* Tag for matching

Allow manual close

URL

Description

Enabled

Add
Cancel

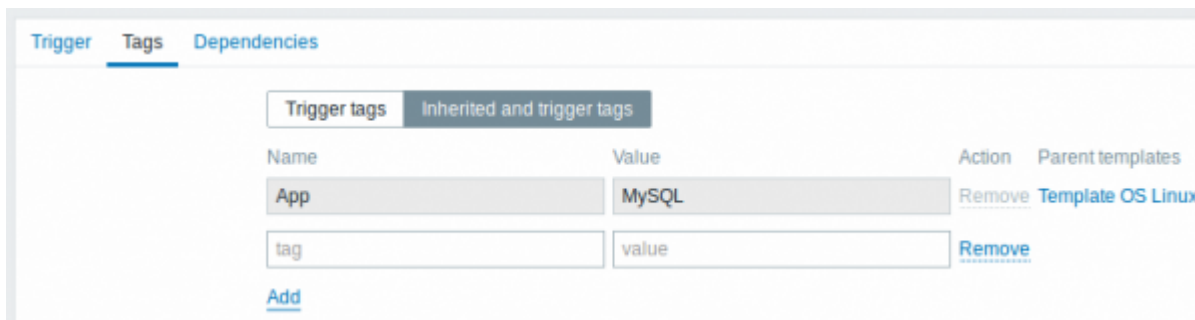
All mandatory input fields are marked with a red asterisk.

Parameter	Description
<i>Name</i>	Trigger name. Supported <a href="#">macros</a> are: {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {ITEM.VALUE}, {ITEM.LASTVALUE} and {\${MACRO}} user macros. <b>\$1, \$2...\$9</b> macros can be used to refer to the first, second...ninth constant of the expression. <i>Note:</i> \$1-\$9 macros will resolve correctly if referring to constants in relatively simple, straightforward expressions. For example, the name "Processor load above \$1 on {HOST.NAME}" will automatically change to "Processor load above 5 on New host" if the expression is {New host:system.cpu.load[percpu,avg1].last()}>5

Parameter	Description
<i>Operational data</i>	Operational data allow to define arbitrary strings along with macros. The macros will resolve dynamically to real time data in <i>Monitoring</i> → <i>Problems</i> . While macros in the trigger name (see above) will resolve to their values at the moment of a problem happening and will become the basis of a static problem name, the macros in the operational data maintain the ability to display the very latest information dynamically. Supported <b>macros</b> are: {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT}, {ITEM.VALUE}, {ITEM.LASTVALUE} and {\$MACRO} user macros. This field is supported since Zabbix 4.4.
<i>Severity</i>	Set the required trigger <b>severity</b> by clicking the buttons.
<i>Expression</i>	Logical <b>expression</b> used to define the conditions of a problem. A problem is created after all the conditions included in the expression are met, i.e. the expression evaluates to TRUE. The problem will be resolved as soon as the expression evaluates to FALSE, unless additional recovery conditions are specified in <i>Recovery expression</i> .
<i>OK event generation</i>	OK event generation options: <b>Expression</b> - OK events are generated based on the same expression as problem events; <b>Recovery expression</b> - OK events are generated if the problem expression evaluates to FALSE and the recovery expression evaluates to TRUE; <b>None</b> - in this case the trigger will never return to an OK state on its own. Supported since Zabbix 3.2.0.
<i>Recovery expression</i>	Logical <b>expression</b> (optional) defining additional conditions that have to be met before the problem is resolved, after the original problem expression has already been evaluated as FALSE. Recovery expression is useful for trigger <b>hysteresis</b> . It is <b>not</b> possible to resolve a problem by recovery expression alone if the problem expression is still TRUE. This field is only available if 'Recovery expression' is selected for <i>OK event generation</i> . Supported since Zabbix 3.2.0.
<i>PROBLEM event generation mode</i>	Mode for generating problem events: <b>Single</b> - a single event is generated when a trigger goes into the 'Problem' state for the first time; <b>Multiple</b> - an event is generated upon every 'Problem' evaluation of the trigger.
<i>OK event closes</i>	Select if OK event closes: <b>All problems</b> - all problems of this trigger <b>All problems if tag values match</b> - only those trigger problems with matching event tag values Supported since Zabbix 3.2.0.
<i>Tag for matching</i>	Enter event tag name to use for event correlation. This field is displayed if 'All problems if tag values match' is selected for the <i>OK event closes</i> property and is mandatory in this case. Supported since Zabbix 3.2.0.
<i>Allow manual close</i>	Check to allow <b>manual closing</b> of problem events generated by this trigger. Manual closing is possible when acknowledging problem events. Supported since Zabbix 3.2.0.
<i>URL</i>	If not empty, the URL entered here is available as a link in several frontend locations, e.g. when clicking on the problem name in <i>Monitoring</i> → <i>Problems</i> (URL option in the <i>Trigger</i> menu) and <i>Problems</i> dashboard widget. Supported macros: {EVENT.ID}, {ITEM.VALUE}, {ITEM.LASTVALUE}, {TRIGGER.ID}, several {HOST.*} macros, user macros (secret macros in URLs will not be resolved).

Parameter	Description
<i>Description</i>	Text field used to provide more information about this trigger. May contain instructions for fixing specific problem, contact detail of responsible staff, etc. <i>Starting with Zabbix 2.2</i> , the description may contain the same set of macros as trigger name.
<i>Enabled</i>	Unchecking this box will disable the trigger if required.

The **Tags** tab allows you to define trigger-level tags. All problems of this trigger will be tagged with the values entered here.



In addition the *Inherited and trigger tags* option allows to view tags defined on template level, if the trigger comes from that template. If there are multiple templates with the same tag, these tags are displayed once and template names are separated with commas. A trigger does not "inherit" and display host-level tags.

Parameter	Description
<i>Name/Value</i>	<p>Set custom tags to mark trigger events.</p> <p>Tags are a pair of tag name and value. You can use only the name or pair it with a value. A trigger may have several tags with the same name, but different values.</p> <p>User macros, user macro context, low-level discovery macros and macro <a href="#">functions</a> with <code>{{ITEM.VALUE}}</code>, <code>{{ITEM.LASTVALUE}}</code> and low-level discovery macros are supported in event tags. Low-level discovery macros can be used inside macro context. <code>{TRIGGER.ID}</code> macro is supported in trigger tag values. It may be useful for identifying triggers created from trigger prototypes and, for example, suppressing problems from these triggers during maintenance.</p> <p>If the total length of expanded value exceeds 255, it will be cut to 255 characters.</p> <p>See all <a href="#">macros</a> supported for event tags.</p> <p><a href="#">Event tags</a> can be used for event correlation, in action conditions and will also be seen in <i>Monitoring</i> → <i>Problems</i> or the <i>Problems</i> widget.</p> <p>Supported since Zabbix 3.2.0.</p>

The **Dependencies** tab contains all the [dependencies](#) of the trigger.

Click on *Add* to add a new dependency.

You can also configure a trigger by opening an existing one, pressing the *Clone* button and then saving under a different name.

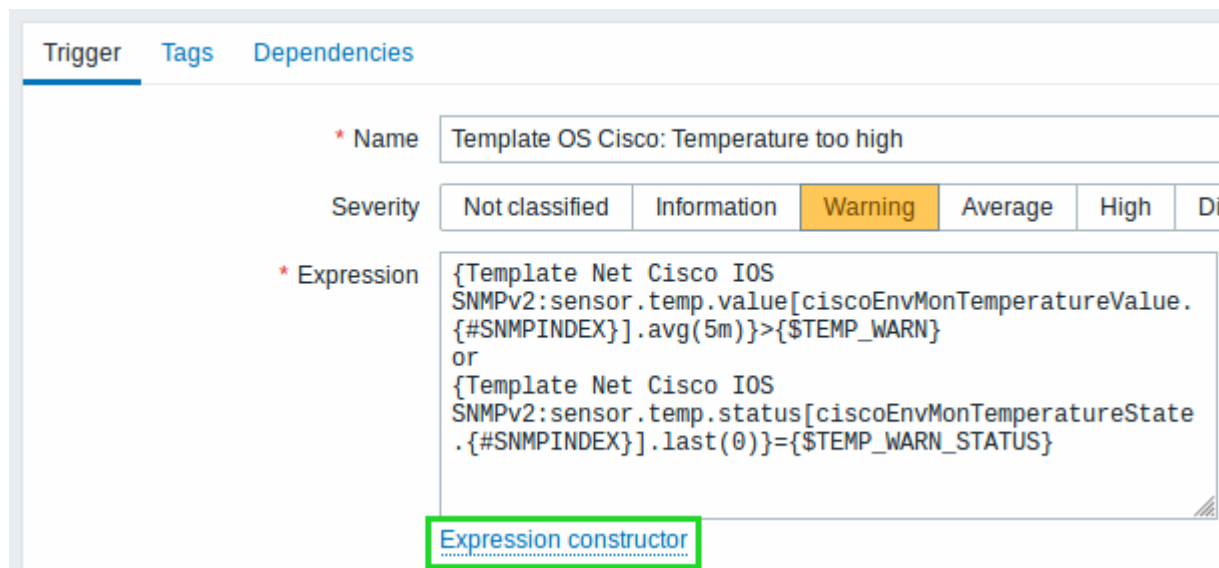
### Testing expressions

It is possible to test the configured trigger expression as to what the expression result would be depending on the received value.

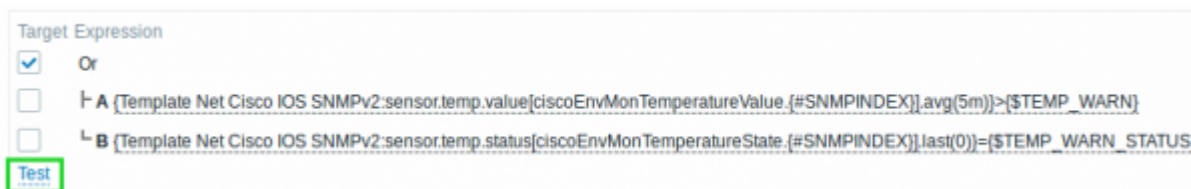
Following expression from an official template is taken as an example:

```
{Template Net Cisco IOS
SNMPv2:sensor.temp.value[ciscoEnvMonTemperatureValue.{#SNMPINDEX}].avg(5m)}>
{$TEMP_WARN}
or
{Template Net Cisco IOS
SNMPv2:sensor.temp.status[ciscoEnvMonTemperatureState.{#SNMPINDEX}].last(0)}
={$TEMP_WARN_STATUS}
```

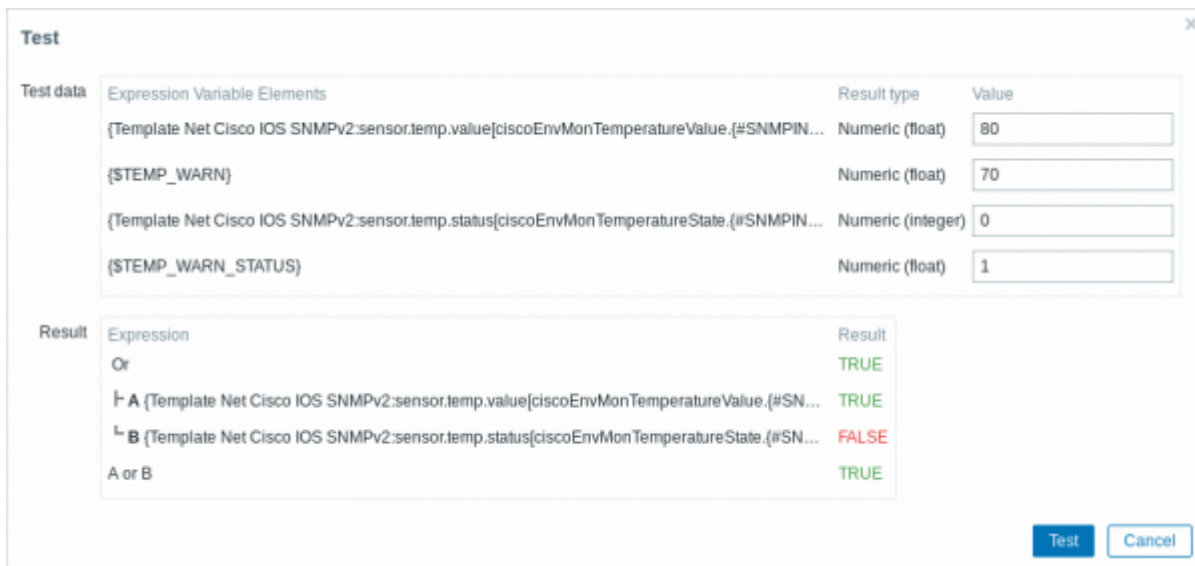
To test the expression, click on *Expression constructor* under the expression field.



In the Expression constructor, all individual expressions are listed. To open the testing window, click on *Test* below the expression list.



In the testing window you can enter sample values ("80, 70, 0, 1" in this example) and then see the expression result, by clicking on the *Test* button.



The result of the individual expressions as well as the whole expression can be seen.

“TRUE” result means the specified expression is correct. In this particular case A, “80” is greater than {\$TEMP\_WARN} specified value, “70” in this example. Respectively, “TRUE” result appears.

“FALSE” result means the specified expression is incorrect. In this particular case B, {\$TEMP\_WARN\_STATUS}, “1” in this example, needs to be equal with specified “0” value, that is wrong. Respectively, “FALSE” result appears.

Chosen Expression type is “OR”/“TRUE”. If at least one of the specified conditions (A or B in this case) is TRUE, overall result will be TRUE as well. That means, current value exceeds the warning value and a Problem has occurred.

From: <https://www.zabbix.com/documentation/5.2/> - **Zabbix Documentation 5.2**

Permanent link: <https://www.zabbix.com/documentation/5.2/manual/config/triggers/trigger>

Last update: **2020/03/13 09:59**

