

1 User macros

Overview

For greater flexibility, Zabbix supports user macros, which can be defined on global, template and host level. These macros have a special syntax: **{*\$MACRO*}**.

The macros can be used in:

- item names
- item key parameters
- trigger names
- trigger expression parameters and constants (see examples)
- several other [locations](#)

The following characters are allowed in the macro names: **A-Z** , **0-9** , **_** , **.**

Zabbix substitutes macros according to the following precedence:

1. host level macros (checked first)
2. macros defined for first level templates of the host (i.e., templates linked directly to the host), sorted by template ID
3. macros defined for second level templates of the host, sorted by template ID
4. macros defined for third level templates of the host, sorted by template ID
5. ...
6. global macros (checked last)

In other words, if a macro does not exist for a host, Zabbix will try to find it in the host templates of increasing depth. If still not found, a global macro will be used, if exists.

If Zabbix is unable to find a macro, the macro will not be substituted.

To define user macros, go to the corresponding locations in the frontend:

- for global macros, visit *Administration* → *General* → *Macros*
- for host and template level macros, open host or template properties and look for the *Macros* tab

If a user macro is used in items or triggers in a template, it is suggested to add that macro to the template even if it is defined on a global level. That way, exporting the template to XML and importing it in another system will still allow it to work as expected.

Most common use cases of global and host macros:

1. taking advantage of templates with host specific attributes: passwords, port numbers, file names, regular expressions, etc
2. global macros for global one-click configuration changes and fine tuning

Examples

Example 1

Use of host-level macro in the “Status of SSH daemon” item key:

```
net.tcp.service[ssh,,{SSH_PORT}]
```

This item can be assigned to multiple hosts, providing that the value of **{SSH_PORT}** is defined on those hosts.

Example 2

Use of host-level macro in the “CPU load is too high” trigger:

```
{ca_001:system.cpu.load[,avg1].last(0)}>{$MAX_CPULOAD}
```

Such a trigger would be created on the template, not edited in individual hosts.

If you want to use amount of values as the function parameter (for example, **max(#3)**), include hash mark in the macro definition like this: **SOME_PERIOD ⇒ #3**

Example 3

Use of two macros in the “CPU load is too high” trigger:

```
{ca_001:system.cpu.load[,avg1].min({CPULOAD_PERIOD})}>{$MAX_CPULOAD}
```

Note that a macro can be used as a parameter of trigger function, in this example function **min()**.

In trigger expressions user macros will expand if referencing a parameter or constant. They will NOT expand if referencing the host, item key, function, operator or another trigger expression.

From:

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

Permanent link:

<https://www.zabbix.com/documentation/2.0/manual/config/macros/usermacros>

Last update: **2015/05/05 07:56**

