

## 21.4 User parameters

### Overview

Sometimes you may want to run an agent check that does not come predefined with Zabbix. This is where user parameters come to help.

You may write a command that retrieves the data you need and include it in the user parameter in the [agent configuration file](#) ('UserParameter' configuration parameter).

A user parameter has the following syntax:

```
UserParameter=<key>,<command>
```

As you can see, a user parameter also contains a key. The key will be necessary when configuring an item. Enter a key of your choice that will be easy to reference (it must be unique within a host). Restart the agent.

Then, when [configuring an item](#), enter the key to reference the command from the user parameter you want executed.

User parameters are commands executed by Zabbix agent. Up to 512KB of data can be returned. Note, however, that the text value that can be eventually stored in database is limited to 64KB on MySQL (see info on other databases in the [table](#)).

The return value of the command is standard output; standard error is discarded. **/bin/sh** is used as a command line interpreter under UNIX operating systems. User parameters obey the agent check timeout; if timeout is reached the forked user parameter process is terminated.

See also:

- [Step-by-step tutorial](#) on making use of user parameters
- [Command execution](#)

### Examples of simple user parameters

A simple command:

```
UserParameter=ping,echo 1
```

The agent will always return '1' for an item with 'ping' key.

A more complex example:

```
UserParameter=mysql.ping,mysqladmin -uroot ping|grep -c alive
```

The agent will return '1', if MySQL server is alive, '0' - otherwise.

## Flexible user parameters

Flexible user parameters accept parameters with the key. This way a flexible user parameter can be the basis for creating several items.

Flexible user parameters have the following syntax:

```
UserParameter=key[*], command
```

Parameter	Description
<b>Key</b>	Unique item key. The [*] defines that this key accepts parameters within the brackets. Parameters are given when configuring the item.
<b>Command</b>	Command to be executed to evaluate value of the key. <i>For flexible user parameters only:</i> You may use positional references \$1...\$9 in the command to refer to the respective parameter in the item key. Zabbix parses the parameters enclosed in [ ] of the item key and substitutes \$1,...,\$9 in the command accordingly. \$0 will be substituted by the original command (prior to expansion of \$0,...,\$9) to be run. Positional references with the \$ sign are interpreted regardless of whether they are enclosed between double (") or single (') quotes. To use positional references unaltered, specify a double dollar sign - for example, awk '{print \$\$2}'. In this case \$\$2 will actually turn into \$2 when executing the command.

Positional references with the \$ sign are searched for and replaced by Zabbix agent only for flexible user parameters. For simple user parameters, such reference processing is skipped and, therefore, any \$ sign quoting is not necessary.

Unless [UnsafeUserParameters](#) agent daemon configuration option is enabled, it is not allowed to pass flexible parameters containing these symbols: \ ' " ` \* ? [ ] { } ~ \$ ! & ; ( ) < > | # @. Additionally, newline is not allowed either.

User parameters that return text (character, log, text types of information) now can return whitespace only as well, setting the return value to an empty string (supported since 2.0). If non-valid value is returned, ZBX\_NOTSUPPORTED will be sent back by the agent.

### Example 1

Something very simple:

```
UserParameter=ping[*],echo $1
```

We may define unlimited number of items for monitoring all having format ping[something].

- ping[0] - will always return '0'
- ping[aaa] - will always return 'aaa'

### Example 2

Let's add more sense!

```
UserParameter=mysql.ping[*],mysqladmin -u$1 -p$2 ping | grep -c alive
```

This parameter can be used for monitoring availability of MySQL database. We can pass user name and password:

```
mysql.ping[zabbix,our_password]
```

### Example 3

How many lines matching a regular expression in a file?

```
UserParameter=wc[*],grep -c "$2" $1
```

This parameter can be used to calculate number of lines in a file.

```
wc[/etc/passwd,root]  
wc[/etc/services,zabbix]
```

From:

<https://www.zabbix.com/documentation/2.2/> - **Zabbix Documentation 2.2**

Permanent link:

<https://www.zabbix.com/documentation/2.2/manual/config/items/userparameters>

Last update: **2018/01/25 13:00**

