

4 Пользовательские параметры

Обзор

Иногда вы можете захотеть выполнять проверку через агент, которая не предопределена в Zabbix. Здесь на помощь придут пользовательские параметры.

Вы можете написать команду, которая вернет необходимые вам данные и добавить эту команду как пользовательский параметр в [файл конфигурации агента](#) (параметр конфигурации 'UserParameter').

Пользовательский параметр имеет следующий синтаксис:

```
UserParameter=<ключ>,<команда>
```

Как вы можете видеть, пользовательский параметр также содержит ключ. Ключ будет необходим при настройке элемента данных. Укажите ключ на ваш выбор, на который будет легко ссылаться (он должен быть уникальным в пределах узла сети). Перезапустите агента.

Затем, когда [настроите элемент данных](#), укажите ключ на который ссылается команда из пользовательского параметра, которую вы хотите выполнять.

Команды из пользовательских параметров выполняются Zabbix агентом. Может быть возвращено до 512КБ (64КБ до Zabbix 2.0.5). В операционных системах UNIX используется интерпретатор командной строки **/bin/sh**. Таким образом, вы можете повысить функциональность Zabbix агентов.

Смотрите [пошаговую инструкцию](#) по использованию пользовательских параметров.

Примеры простых пользовательских параметров

Простая команда:

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

Агент будет всегда возвращать '1' для элемента данных с ключем 'ping'.

Более сложный пример:

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

Агент будет возвращать '1', если MySQL сервер доступен, '0' - в противном случае.

Гибкие пользовательские параметры

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=ключ[*],команда
```

Параметр	Описание
Ключ	Уникальный ключ элемента данных. [*] задает, что ключ может принимать параметры из скобок.
Команда	Команда, которая будет выполнена для получения значения ключа. Zabbix агент производит анализ содержимого внутри квадратных скобок [] и заменяет им переменные \$1,...,\$9 в команде. Переменная \$0 будет заменена оригинальной командой (до расширения \$0,...,\$9) для выполнения.

Для использования ссылок на позиции не измененными, укажите двойной символ доллара - например, `awk '{print $$2}'`.

Если опция [UnsafeUserParameters](#) не определена в конфигурационном файле Zabbix агента, то в командах недопустимы следующие символы: \ ' " ` * ? [] { } ~ \$! & ; () < > | # @. Так же, не допускается использование новой строки.

Все элементы данных которые возвращают текст (символ, журнал, текстовые типы информации) теперь могут возвращать только пробелы (при необходимости), при этом возвращенное значение становится пустой строкой (поддерживается начиная с 2.0). Если командой возвращено несоответствующее требованиям значение, тогда агентом будет отправлено сообщение ZBX_NOTSUPPORTED.

Пример 1

Очень простой пример:

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

Мы можем создать неограниченное количество элементов данных для наблюдения за чем угодно, указав в таком формате `ping[что угодно]`.

- `ping[0]` - будет всегда возвращать '0'
- `ping[aaa]` - будет всегда возвращать 'aaa'

Пример 2

Давайте добавим больше смысла!

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

Этот параметр может быть использован для мониторинга доступности баз данных MySQL. В качестве параметров мы можем передать имя пользователя и пароль. Ключ элемента данных в таком случае выглядит вот так:

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

Пример 3

Сколько строк в файле, которые соответствуют регулярному выражению?

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

Этот параметр может быть использован для подсчета количества строк в файле.

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

From:

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

Permanent link:

<https://www.zabbix.com/documentation/2.0/ru/manual/config/items/userparameters>

Last update: **2014/12/30 17:23**

