

## 4 Пользовательские параметры

### Обзор

Иногда вы можете захотеть выполнять проверку через агент, которая не предопределена в Zabbix. В этом случае вам 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 потребуется при настройке элемента данных. Введите ключ на ваш выбор, на который будет легко ссылаться (он должен быть уникальным в пределах узла сети). Перезапустите агента.

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

Команды из пользовательских параметров выполняются Zabbix агентом. Может быть возвращено до 512КБ данных до шагов предварительной обработки элементов данных. Однако, обратите внимание на то, что такое текстовое значение, которое может быть сохранено в базе данных ограничено 64КБ в MySQL (смотрите информацию по остальным базам данных в [таблице](#)).

В операционных системах UNIX используется интерпретатор командной строки **/bin/sh**. Пользовательские параметры подчиняются времени ожидания проверок на агента; если время ожидания будет превышено, созданный по пользовательским параметрам процесс будет завершен.

Смотрите также:

- [Пошаговую инструкцию](#) по использованию пользовательских параметров.
- [Выполнение команд](#)

### 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

Гибкие пользовательские параметры допускают параметры с указанным ключем. В этом случае гибкие пользовательские параметры могут быть основой для создания нескольких элементов данных.

Гибкие пользовательские параметры имеют следующий синтаксис:

```
UserParameter=ключ[*],команда
```

Параметр	Описание
<b>Ключ</b>	Уникальный ключ элемента данных. [*] задает, что ключ может принимать параметры из скобок. Параметры указываются при настройке элемента данных.
<b>Команда</b>	Команда, которая выполняется для получения значения ключа. <i>Только для гибких пользовательских параметров:</i> Вы можете использовать ссылки на позиции \$1...\$9 для того, чтобы обратиться к соответствующему параметру в ключе элемента данных. Zabbix разбирает параметры заключенные в [ ] ключа элемента данных и заменяет \$1,...,\$9 в команде соответственно. \$0 будет заменена оригинальной командой (до раскрытия \$0,...,\$9) для выполнения. Ссылки на позиции интерпретируются Zabbix агентом независимо от того заключены ли они в двойные (") или в одинарные (') кавычки. Для использования ссылок на позиции без изменения, укажите двойной символ доллара - например, awk '{print \$\$2}'. В этом случае \$\$2 фактически превратится в \$2 при выполнении команды.

Ссылки на позиции, начинающиеся со знака \$, ищутся и заменяются Zabbix агентом только в случае гибких пользовательских параметров. В случае простых пользовательских параметров такие ссылки пропускаются и, следовательно, экранирование любых знаков \$ не требуется. По умолчанию некоторые символы не разрешены в пользовательских параметрах. Смотрите документацию по [UnsafeUserParameters](#) для получения полного списка таких символов.

### 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 можно использовать для мониторинга доступности баз данных MySQL. В качестве параметров мы можем передать имя пользователя и пароль:

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

## Пример 3

Сколько строк в файле, которые соответствуют регулярному выражению?

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

Этот параметр можно использовать для подсчета количества строк в файле.

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

## Результат команды

Результирующим значением команды является стандартный вывод вместе со стандартным выводом ошибок.

Текстовый элемент данных (символ, журнал или текст типами информации) не станет неподдерживаемым в случае вывода ошибки стандартным выводом.

Пользовательские параметры, которые возвращают текст (символ, журнал, текстовый типы информации) могут возвращать пробел. В случае ошибочного результата элемент данных станет неподдерживаемым.

From:

<https://www.zabbix.com/documentation/current/> - **Zabbix Documentation 4.4**

Permanent link:

<https://www.zabbix.com/documentation/current/ru/manual/config/items/userparameters?rev=1531395944>

Last update: **2019/04/02 17:21**

