

4 Parâmetros de usuário

Visão geral

Algumas vezes você pode precisar executar uma verificação que não faça parte do kit de testes nativos do Zabbix. Nestes casos os parâmetros de usuário poderão estender as funcionalidades da solução para novos testes desenvolvidos por você.

Você pode desenvolver um comando que extraia os dados que você precisa e incluí-lo como um 'User Parameter' no [arquivo de configuração do agente](#).

Um parâmetro de usuário tem a sintaxe a seguir:

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

Como você deve ter percebido, um 'User Parameter' também contém uma chave. Esta chave é o identificador único da nova funcionalidade e será necessária quando você for configurar o item. Você pode definir o nome que preferir. Uma vez que tenha terminado os ajustes no arquivo de configurações do Zabbix Agent, será necessário o reinício do processo do agente para que as novas configurações estejam acessíveis.

Logo, quando configurar o [item](#), informe o nome de chave definido no 'User Parameter' para que o Zabbix saiba o que coletar.

Os parâmetros de usuário são executados pelo Zabbix Agent. O limite de tamanho do dado retornado é de 512KB. `/bin/sh` será utilizado como interpretador de comando em ambientes UNIX. Os parâmetros de usuário respeitam o parâmetro de 'timeout' do agente; se o timeout for alcançado o processo iniciado pelo 'User Parameter' será terminado.

Consulte também: * [tutorial passo-a-passo](#) sobre como usar os parâmetros de usuário. * [Execução de comandos](#)

Exemplos de parâmetros de usuário simples

Um simples comando:

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

O agente sempre irá retornar o valor '1' para um item com a chave 'ping'.

Um comando mais complexo:

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

O agente irá retornar '1', se o MySQL server estiver em execução ou '0' caso não esteja.

Parâmetros de usuário flexíveis

Os parâmetros de usuário flexíveis aceitam parâmetros em suas chaves. Desta forma um mesmo 'User Parameter' poderá ser chamado por vários itens no mesmo host, bastando para isso que os seus parâmetros sejam diferentes.

Os parâmetros de usuário flexíveis tem a seguinte sintaxe:

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

Parâmetro	Descrição
Key	Chave única do item. O “[*]” define que este parâmetro de usuário aceitará parâmetros entre colchetes. Os parâmetros serão informados ao configurar os itens.
Command	Comando a ser executado quando a chave for necessária. <i>Para parâmetros flexíveis apenas:</i> Você pode utilizar referências posicionais \$1...\$9 no comando para utilizar os parâmetros da chave. O Zabbix analisa os parâmetros entre “[]” da chave do item e os substitui no formato do shell (\$1,...,\$9). Referências posicionais são interpretadas independente de estarem escapadas com aspas duplas ou simples. Para utilizar referências posicionais sem alterações, defina o caractere de dolar (\$) de forma duplicada - por exemplo, awk '{print \$\$2}'. Neste caso, o \$\$2 será alterado para \$2 no momento da execução do comando.

As referências posicionais com o caractere \$ são pesquisadas e substituídas pelo Zabbix Agent somente para parâmetros de usuário que são flexíveis. Para parâmetros de usuário simples, esta referência será ignorada e o sinal de \$ não será necessário.

Alguns símbolos não são permitidos nos parâmetros de usuário por padrão. Veja o manual de [parâmetros inseguros](#) para uma lista completa.

Exemplo 1

Algo muito simples:

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

Podemos definir um número ilimitado de itens para monitorar com o formato 'ping[something]'.

- ping[0] - sempre retornará '0'
- ping[aaa] - sempre retornara 'aaa'

Exemplo 2

Vamos fazer algo mais útil!

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

Este parâmetro pode ser utilizado para verificar a disponibilidade de um BD MySQL. Nós podemos informar ainda um usuário e senha:

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

Exemplo 3

Quantas linhas correspondem a determinada expressão regular em um arquivo?

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

Este parâmetro pode ser utilizado para calcular a quantidade de linhas em um arquivo.

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

Resultado do comando

O valor de retorno do comando será o output padrão junto com as mensagens de erro.

Um item de texto (tipo de dado de caractere, log ou texto) ficará como 'não suportado' caso de retorno de erro no output padrão.

Parâmetros de usuário que retornam textos poderão retornar caracteres em branco. No caso de um resultado inválido o item passa a ser não suportado.

From:

<https://www.zabbix.com/documentation/3.0/> - **Zabbix Documentation 3.0**

Permanent link:

<https://www.zabbix.com/documentation/3.0/pt/manual/config/items/userparameters>

Last update: **2016/02/17 13:27**

