

4 Paramètres utilisateur

Aperçu

Parfois, vous pouvez vouloir exécuter une vérification d'agent qui n'est pas prédéfinie dans Zabbix. C'est là que les paramètres utilisateur viennent aider.

Vous pouvez écrire une commande qui récupère les données dont vous avez besoin et les inclure dans le paramètre utilisateur dans le [fichier de configuration de l'agent](#) (paramètre de configuration 'UserParameter').

Un paramètre utilisateur a la syntaxe suivante :

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

Comme vous pouvez le voir, un paramètre utilisateur contient également une clé. La clé sera nécessaire lors de la configuration d'un élément. Entrez une clé de votre choix qui sera facile à référencer (elle doit être unique pour un hôte). Redémarrez l'agent.

Ensuite, lors de la [configuration d'un élément](#), entrez la clé pour référencer la commande à partir du paramètre utilisateur que vous voulez exécuter.

Les paramètres utilisateur sont des commandes exécutées par l'agent Zabbix. Jusqu'à 512 Ko de données peuvent être renvoyées avant les étapes de pré-traitement des éléments. Notez, cependant, que la valeur de texte qui peut éventuellement être stockée dans la base de données est limitée à 64 Ko sur MySQL (voir les informations sur les autres bases de données dans le [tableau](#)).

/bin/sh est utilisé comme interpréteur de ligne de commande sur les systèmes d'exploitation UNIX. Les paramètres utilisateur obéissent à la temporisation de vérification de l'agent ; Si le timeout est atteint, le processus forké du paramètre utilisateur est terminé.

Voir aussi :

- [Tutoriel pas à pas](#) sur l'utilisation des paramètres utilisateur
- [Exécution de commandes](#)

Exemples de paramètres utilisateur simple

Une commande simple :

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

L'agent renverra toujours '1' pour une élément avec la clé 'ping'.

Un exemple plus complexe :

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

L'agent renverra '1', si le serveur MySQL est actif, '0' - sinon.

Paramètres utilisateurs flexibles

Les paramètres utilisateur flexibles acceptent les paramètres avec la clé. De cette façon, un paramètre utilisateur flexible peut être la base pour créer plusieurs éléments.

Les paramètres utilisateur flexibles ont la syntaxe suivante :

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

Paramètre	Description
Clé	Clé d'élément unique. [*] définit que cette clé accepte les paramètres entre parenthèses. Les paramètres sont donnés lors de la configuration de l'élément.
Commande	Commande à exécuter pour évaluer la valeur de la clé. <i>Pour les paramètres flexibles uniquement :</i> Vous pouvez utiliser les références positionnelles \$1...\$9 dans la commande pour faire référence au paramètre respectif dans la clé d'élément. Zabbix analyse les paramètres inclus dans les [] de la clé d'élément et substitue \$1,...,\$9 dans la commande en conséquence. \$0 sera remplacé par la commande d'origine (avant l'expansion de \$0,...,\$9) à exécuter. Les références positionnelles sont interprétées indépendamment du fait qu'elles soient placées entre des guillemets doubles (") ou simple ('). Pour utiliser des références positionnelles non modifiées, spécifiez un signe dollar (\$) - par exemple, awk '{print \$\$2}'. Dans ce cas, \$\$2 se transforme en '\$2' lors de l'exécution de la commande.

Les références positionnelles avec le signe \$ sont recherchées et remplacées par l'agent Zabbix uniquement pour les paramètres utilisateur flexibles. Pour les paramètres utilisateur simples, ce traitement de référence est ignoré et, par conséquent, toute citation de signe \$ n'est pas nécessaire. Certains symboles ne sont pas autorisés dans les paramètres utilisateur par défaut. Voir la documentation des [UnsafeUserParameters](#) pour une liste complète.

Exemple 1

Quelque chose de très simple :

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

Nous pouvons définir un nombre illimité d'éléments pour la surveillance de tous les formats ping[quelque chose].

- ping[0] - retournera toujours '0'
- ping[aaa] - retournera toujours 'aaa'

Exemple 2

Ajoutons un peu de sens !

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

Ce paramètre peut être utilisé pour superviser la disponibilité de la base de données MySQL. Nous pouvons passer le nom d'utilisateur et le mot de passe :

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

Exemple 3

Combien de lignes correspondent à une expression régulière dans un fichier ?

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

Ce paramètre peut être utilisé pour calculer le nombre de ligne dans un fichier.

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

Résultat de la commande

La valeur de retour de la commande est la sortie standard avec l'erreur standard.

Un élément de texte (caractère, journal ou information de type texte) ne deviendra pas non supporté en cas de sortie d'erreur standard.

Les paramètres utilisateurs qui renvoient du texte (caractère, journal ou information de type texte) peuvent renvoyer des espaces. En cas de retour incorrect, l'élément deviendra non supporté.

From:

<https://www.zabbix.com/documentation/4.0/> - **Zabbix Documentation 4.0**

Permanent link:

<https://www.zabbix.com/documentation/4.0/fr/manual/config/items/userparameters>

Last update: **2019/03/04 13:21**

