

FR

A quoi servent les macros ?

L'une des caractéristiques qui rend Shinken Enterprise flexible, c'est sa capacité à utiliser des données dans la définition des [Commandes](#). Ces données permettent de référencer des informations provenant des hôtes, des services, ou d'autres sources dans les commandes.

Remplacement de données

Avant d'exécuter une commande, Shinken Enterprise va remplacer toutes les données trouvées dans la définition de la commande avec leur valeurs correspondantes. Ce remplacement s'opère pour toute commande que Shinken Enterprise exécute : vérification d'hôte et de check, notification, exécution d'événements, etc .

Ce remplacement est récursif. Si une macro contient à son tour une macro, cette seconde macro sera résolue. Ce processus continue jusqu'à ce que la commande ne contienne plus de macro.



L'utilisation littérale du caractère '\$' nécessitera l'utilisation de '\$\$'. C'est également le cas dans les règles de Clusters, qui peuvent aussi contenir des macros.

Exemple 1 : Adresse générique

Lorsque vous utilisez un hôte ou un service dans les données de définitions de commande , ils se réfèrent à des valeurs pour l'hôte ou le service pour lequel la commande est en cours d'exécution

Prenons un exemple. Imaginons que nous utilisons une définition de l'hôte "Linuxbox" qui est vérifié par une commande check_ping, qui est définie comme suit.

```
/var/lib/Shinken Enterprise/libexec/check_ping -H $HOSTADDRESS$ -w 100.0,90% -c 200.0,60%
```

Linuxbox a pour adresse 192.168.1.2.

La macros sera remplacée et la ligne de commande finale suivante sera exécutée :

```
/var/lib/Shinken Enterprise/libexec/check_ping -H 192.168.1.2 -w 100.0,90% -c 200.0,60%
```

Ce remplacement a lieu pour chaque exécution différente de la commande. Cette même commande peut donc servir à vérifier des hôtes différents, mais elle peut être rendue encore plus générique.

Exemple 2 : Argument de commande

Vous pouvez également passer des arguments dans une commande, ce qui permet de garder une définition de commande générique.

Les différents arguments sont séparés par le caractère '!'. On peut donc, dans l'hôte, définir les arguments de la commande check_ping comme étant:



Si vous devez utiliser le caractère (!) dans les arguments de votre commande, vous pouvez éviter son interprétation en le préfixant d'un anti-slash (\). si vous avez besoin de l'anti-slash dans une commande, il suffit de doubler l'anti-slash (\\).

```
200.0,80%!400.0,40%
```

Ces arguments deviennent alors disponibles dans la commande par les macros \$ARGn\$. \$ARG1\$ sera remplacé en "200.0,80%" et \$ARG2\$ en "400.0,40%". La définition de la commande peut alors être réécrite :

```
/var/lib/Shinken Enterprise/libexec/check_ping -H $HOSTADDRESS$ -w $ARG1$ -c $ARG2$
```

Exemple 3 : Utilisation des données

Qu'en est-il si plusieurs hôtes partagent les mêmes arguments de commande ?

Le plus simple est de définir un template (voir la [Logique des templates](#)) contenant ces données spécifiques.

Nous allons créer un template contenant des données. Pour cet exemple, nous les nommerons WARNINGPING et CRITICALPING, et contiendront ces valeurs.



Dans l'interface de configuration, elles sont disponibles dans l'onglet de données du template. Dans un fichier d'import, il sera nécessaire de les préfixer avec ' _ '.

On spécifiera dans le template que la commande est appelée avec, en argument, ces données:

```
$_HOSTWARNINGPING$!$_HOSTCRITICALPING$
```

La commande n'a pas besoin d'être modifiée, de par l'application réursive des macros.

Cela ouvre une nouvelle possibilité: si, on imagine, que sur un hôte donné, le seuil doit être différent, il suffira alors de surcharger les données WARNINGPING et CRITICALPING dans l'hôte, et la commande lancée sera spécialisée pour cet hôte uniquement.

Utilisation avancée : Données à la demande

Normalement, lorsque vous utilisez des données d'hôte et de service dans la définition des commandes, elles se réfèrent à des valeurs pour l'hôte ou le service pour lequel la commande est en cours d'exécution. Par exemple, si une commande de vérification d'un hôte est lancée sur un nom d'hôte "linuxbox", toutes les données "ref:standard<thebasics/datalist>" se référeront aux valeurs de cet hôte ("linuxbox").

Si vous voulez référencer des valeurs pour un autre hôte ou service dans une commande, (pour laquelle la commande n'est pas lancée) vous pouvez utiliser ce qu'on appelle les données "à la demande". Ces données sont normales, excepté le fait qu'elles contiennent un identifiant pour le service ou l'hôte dont elles doivent tirer la valeur. Voici le format simple de données à la demande

- * "\$HOSTdataNAME:host_name\$"
- * "\$SERVICEdataNAME:host_name:service_description\$"

Remplacez "HOSTdataNAME" et "SERVICEdataNAME" avec le nom de données standards trouvées d'un hôte ou service :ref:here <thebasics/datalist>.

Note: le nom de données est séparé de l'identifiant de l'hôte ou du service par le séparateur 2 points (:). Pour les données de service à la demande, l'identifiant du service est constitué à la fois du nom de l'hôte et de la description du service . Ils sont également séparés par les 2 points (:).

Les données de service à la demande peuvent avoir un nom d'hôte vide, dans ce cas on prend automatiquement le nom associé au service .

Exemples de données de service et d'hôte à la demande :

- \$HOSTDOWNTIME:myhost\$ // données d'hôte à la demande
- \$SERVICESTATEID:server:database\$ // données de service à la demande
- \$SERVICESTATEID::CPU Load\$ // données de service à la demande sans nom d'hôte

Les données à la demande sont également disponibles pour les hostgroup, servicegroup, contact, et contactgroup datas. Par exemple:

- \$CONTACTEMAIL:john\$ // données contact à la demande
- \$CONTACTGROUPMEMBERS:linux-admins\$ // données contactgroup à la demande
- \$HOSTGROUPALIAS:linux-servers\$ // données hostgroup à la demande
- \$SERVICEGROUPALIAS:DNS-Cluster\$ // données servicegroup à la demande

Nettoyage des données

Certaines données sont nettoyées d'une enveloppe de méta caractères potentiellement dangereux avant d'être échangés dans la commande à exécuter. Le choix des caractères à nettoyer dépend du paramétrage de la valeur "illegal_data_output_chars". Les données suivantes sont nettoyées de caractères potentiellement dangereux :

- \$HOSTOUTPUT\$
- \$LONGHOSTOUTPUT\$

- \$HOSTPERFDATA\$
- \$HOSTACKAUTHOR\$
- \$HOSTACKCOMMENT\$
- \$SERVICEOUTPUT\$
- \$LONGSERVICEOUTPUT\$
- \$SERVICEPERFDATA\$
- \$SERVICEACKAUTHOR\$
- \$SERVICEACKCOMMENT\$