Le Receiver

Concept général

Les macros sont des objets spéciaux permettant d'effectuer des remplacements de données dans les différents éléments de Shinken Entreprise. Elles sont utilisées pour permettre une factorisation de la configuration, ainsi que pour fournir une grande flexibilité dans la création et modification d'une configuration de supervision.

Un exemple pour illustration

On dispose d'une commande qui se charge de contacter un hôte pour déterminer si il est joignable ou non.

On veut donc que la commande récupère automatiquement l'adresse de l'hôte sans avoir à spécifier manuellement l'adresse pour chaque hôte.

Pour résoudre ce problème, on utilise donc une macro. Dans Shinken Entreprise, on peut utiliser la macro $HOSTADDRESS$ qui va contenir l'adresse de l'hôte courant.

Ainsi, en utilisant cette macro dans notre commande, lorsque la commande sera utilisée lors de la vérification d'un hôte, l'adresse de l'hôte sera automatiquement remplacée dans la commande.

Commande avant remplacement
`check_ping -H "$HOSTADDRESS$" (...autres paramètres)`

Commande après remplacement
`check_ping -H "192.168.1.12" (...autres paramètres)`

Les différents types de macros

Dans l'exemple précédent, une macro est utilisée pour permettre le remplacement dynamique d'une donnée locale à l'hôte dans une commande.

Il s'agit seulement d'un exemple des différents types de macros existantes. Ces macros peuvent être séparées en 3 grandes catégories:

- Les macros permettant un accès aux données locales
- Les macros spécifiques, permettant de transférer les données
- Les macros permettant un accès aux données globales

Les données locales

Les macros concernant les données locales permettent de faire référence à une donnée ou un attribut d'un objet bien particulier. Ces données locales peuvent être des attributs d'un élément, ou bien des données personnalisées.

Sur une commande qui utilise la

Accès aux attributs d'un élément

Accès aux données personnalisées d'un élément

Macros spécifiques

Référence aux arguments d'un commande

Cas du Duplicate Foreach

Les données globales

Utilisation des macros

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.

Le nom d'une donnée peut contenir uniquement des caractères alphanumériques (a-zA-Z0-9).

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 modèle (voir la Logique des modèles) contenant ces données spécifiques.

Nous allons créer un modèle 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 modèle. Dans un fichier d'import, il sera nécessaire de les préfixer avec '_'.

On spécifiera dans le modèle 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 récursive 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.

Il est possible d'effectuer de même pour les Checks, et donc d'appeler en argument $_SERVICEWARNINGPING$!$_SERVICECRITICALPING$ ce qui ira prendre les valeurs de la données WARNINGPING et CRITICALPING du Check.