Résolution des problèmes liés à l'import des sources

Introduction

Dans l'interface de Configuration, les sources peuvent être utilisées pour importer des éléments. Chaque objet trouvé dans les sources est ensuite affiché en tant que Nouveau ou présente des Différences sur les objets déjà présents dans Staging.

Lors de l'import d'éléments depuis les sources, des erreurs ou avertissements peuvent être remontés. Ces différents avertissements et leur résolution sont décrits dans les sections suivantes.

Sommaire

Impossible d'écrire dans le fichier de configuration

Lors de l'import d'un fichier de configuration, Shinken peut avoir besoin d'écrire dedans pour y mettre des informations sur les objects qu'il vient d'importer (pour éviter de les importer en double).

Si Shinken n'a pas les droits d'écriture sur un fichier de configuration qu'il essaye d'importer, l'import de la source est stoppé et le message d'erreur suivant est affiché.

```
CRITIQUE: Le chargement de la configuration a échoué ave
erreurs suivantes :
* Cannot open config file '/etc/shinken-user/source-data/so
-data-cfg-testpack-import/./elements/test.cfg' for reading:
no 13] Permission denied: u'/etc/shinken-user/source-data
ce-data-cfg-testpack-import/./elements/test.cfg'
```

Pour résoudre ce problème, il faut donner les droits en écriture sur le fichier en question (le chemin vers le fichier est présent dans le message d'erreur). La manière la plus simple est d'attribuer la propriété du fichier à l'utilisateur Shinken:

```
chown shinken:shinken mon_fichier.cfg
```

D'autres méthodes permettant à l'utilisateur "shinken" d'écrire dans le fichier de configuration peuvent être utilisées.

Propriété manquante

Certains élements importés sont ignorés lorsqu'il manque une propriété obligatoire. Quand c'est le cas, un message affichant la propriété manquante est affiché dans le résultat de la source.

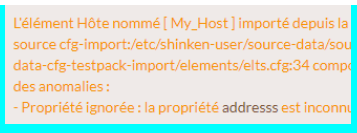
Dans l'exemple, la source contient un hôte sur lequel la propriété "*host_name*" n'est pas définie.

```
L'élément Hôte nommé [ ] importé depuis la source
import:/etc/shinken-user/source-data/source-data
testpack-import/elements/elts.cfg:34 comporte de
anomalies :
- Nom d'élément manquant
```

Propriété inconnue

Quand une propriété non obligatoire est inconnue, l'élément est importé mais la propriété inconnue est ignorée. La propriété est affichée dans le message d'avertissement sur la source qui contient l'élément.

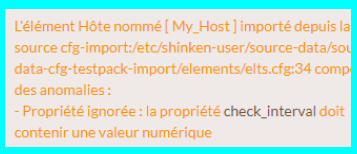
Dans l'exemple, on voit qu'une erreur a été faite sur le nom du champ "*address*".



Propriété non valide ignorée

Quand une propriété définie dans une source possède une valeur incorrecte, cette propriété est ignorée, et l'erreur est indiqué.

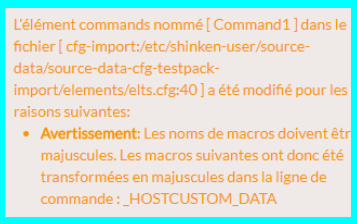
Dans l'exemple, la valeur de la propriété "*check_interval*" doit être un entier positif ou -1, mais une chaîne de caractères a été spécifiée. Le champ est donc ignoré pour cet élément.



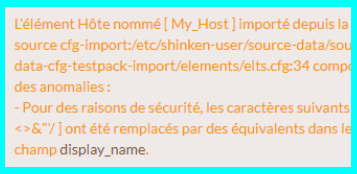
Propriété non valide modifiée

Des propriétés valides mais malformées peuvent également être importés, avec un reformatage effectué par le mécanisme d'import des sources. Lorsque l'import des sources modifie certaines valeurs de propriétés d'éléments importés, un message d'avertissement indique l'opération effectuée sur la propriété.

Dans le premier exemple, Shinken indique que la donnée _HOSTCUSTOM_DATA trouvée dans la commande "*Command1*" a été mise en majuscule pour des raisons de cohérence.



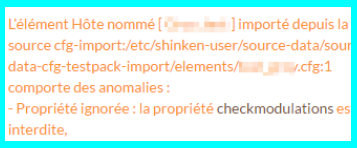
Le deuxième exemple indique que la propriété *display_name* de l'hôte "*My_Host*" a été transformée pour des raisons de sécurité



Propriété interdite

Quand une propriété non obligatoire est interdite (clé obsolète), l'élément est importé mais la propriété interdite est ignorée. La propriété est affichée dans le message d'avertissement sur la source qui contient l'élément.

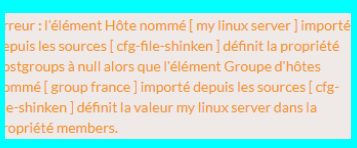
Dans l'exemple, on voit que la propriété "*checkmodulations*" tente d'être utilisée, alors que cette clé n'est plus valide dans la version de Shinken Enterprise.



Incohérence entre un élément et un groupe

Lors de l'import si un groupe possède un membre et que ce même membre possède dans sa définition un groupe défini à 'null' l'import ne peux choisir si l'élément est ou n'est pas dans le groupe.

Vous devez modifier la définition de l'élément.



Conflits lors de l'import des objets

Précisions sur le fonctionnement de l'import des sources

Le mécanisme d'import permet également d'importer un objet dont la définition est répartie entre plusieurs sources. Un hôte par exemple, peut être présent dans 2 sources (ou plus):

- La première (fichier Cfg), définit un certain nombre de propriétés, comme son nom, adresse ainsi que quelques modèles à utiliser
- La deuxième (Découverte réseau), définit un certain nombre de propriétés supplémentaires comment des modèles d'hôte à utiliser.

L'objet final sera donc un hôte ayant les propriétés de la première et de la deuxième source.

Lorsque les sources importées, l'interface de Configuration rassemble ces objets suivant différents critères. D'une manière générale, des objets sont rassemblés si ils possèdent au moins une clé de synchronisation commune.

Les clés de synchronisation utilisées dépendent du type de l'élément:

- **Hôtes**: Nom, adresse, SE_UUID (Identifiant unique pouvant être défini dans les fichiers de configuration)
- **Checks**: SE_UUID
- **Autres**: Nom de l'élément, SE_UUID

Par exemple, 2 hôtes avec la même adresse seront regroupés par le mécanisme d'import des sources en un seul hôte. Les 2 hôtes suivants:

```
define host {
    host_name           Hote 1
    address             localhost
    propriété_commune   valeur1
    propriété1          valeur1
}


define host {
    host_name           Hote 2
    address             localhost
    propriété_commune   valeur2
    propriété1          valeur2
}
```

seront regroupés en un seul hôte contenant les 2 propriétés:

```
define host {
    host_name           Hote 1
    address             localhost
    propriété_commune   valeur1
    propriété1          valeur1
    propriété2          valeur2
}
```

L'hôte résultant de la fusion des 2 hôtes ayant une clé de synchronisation commune contient les propriétés définies sur les hôtes ayant cette clé commune.

Lorsque la même propriété est définie par plusieurs hôtes, la première valeur définie est prise. Dans l'exemple, on voit que la propriété commune a pour valeur la première valeur définie, soit "*valeur1*".

Cas spécial du SE_UUID

Parmi les clés de synchronisation d'un objet, le SE_UUID est traité d'une manière différente.

Lorsqu'un élément possède un SE_UUID dans sa définition, il est considéré comme unique dans le mécanisme d'import des sources, et il ne sera pas fusionné avec un élément ayant un SE_UUID différent. Il peut par contre toujours être fusionné avec des éléments sans SE_UUID ou avec d'autres éléments ayant le même SE_UUID.

C'est utile dans le cas où on veut définir 2 hôtes avec la même adresse par exemple. Le SE_UUID permet de dire qu'il s'agit de 2 hôtes distincts et ils ne seront pas fusionnés.

```
define host {

   # Shinken Enterprise. Lines added by import core. Do not remove it, it's used by Shinken Enterprise to
update your objects if you re-import them.
      _SE_UUID             core-service-60f9373a47d511e8a3c1080027a3ae8c
      _SE_UUID_HASH        61917cb49e8f3b524ee2e9482f59a2e9
   # End of Shinken Enterprise part

    host_name               Host 1
    address                 host_addr
}

define host {

   # Shinken Enterprise. Lines added by import core. Do not remove it, it's used by Shinken Enterprise to
update your objects if you re-import them.
      _SE_UUID             core-service-6454ba8a47d511e897f8080027a3ae8c
      _SE_UUID_HASH        bc7ea10870bdcb985dd1ba4dbcc62688
   # End of Shinken Enterprise part


    host_name               Host 2
    address                 host_addr
    retry_interval          4
    check_running_timeout   4
}
```

Les 2 hôtes ne seront pas fusionnés à cause de la présence du SE_UUID.

Un SE_UUID à la forme suivante:

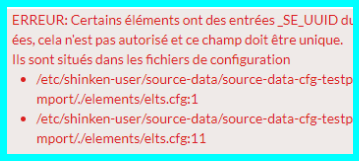
> _UUID core-**type_element**-**identifiant_unique**

- **type_element**: Le type de l'élément sur lequel on veut spécifier ce SE_UUID. Les différentes valeurs possibles sont les différents types définissables par fichier de configuration: host, service, contact,...
- **identifiant_unique**: Chaine de caractères (alphanumériques) permettant d'identifier de manière unique l'élément.

SE_UUID dupliqué

Erreur

Le SE_UUID permet d'identifier de manière unique un élément dans un fichier de configuration.



Lorsque le même SE_UUID est présent 2 fois dans les sources, Shinken marque les 2 objets ayant le même SE_UUID comme invalides et n'importe pas la source contenant l'erreur pour ne pas provoquer d'erreurs supplémentaires.

Cette erreur est courante lorsqu'on copie un fichier ou la définition d'un élément sans changer les SE_UUID des objets.

Résolution

- L'erreur à l'import indique le fichier et la ligne des éléments à problème. Supprimer ou changer les SE_UUID de ces objets suffit pour résoudre le conflit.

L'élément peut être fusionné avec plusieurs éléments existants

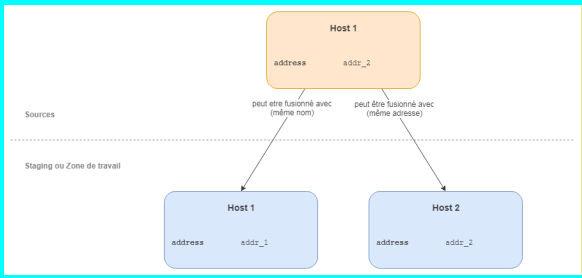
Erreur

Quand un élément est importé depuis les sources, Shinken regarde dans les différents espaces de données s'il existe un élément existant avec lequel il peut être fusionné.

Par exemple, si on définit un hôte dans une source, et qu'il existe déjà un hôte avec la même adresse ou le même nom dans Staging, Shinken fusionne ces hôtes car il y a de très grandes chances qu'il s'agisse de la même machine.

Si un élément importé d'une source peut être fusionné avec plusieurs éléments existants, Shinken n'effectue pas de fusion arbitraire, mais affiche un avertissement indiquant l'erreur.

Le schéma suivant illustre ce cas de figure:



Elément ignoré: Shinken ne peut pas déterminer dans
élément Hôte existant intégrer les propriétés de l'élé
Hôte nommé [Host 1] importé depuis les sources [cf
testpack-import,syncui] .
Les elements Hôtes existants trouvés sont :

- Host 1
- Host 2

Tous ces éléments ont des clés de synchronisation communes (par exemple nom, adresse, SE_UUID, ...).
Veuillez consulter la page de documentation des sour
pour corriger ce problème.

Résolution

- Changer le nom et/ou l'adresse de l'élément importé. Un conflit de nom est très souvent une erreur d'inattention. Changer le nom de l'hôte permet de résoudre le conflit. L'élément importé depuis les sources sera alors fusionné avec l'élément de Staging qui a la même adresse (ou autre clé de synchronisation)

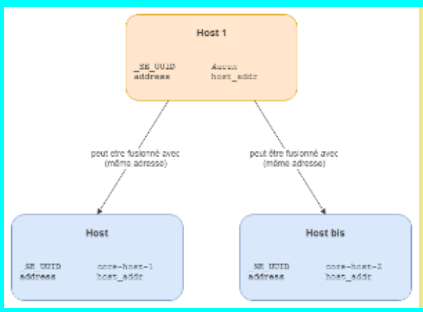
L'éléments peut être fusionné avec plusieurs éléments des sources (clés de synchronisation communes avec d'autres éléments mais SE_UUID différents)

Erreur

Lorsque des éléments sont importés avec des SE_UUID, Shinken les empeche d'être fusionnés avec des éléments de SE_UUID différents. Un élément présent dans une source qui ne possède pas de SE_UUID peut être fusionné avec un autre élément qui a une clé de synchronisation commune.

Mais, quand un élément d'une source a plusieurs éléments distincts avec lesquels il peut être fusionné, Shinken ne peut pas décider avec quel élément le fusionner. Dans ce cas, plutôt que de choisir arbitrairement un élément pour la fusion des propriétés, Shinken affiche un avertissement indiquant le problème.

Le schéma suivant illustre le problème:



Élément ignoré: L'élément Hôte nommé [Host 1] importé depuis les sources [cfg-file-testpack-import] est en conflit avec les 3 éléments suivants:

- L'élément host nommé [Host] importé depuis la source [cfg-file-testpack-import] dans le fichier [cfg-import:/etc/shinken-user/source-data/source-data-cfg-testpack-import/elements/elts.cfg:1] avec les clés [host_addr, host,core-host-719e2ebc486911e88974080027a3ae8c]
- L'élément host nommé [Host bis] importé depuis la source [cfg-file-testpack-import] dans le fichier [cfg-import:/etc/shinken-user/source-data/source-data-cfg-testpack-import/elements/elts.cfg:12] avec les clés [host_addr,host bis,core-host-75ae7296486911e88532080027a3ae8c]
- L'élément host nommé [Host 1] importé depuis la source [cfg-file-testpack-import] dans le fichier [cfg-import:/etc/shinken-user/source-data/source-data-cfg-testpack-import/elements/elts.cfg:23] avec les clés [host 1, host_addr]

Tous ces éléments contiennent des clés de synchronisation communes mais des SE_UUID différents. Shinken ne peut pas déterminer comment regrouper les propriétés de ces éléments. Veuillez consulter la page de documentation pour corriger ce problème.

Résolution

Plusieurs solutions sont envisageables pour résoudre ce problème:

- Si il ne faut pas que l'hôte "*Host 1*" soit fusionné, lui positionner un SE_UUID.
- Si "*Host 1*" doit être fusionné systématiquement avec "*Host*" ou "*Host bis*", il faut lui mettre le même SE_UUID que l'élément avec lequel il doit être fusionné. Une autre solution, si applicable, peut être de ne pas répartir la définition d'un même élément dans plusieurs fichiers de configuration pour éviter les problèmes de ce type.