

Zone d'entête

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.

Propriété manquante

Certains éléments 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.

? Unknown Attachment

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*".

? Unknown Attachment

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.

? Unknown Attachment

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

? Unknown Attachment

? Unknown Attachment

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 sont 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 assemblé avec aucun autre élément.

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**: Chaîne de caractères (alphanumériques) permettant d'identifier de manière unique l'élément.

TODO

Un script présent dans l'archive d'installation de Shinken Entreprise permet de générer ces SE_UUID:

```
lib/nom_du_script_pour_générer_les_SE_UUID -h
```


Expliquer comment fonctionne le script

SE_UUID dupliqué

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.

 Unknown Attachment

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

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

Lorsque des éléments sont importés avec des SE_UUID, Shinken les considère comme des objets uniques.

L'élément possède des clés de synchronisation communes avec un autre élément et a le même nom qu'un autre élément de Staging ou de la Zone de travail

- Conflit SE_UUID (2 fois le meme SE_UUID, conflit de merge SEF-2807)