

# 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

- [Introduction](#)
- [Impossible d'écrire dans le fichier de configuration](#)
- [Propriété manquante](#)
- [Propriété inconnue](#)
- [Propriété non valide ignorée](#)
- [Propriété non valide modifiée](#)
- [Propriété interdite](#)
- [Incohérence entre un élément et un groupe](#)
- [Conflits lors de l'import des objets](#)

## 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 objets 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.

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 /etc/shinken-user/source-data/cfg-documentation/edition_element/commands.cfg
```

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

• Le chargement de la configuration a échoué avec les erreurs suivantes :  
\* [config] cannot open config file '/etc/shinken-user/source-data/cfg-documentation/edition\_element/commands.cfg' for reading: [Errno 13] Permission denied: u'/etc/shinken-user/source-data/cfg-documentation/edition\_element/commands.cfg'



### Erreur listée pour la recherche sur cette page

Le chargement de la configuration a échoué avec les erreurs suivantes :

```
* [config] cannot open config file '/etc/shinken-user/source-data/cfg-documentation/edition_element/commands.cfg' for reading: [Errno 13] Permission denied: u'/etc/shinken-user/source-data/cfg-documentation/edition_element/commands.cfg'
```

## 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.

• L'élément Hôte nommé [ ] comporte des anomalies qui ont empêché son import  
• Nom d'élément manquant



### Erreur listée pour la recherche sur cette page

L'élément Hôte nommé [ ] importé depuis la source `cfg-import:/etc/shinken-user/source-data/source-data-cfg-testpack-import/elements/elts.cfg:34` comporte des 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".

- L'élément Hôte nommé [ customer4-switch ] comporte des anomalies :
  - Propriété ignorée : la propriété **address** est inconnue



### Erreur listée pour la recherche sur cette page

L'élément Hôte nommé [ customer4-switch ] comporte des anomalies :

- Propriété ignorée : la propriété **address** est inconnue

## Propriété non valide ignorée

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

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.

- L'élément Hôte nommé [ customer4-switch ] comporte des anomalies :
  - Propriété ignorée : la propriété check\_interval doit contenir une valeur nu



### Erreur listée pour la recherche sur cette page

L'élément Hôte nommé [ customer4-switch ] comporte des anomalies :

- Propriété ignorée : la propriété check\_interval doit contenir une valeur numérique

## Propriété non valide modifiée

Des propriétés valides, mais malformées peuvent également être importées, 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 cet exemple, Shinken indique que la donnée \_HOSTCUSTOM\_DATA trouvée dans la commande "Command1" a été mise en majuscule pour des raisons de cohérence.

- L'élément Commande nommé [ command1 ] comporte des anomalies :
  - Le nom de la donnée `_hostcustom_data` doit être en majuscule. Il a été transformé lors de l'import en `_HOSTCUSTOM_DATA`.



### Erreur listée pour la recherche sur cette page

L'élément "commands" nommé [ Command1 ] importé depuis la source [cfg-import/etc/shinken-user/source-data/source-data-cfg-testpack-import/elements/elts.cfg:40](#) a été modifié pour les raisons suivantes:

- Avertissement: Les noms de macros doivent être en majuscules. Les macros suivantes ont donc été transformées en majuscules dans la ligne de commande : `_HOSTCUSTOM_DATA`

## 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.

- L'élément Hôte nommé [ customer-6 ] comporte des anomalies :
  - Propriété ignorée : la propriété **checkmodulations** est int



### Erreur listée pour la recherche sur cette page

L'élément Hôte nommé [ customer-6 ] comporte des anomalies :

- Propriété ignorée : la propriété **checkmodulations** est interdite

## 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 peut choisir si l'élément est ou n'est pas dans le groupe.

Vous devez modifier la définition de l'élément.

• Erreur : l'élément hôte nommé [ customer-6 ] importé depuis les sources [ documentation.sync, la propriété hostgroups à null alors que l'élément Groupe d'hôtes nommé [ France ] importé c source [ documentation ] définit la valeur customer-6 dans la propriété members.



### Erreur listée pour la recherche sur cette page

Erreur : l'élément Hôte nommé [ customer-6 ] importé depuis les sources [ documentation, syncui ] définit la propriété hostgroups à null alors que l'élément Groupe d'hôtes nommé [ France ] importé depuis la source [ documentation ] définit la valeur customer-6 dans la propriété "members".

## 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 sont importées, l'interface de Configuration mélange les éléments qui 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 présents dans deux sources distinctes :

#### Source 1

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

#### Source 2

```
define host {
    host_name      Hote 2
    address        localhost
    propriété_commune valeur2
    propriété2     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 valeur effective est celle dont la source a le plus petit numéro d'ordre. Dans l'exemple, on voit que la propriété commune a pour valeur la première valeur définie, soit "*valeur1*" et que le nom de l'hôte est "*Hote 1*".

## 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**: Chaîne 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.

• Certains éléments ont des entrées \_SE\_UUID dupliquées, cela n'est pas autorisé et ce champ unique  
ils sont situés dans les fichiers de configuration  
/etc/shinken-user-source-data/cfg-documentation/edition\_element/hosts.cfg:40  
/etc/shinken-user-source-data/cfg-documentation/edition\_element/hosts.cfg:59

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

### Erreur listée pour la recherche sur cette page

Certains éléments ont des entrées \_SE\_UUID dupliquées, cela n'est pas autorisé et ce champ doit être unique.

Ils sont situés dans les fichiers de configuration

- /etc/shinken-user/source-data/cfg-documentation/edition\_element/hosts.cfg:40
- /etc/shinken-user/source-data/cfg-documentation/edition\_element/hosts.cfg:59

## 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.
- 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)

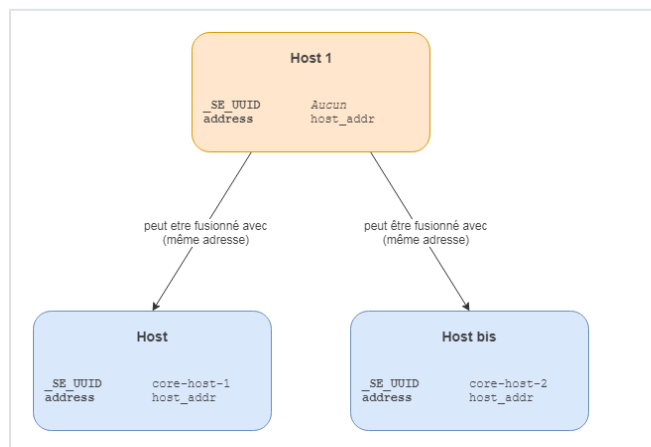
## L'éléments peut être fusionné avec plusieurs éléments des sources (clés de synchronisation communes avec d'autres éléments mais SE\_UUID différents)

### Erreur

Lorsque des éléments sont importés avec des SE\_UUID, Shinken les empêche 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:



### Problème concernant les clés de synchronisations

(Ce problème spécifique pouvant générer de longues listes de conflits, seuls les 2 premiers cas rencontrés sont affichés pour éviter la lenteur de chargement de la page.)

1. Pour fusionner différentes définitions sur la même source, les définitions doivent avoir les mêmes clés de synchronisation. Dans ce cas, les clés de synchronisation possibles sont les propriétés : `host_name`, `address`. Les clés présentées en rouge sont les clés en conflit avec l'élément défini dans `cfg-file-sample/etc/shinken-user/source-data/source-data-cfg-sample/elements/hosts.cfg:1`. Voir le fonctionnement des clés de synchronisation dans la documentation.

| Élément défini depuis   | Clés de synchronisations                        |
|---|---|
| <code>cfg-file-sample/etc/shinken-user/source-data/source-data-cfg-sample/elements/hosts.cfg:1</code> | <code>host_addr</code> , <code>host1</code>     |
| <code>cfg-file-sample/etc/shinken-user/source-data/source-data-cfg-sample/elements/hosts.cfg:8</code> | <code>host_addr</code> , <code>host1_bis</code> |

2. Pour fusionner différentes définitions sur la même source, les définitions doivent avoir les mêmes clés de synchronisation. Dans ce cas, les clés de synchronisation possibles sont les propriétés : `host_name`, `address`. Les clés présentées en rouge sont les clés en conflit avec l'élément défini dans `cfg-file-sample/etc/shinken-user/source-data/source-data-cfg-sample/elements/hosts.cfg:8`. Voir le fonctionnement des clés de synchronisation dans la documentation.

| Élément défini depuis   | Clés de synchronisations                        |
|---|---|
| <code>cfg-file-sample/etc/shinken-user/source-data/source-data-cfg-sample/elements/hosts.cfg:1</code> | <code>host_addr</code> , <code>host1</code>     |
| <code>cfg-file-sample/etc/shinken-user/source-data/source-data-cfg-sample/elements/hosts.cfg:8</code> | <code>host_addr</code> , <code>host1_bis</code> |

- L'élément Hôte nommé [ `host1` ] comporte des anomalies qui ont empêché son import :
  - Cet Hôte est concerné par le problème de clés de synchronisations présenté dans le résumé de cette page,
- L'élément Hôte nommé [ `host1_bis` ] comporte des anomalies qui ont empêché son import :
  - Cet Hôte est concerné par le problème de clés de synchronisations présenté dans le résumé de cette page,



#### Erreur listée pour la recherche sur cette page

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.