

# Voir la configuration du module ( cfg-file-import )

## Sommaire

- Présentation
- Configuration générale
- Options de mélange des sources
  - Mode de mélange des sources
  - Détecter les nouveaux éléments
  - Calculer les différences
  - Suppression des éléments absents
    - Détecter les éléments qui ne sont plus présent dans la source
- Choix de l'espace où seront placé les éléments importés
  - Mettre dans l'espace
  - Utilisateur utilisé pour la sauvegarde des changements
- Clés spécifiques à la source
  - Configuration du fichier de définition des données de la source
  - Import de fichiers de données globales
  - Création d'un \_SE\_UUID pour chaque élément de la source
  - Récupération du \_SE\_UUID des éléments existants pour l'écrire dans le fichier
- Précisions techniques
  - Clés de synchronisation
    - Propriétés par défaut utilisé pour la construction des clés de synchronisation

## Présentation

L'onglet de configuration afficher les paramètres de configuration définies dans le fichier de configuration de votre module peut être consultée dans cet onglet ( voir la page [Collecteur de type cfg-file-import \( format Shinken ou nagios \)](#) ) :

Elle est divisée en 5 chapitres :

- Configuration générale
- Options de mélange des sources
- Choix de l'espace où seront placé les éléments importés
- Paramètres spécifiques à la source
- Clé de synchronisation ( sync\_key )

Remarque : *Cet onglet ne permet pas encore d'éditer la configuration ( => Sera implémenté dans une prochaine version )*

## Configuration générale

Clé	Valeur
Ordre	2
Intervalle d'import	5
Modules	Mon-Collecteur-de-Fichier
Type de module	cfg-file-import

Ce chapitre contient tous les paramètres pour le chargement et fonctionnement des sources en général.

Nom du paramètre dans l'interface	Nom du paramètre dans le fichier	Description
Modules	<input type="text" value="source_name"/>	Nom de la source affichée dans l'interface de configuration en page d'accueil. <ul style="list-style-type: none"><li>• <b>Dois être unique,</b></li><li>• d'une <b>longueur inférieure à 40 caractères,</b></li><li>• et ne dois <b>pas contenir le caractère "\$".</b></li></ul>
Type de Module	<input type="text" value="module_type"/>	Type de module chargé par le Synchroniser. Pour cette source le module_type est "cfg-file-import".

Intervalle d'import	<input type="text" value="import_interval"/>	<p>Délai écoulé entre les imports automatiques de la source.</p> <ul style="list-style-type: none"> <li>• Un nombre ( <i>en minutes</i> )</li> <li>• Si 0, la source ne sera jamais exécutée automatiquement.</li> </ul>
Ordre	<input type="text" value="order"/>	<p>L'ordre de la source dans l'interface de configuration, qui a un impact dans la fusion des données lors des imports de sources.</p> <ul style="list-style-type: none"> <li>• Un nombre</li> <li>• Voir la page du <a href="#">Synchronizer</a> pour plus d'information au sujet des fusions.</li> </ul> <p>Remarque : <i>Si vous changez l'ordre depuis l'interface ( page d'accueil ), le fichier cfg sera mis à jour.</i></p>

## Options de mélange des sources

Clé	Valeur
compute_deleted_element_not_in_source	disabled (La valeur "source_space" de la propriété "put_in" rend cette propriété inactive)
compute_element_difference	authorized
compute_new_element	authorized
merge_mode	all

## Mode de mélange des sources

Grâce au paramètre "Mode de mélange des sources" ( *merge\_mode dans le fichier de configuration de la source* ) il est possible de modifier comment est gérée une source lors de l'étape du mélange des sources.

Voir la page : [Concept général et utilisation des sources](#)

La valeur par défaut est : **all**

Les valeurs possibles sont :

Nom du paramètre dans l'interface	Nom du paramètre dans le fichier		
merge_mode	<input type="text" value="merge_mode"/>	Ce paramètre peut avoir les valeurs suivantes :	
		Valeurs dans le fichier	Description
		dont_merge_with_other	<p>Les éléments récoltés par cette source seront importés, mais pas fusionnés avec les autres sources.</p> <p>Ce qui implique que si un élément de cette source possède une clé de synchronisation commune avec une autre source alors ces éléments ne pourront être importés et provoquera un conflit.</p> <p>A utiliser si votre source vous donne un élément complet et qui ne doit pas être modifié ni par l'interface de configuration ni d'autres sources.</p>
		only_with_syncui	<p>Si un élément importé par cette source est aussi récolté par "Syncui", ils seront fusionnés pour ne donner qu'un élément.</p> <p>A utiliser si votre source vous donne un élément complet et que vous vous voulez le modifier via l'interface de configuration.</p>
		all_without_syncui	<p>Si un même élément est importé par plusieurs sources (à part "Syncui"), alors les éléments seront fusionnés pour n'en donner qu'un.</p> <p>A utiliser si votre source vous donne un élément à compléter avec d'autres sources mais qui ne doit pas être modifié par l'interface de configuration.</p>
		<b>all (valeur par défaut)</b>	<p>Si un même élément est importé par plusieurs sources, alors les éléments seront fusionnés pour n'en donner qu'un.</p> <p>Cas par défaut : les éléments de cette source seront complétés avec les autres sources et l'interface de configuration.</p>



Syncui est une source qui importe tous les éléments de l'interface de configuration.

Chaque fois qu'une source est importée, elle l'est aussi. Si une source ne se mélange pas avec Syncui, les modifications sur l'interface de configuration ne seront pas prises en compte .



Les sources désactivés ne sont pas prises en compte lors du mélange des sources.

## Détecter les nouveaux éléments

Les éléments proposés par la source sont soumis au calcul des nouveautés ( cf. [Les Modules de détection et de qualification](#) ).

Nom du paramètre dans l'interface	Nom du paramètre dans le fichier		
compute_new_element	compute_new_element	Ce paramètre peut avoir les valeurs suivantes :	
		Valeurs dans le fichier	Description
		authorized <i>(valeur par défaut)</i>	Activée => le mécanisme de calcul des nouveautés analysera les éléments de la source pour identifier les nouveaux éléments par rapport à l'espace de données Staging.
		never_new	Désactivée => les éléments de la source n'apparaîtront pas en tant que nouveau, même s'ils n'ont jamais été importés par rapport à l'espace de données Staging.

## Calculer les différences

Les éléments proposés par la source sont soumis au calcul des différences ( cf. [Les Modules de détection et de qualification](#) ).

Nom du paramètre dans l'interface	Nom du paramètre dans le fichier		
compute_element_differece	compute_element_difference	Ce paramètre peut avoir les valeurs suivantes :	
		Valeurs dans le fichier	Description
		authorized <i>(valeur par défaut)</i>	Activée => le mécanisme de calcul des différences analysera les éléments de la source pour identifier toute différence par rapport à l'espace de données Staging.
		never_differece	Désactivée => les éléments de la source n'apparaîtront pas en tant que différence, même s'ils possèdent des différences par rapport a l'espace de données Staging.

## Suppression des éléments absents

### Détecter les éléments qui ne sont plus présent dans la source

Permet de proposer en suppression, les éléments qui ne sont plus présents dans la ou les source/s.

Nom du paramètre dans l'interface	Nom du paramètre dans le fichier		
-----------------------------------	----------------------------------	--	--

compute_deleted_element_not_in_source	compute_deleted_element_not_in_source	Ce paramètre peut avoir les valeurs suivantes :	
		Valeurs dans le fichier	Description
		disabled <i>(valeur par défaut)</i>	Les éléments qui ne sont plus présents dans la(les) source(s) ne seront pas proposés à la suppression.
		delete_if_missing_in_all_sources	Les éléments, qui ne sont plus présents dans cette source et qui n'existent pas dans les autres sources (sauf Syncui), seront proposés à la suppression.
		delete_if_missing_in_this_source	Les éléments qui ne sont plus présents dans cette source seront proposés à la suppression.

## Choix de l'espace où seront placés les éléments importés

Clé	Valeur
put_in	source_space
put_in_user	shinken-core

## Mettre dans l'espace

Permet d'appliquer les différences et d'importer les nouveaux éléments de cette source directement en Staging ou Production ou bien de les laisser dans l'espace de calcul des sources comme par défaut.

Nom du paramètre dans l'interface	Nom du paramètre dans le fichier		
put_in	put_in	Ce paramètre peut avoir les valeurs suivantes :	
		Valeurs dans le fichier	Description
		source_space <i>(valeur par défaut)</i>	Les éléments sont sauvegardés dans l'espace de donnée "Sources". Cette espace sera comparée avec l'espace de données "Staging" afin de calculer les nouveautés et les différences.
		staging	Les éléments sont sauvegardés dans les espaces de données "Espace de calcul des Sources" et "Staging".
		production	Les éléments sont sauvegardés dans les espaces de données "Espace de calcul des Sources", "Staging" et "Production".
		production_and_reload_arbiter	Les éléments sont sauvegardés dans les espaces de données "Espace de calcul des Sources", "Staging" et "Production". L'Arbiter est redémarré afin de soumettre les changements à la supervision.



**À noter:** Une source avec le paramètre "put in" à la valeur "production" ou "production\_and\_reload\_arbiter" ne poussera en production que les changements qui proviennent de la source et non tous les changements à appliquer en production (tous les changements visibles dans la page de production).

Ce qui implique que si un utilisateur supprime un élément importé par une source avec le paramètre "put\_in" à la valeur "production" ou "production\_and\_reload\_arbiter" alors que l'élément est réimporté par la source : il y aura en production 2 éléments avec le même nom, car la suppression manuelle de l'élément ne sera pas prise en compte ce qui provoquera une erreur.

Une source avec l'option put\_in à production\_and\_reload\_arbiter ne redémarrera l'Arbiter que si elle a des modifications à pousser en production.

## Utilisateur utilisé pour la sauvegarde des changements

Option qui permet de choisir l'utilisateur qui apparaîtra comme étant le dernier à avoir mis à jour les éléments.

 **À Noter:** l'utilisateur doit être obligatoirement un administrateur Shinken.

Nom du paramètre dans l'interface	Nom du paramètre dans le fichier	Valeurs dans le fichier	Description
put_in_user	<code>put_in_user</code>	shinken-core	Les nouveaux éléments et les éléments différents importés par cette source auront la valeur de cette clé comme utilisateur faisant la dernière modification.

## Clés spécifiques à la source

Clé	Valeur
cfg_path	/etc/shinken-user/source-data/source-data_Mon-Collecteur-de-Fichier/definition_Mon-Collecteur-de-Fichier.cfg
create_and_write_SEUID_in_file	0
update_cfg_with_staging_se_uuid	0

Ce chapitre liste tous les paramètres qui sont spécifiques au fonctionnement de cette source.

## Configuration du fichier de définition des données de la source

Le fichier de définition des données de la source est le fichier qui est désigné par le `cfg_path` dans la configuration de la source. Ce fichier doit être unique, mais il permet de définir plusieurs emplacements à importer.

Il présente trois paramètres :

paramètre	Exemple	Description
<code>cfg_dir</code>	<code>cfg_dir=.</code>	Il permet d'indiquer à Shinken de lire le contenu d'un dossier et de ses sous-dossiers. Tous les fichiers se terminant par <code>.cfg</code> seront lu et importé par cette source.
<code>cfg_file</code>	<code>cfg_file=mes_hotes.cfg</code>	Il permet d'indiquer un seul fichier à lire. Ce n'est pas obligé de se terminer par <code>.cfg</code> , mais il doit en conserver la syntaxe
<code>resource_file</code>	<code>resource_file=mon_infrastructure</code>	Il permet d'indiquer un seul fichier à lire. Ce n'est pas obligé de se terminer par <code>.cfg</code> , mais il doit en conserver la syntaxe

Ces paramètres peuvent être cumulés afin de lire plusieurs répertoires ou plusieurs fichiers.

Voici un exemple de ce fichier de définition qui permet de lire les dossiers et fichiers suivants :

- `elements_prod_Paris`
- `elements_prod_Marseille`
- `infratrucure_reseau`

Si d'autres dossiers ou fichiers sont présents dans le répertoire de données de la source, ils ne seront pas lus dans cet exemple.

### Exemple de fichier de définition

```
cfg_dir=elements_prod_Paris
cfg_dir=elements_prod_Marseille
cfg_file=infratrucure_reseau
```

 La syntaxe des fichiers importé par ces définition est rédigée dans cette page : [Syntaxe des fichiers d'imports](#)

## Import de fichiers de données globales

Pour une organisation plus facile et une configuration plus propre, les données globales peuvent également être configurées dans les sources.

- Le dossier d'exemple d'une source cfg est situé dans `/etc/shinken-user-example/configuration/daemons/synchronizers/sources/cfg-file-import/source-data_cfg-file-import_example` et contient un dossier `global-data` dans lequel les définitions des données globales peuvent être placées (un fichier échantillon est également présent dans ce dossier)
- Lorsque la source est importée, ces fichiers de données globales seront copiés dans `/etc/shinken/resource.d/source_name`

Ce mécanisme vous permet également de surcharger les valeurs définies dans `/etc/shinken/resource.d` avec des fichiers définis dans les sources.

Par contre, ces fichiers contenant des définitions de donnée globale ne sont pris en compte **qu'après le rechargement du Synchronizer**.

## Création d'un `_SE_UUID` pour chaque élément de la source

Si vous souhaitez importer des éléments et que ceux-ci ne se mélangent pas avec les éléments existants, la source peut créer un identifiant unique (`_SE_UUID`).

Voici son fonctionnement :

- Lors du premier import d'un élément, la source va regarder si cet élément dispose d'un `_SE_UUID`.
- Si ce n'est pas le cas, alors un `_SE_UUID` sera généré pour cet élément
- Cet identifiant sera écrit dans le fichier de l'élément et permettra de conserver l'unicité de cet élément.

Nom du paramètre dans l'interface	Nom du paramètre dans le fichier	Description
create_and_write_SEUUID_in_file	create_and_write_SEUUID_in_file	Permet de créer un identifiant unique ( <code>_SE_UUID</code> ).  Deux valeurs possible <ul style="list-style-type: none"> <li>• <b>0</b> : ce paramètre n'est pas actif ( défaut )</li> <li>• <b>1</b> : ce paramètre est actif</li> </ul>

## Récupération du `_SE_UUID` des éléments existants pour l'écrire dans le fichier

Si vous souhaitez fusionner avec un élément existant en staging et conserver ce lien même si toutes les propriétés utilisées en tant que clés de synchronisation sont modifiées, la source peut écrire le `_SE_UUID` de l'élément existant dans le fichier cfg.

Voici son fonctionnement :

- Lors du premier import d'un élément, la source va regarder si cet élément dispose d'un `_SE_UUID`.
- Si ce n'est pas le cas, alors une recherche d'un élément existant sera réalisée à partir des clés de synchronisation.
- Si un élément dispose des mêmes clés de synchronisation, alors la source va écrire le `_SE_UUID` de cet élément dans le fichier cfg et l'ajouter dans la définition de l'élément.
- Une fois importé, l'élément ne sera pas en nouveau, mais pourra apporter des différences avec l'élément existant



La recherche d'un élément existant se base uniquement

- sur les éléments présents en staging.
- À partir des clés de synchronisation définies sur la source au moment de l'import

Nom du paramètre dans l'interface	Nom du paramètre dans le fichier	Description
update_cfg_with_staging_se_uuid	update_cfg_with_staging_se_uuid	Permet de rechercher un élément existant à partir des clés de synchronisation et récupérer son <code>_SE_UUID</code>  Deux valeurs possible <ul style="list-style-type: none"> <li>• <b>0</b> : ce paramètre n'est pas actif ( défaut )</li> <li>• <b>1</b> : ce paramètre est actif</li> </ul>

## Précisions techniques

### Clés de synchronisation

Les clés de synchronisation sont des valeurs utilisées lors de l'étape du mélange des sources ( Voir la page [Modules de Sources \( imports \) et de Taggers \( qualification \)](#) ) qui permet de choisir quel élément de cette source se mélange avec quel élément d'une autre source ( Voir la page [Le mélange des sources & les clés de synchronisation \(sync-key\)](#) ).

Le paramètre `properties_used_as_synckey` de la source permet d'ajouter les propriétés qui serviront à créer les clés de synchronisation ( Voir la page [Collecteur de type `cfg-file-import` \( format `Shinken` ou `nagios` \)](#) ).

### Propriétés par défaut utilisé pour la construction des clés de synchronisation

Propriétés par défaut utilisé pour la construction des clés de synchronisation :

Propriété	Type d'élément	Info
Nom de l'élément	Tous les éléments	Cette propriété ne peut pas être retirée des propriétés utilisées pour faire les clés de synchronisation
<code>_SE_UUID</code>	Tous les éléments	Cette propriété ne peut pas être retirée des propriétés utilisées pour faire les clés de synchronisation
<code>address</code>	hôte	