

Collecteur de type `cfg-file-import` (format Shinken ou nagios)

Sommaire

- Concept
- Activation du collecteur
 - Ajouter un collecteur d'import de fichier `.cfg`
 - Activer le collecteur d'import de fichier `cfg-file-nagios` livré par défaut
- Configuration
 - Exemple de fichier de configuration
 - Détails des sections composant le fichier de configuration
 - Identification de la source
 - Options de mélange des sources
 - Mode de mélange des sources
 - Détecter les nouveaux éléments
 - Calculer les différences
 - Détecter les éléments qui ne sont plus présents dans la source
 - Choix de l'espace d'import des éléments
 - Mettre dans l'espace
 - Utilisateur utilisé pour la sauvegarde des changements
 - Intervall d'import et ordre de la source
 - Emplacement des fichiers de configuration
 - Clés de synchronisation (`sync_key`)
 - Ajout des `SE_UUID` dans les fichiers de configuration
 - Propriétés non récupérées
- Configuration du fichier de définition des données de la source
- Définition de Données globales

Concept

Cette source permet d'importer de nouveaux éléments via des fichiers `.cfg`.

La syntaxe d'import de ces fichiers est la même que la syntaxe Nagios, plus des ajouts de Shinken Entreprise:

- Il est donc possible d'importer des fichiers de configuration Nagios dans Shinken, et ainsi migrer sa configuration Nagios vers Shinken Entreprise facilement.
- Les fichiers de configuration Shinken Framework sont évidemment supportés et importables dans Shinken Entreprise.

Cette page vous explique comment définir ce type de collecteur et la page [Collecteur de type \(`cfg-file-import` \) - Import depuis des fichiers au format `.cfg`](#) vous décrit comment l'utiliser.

Activation du collecteur

Vous pouvez essayer ce type de source soit en activant **vos propres collecteurs de type `cfg-file-import`** ou en activant le collecteur `cfg-file-nagios` présents par défaut.

Ajouter un collecteur d'import de fichier `.cfg`

Vous pouvez avoir plusieurs sources du type `cfg-file-import`, pour par exemple trier les éléments.

Choisissez un nom pour ce nouveau collecteur.

- Pour l'exemple, nous allons l'appeler "Mon-Collecteur-de-Fichier".
- Remplacer dans l'exemple le mot "Mon-Collecteur-de-Fichier" par le nom que vous aurez choisi.

Faites les opérations suivantes :

- Copier le fichier de définition de la source d'exemple : `/etc/shinken-user-example/configuration/daemons/synchronizers/sources/cfg-file-import/cfg-file-import-example.cfg` dans le répertoire de définition des sources **`/etc/shinken/sources/`** et modifier son nom.
(Exemple : `/etc/shinken/sources/collector__cfg-file-import__Mon-Collecteur-de-Fichier.cfg`)

```
cp /etc/shinken-user-example/configuration/daemons/synchronizers/sources/cfg-file-import/cfg-file-import-example.cfg /etc/shinken/sources/collector__cfg-file-import__Mon-Collecteur-de-Fichier.cfg
```

- Mettre les droits d'écriture au fichier

```
chmod -R 664 /etc/shinken/sources/collector__cfg-file-import__Mon-Collecteur-de-Fichier.cfg
```

- Ouvrir ce fichier (collector__cfg-file-import__Mon-Collecteur-de-Fichier.cfg) :
 - Modifier la ligne source_name en remplaçant le nom par défaut "cfg-file-example" par le nom que vous avez choisi "**Mon-Collecteur-de-Fichier**".

```
source_name          Mon-Collecteur-de-Fichier
```

- Modifier le chemin du fichier de définition vers celui présent dans le dossier de votre source

```
cfg_path             /etc/shinken-user/source-data/source-data__Mon-Collecteur-de-Fichier/definition__Mon-Collecteur-de-Fichier.cfg
```

- Copier le dossier d'exemple de la source dans le répertoire des données des sources **/etc/shinken-user/source-data/** et donner lui un nom. (Exemple : */etc/shinken-user/source-data/source-data__Mon-Collecteur-de-Fichier*)

```
cp -r /etc/shinken-user-example/configuration/daemons/synchronizers/sources/cfg-file-import/source-data__cfg-file-import__example /etc/shinken-user/source-data/source-data__Mon-Collecteur-de-Fichier/
```

- Renommer le fichier de définition des chemins de la source pour lui donner le nom de votre source

```
mv /etc/shinken-user/source-data/source-data__Mon-Collecteur-de-Fichier/definition-cfg-file-import-example.cfg /etc/shinken-user/source-data/source-data__Mon-Collecteur-de-Fichier/definition__Mon-Collecteur-de-Fichier.cfg
```

- Par défaut, *definition__Mon-Collecteur-de-Fichier.cfg* indique que le collecteur importera tous les fichiers qui sont dans ce répertoire et les sous-répertoires. Mais vous pouvez le modifier cela (regarder le chapitre [Configuration du fichier de définition des données de la source](#) pour plus de détail).
- Chaque collecteur peut importer ses propres variables globales (regarder le chapitre [Définition de Données globales](#)).
- Une fois que les fichiers ont été édités, vérifiez que les droits sont bons (*droit de lecture pour l'utilisateur Shinken*). Si ce n'est pas le cas, effectuez la commande suivante :

```
chmod -R 764 /etc/shinken-user/source-data/source-data__Mon-Collecteur-de-Fichier
chown -R shinken:shinken /etc/shinken-user/source-data/source-data__Mon-Collecteur-de-Fichier
```

- Ajouter le nom de la nouvelle source au Synchronizer en modifiant le paramètre **sources** du fichier **/etc/shinken/synchronizers/synchronizer-master.cfg**.

```
define synchronizer {
    [...]
    sources          Source 1, Source 2, Source 3, Mon-Collecteur-de-Fichier
    [...]
}
```

- Redémarrez le Synchronizer pour qu'il puisse prendre en compte cette nouvelle source

```
service-shinken-synchronizer restart
```

Activer le collecteur d'import de fichier cfg-file-nagios livré par défaut

Par défaut, l'installation ou la mise à jour de Shinken Entreprise va mettre à disposition une définition de collecteur d'import de fichier Nagios au format .cfg.

- La configuration de ce collecteur se trouve par défaut dans le fichier : **/etc/shinken/sources/cfg-file-nagios.cfg**

- Le collecteur **cfg-file-sample** s'active comme les autres sources, c'est-à-dire en modifiant le fichier **/etc/shinken/synchronizers/synchronizer-master.cfg** (ou le *.cfg* que vous utilisez pour définir les options du Synchronizer)
 - Ce collecteur est déjà présent dans la liste des sources livrées par défaut, il n'y a donc rien à modifier.
 - S'il n'y est pas (*car modifié*), vous pouvez le rajouter pour qu'il soit de nouveau actif.

Exemple :

```
sources      source1,source2,source3,cfg-file-nagios
```

- Redémarrez le Synchronizer pour qu'il puisse prendre en compte cette nouvelle source

```
service-shinken-synchronizer restart
```

Configuration

Voici le détail du fichier de configuration de la source qui se trouve (*suivant la procédure choisie*) :

- Soit dans le fichier que vous venez de créer en ajoutant le module (*par exemple /etc/shinken/sources/collector__cfg-file-import__Mon-Collecteur-de-Fichier.cfg*).
- Soit dans le fichier **/etc/shinken/sources/cfg-file-nagios.cfg**.

Exemple de fichier de configuration

Vous trouverez un exemple dans **/etc/shinken-user-example/configuration/daemons/synchronizers/sources/cfg-file-import/cfg-file-import-example.cfg**

```
#####
# cfg-file-import-example
#####
# Daemons that can load this source:
# - synchronizer
# This source import the cfg-config sample from Shinken Enterprise update.
#####

define source {

    #==== Source identity =====
    # Source name. Must be unique
    source_name          cfg-file-import-example

    # Module type (to load module code). Do not edit.
    module_type          cfg-file-import

    # description: display a description on the interface for this source
    description          This source is about loading cfg files compatible with Shinken

    # enabled: is this source enabled or not
    enabled              1

    #==== Import interval and order ====
    # import_interval: in minutes, what is the schedule import interval for this source.
    # note: 0 = don't schedule this source, will run only if an administrator launch it from the interface
    import_interval      5

    # order: source order for a source imply if a source is before an another source when
    # merging data
    order                1

    #==== Location =====
    # Location of the source definition file. This file will be used to define where the .cfg files are to
    be imported.
    cfg_path             /etc/shinken-user/source-data/source-data-cfg-file-
import-example/definition-cfg-file-import-example.cfg
```

```

#=====  

# The list of properties to be used as sync_keys in addition to the item name. Properties not managed by  

# Shinken can be added here.  

# properties_used_as_synckey          address  

#=====  

# Properties not stored  

# Properties which can be defined in the items from the source but which Shinken will not import.  

# not_stored_properties  

#=====  

# With this option if item in cfg hasn't a SE_UUID, the source will ask the synchronizer to search a  

# match in staging or in working area. If found, it will be inserted in the file.  

# update_cfg_with_staging_se_uuid    0  

# With this option if item in cfg hasn't a SE_UUID, the source will create it and write in the file  

# create_and_write_SEUUID_in_file    0 }

```

Détails des sections composant le fichier de configuration

Identification de la source

```

...
#=====  

# Source identity  

# Source name. Must be unique  

source_name          cfg-file-import-example  

# Module type (to load module code). Do not edit.  

module_type          cfg-file-import  

# description: display a description on the interface for this source  

description          This source is about loading cfg files compatible with Shinken  

# enabled: is this source enabled or not  

enabled              1  

...

```

Il est possible de définir plusieurs instances de module de type `cfg-file-import` dans votre architecture Shinken.

- Chaque instance devra avoir un nom unique.

Nom	Type	Unité	Défaut	Commentaire
source_name	Texte	---	cfg-file-import-example	<p>Nous vous conseillons de choisir un nom en fonction de l'utilisation du module pour que votre configuration soit simple à maintenir.</p> <p>Chaîne de caractères composée de lettres, de chiffres et des caractères _ et - .</p> <ul style="list-style-type: none"> • Doit être unique. • D'une longueur maximum à 40 caractères. • Ne dois pas contenir les caractères ?, &, ", ', \$, /, #, \, ;
module_type	Texte	---	cfg-file-import	Ne peut être modifié
description	Texte	---		Contient une description de la source qui vous permet de retrouver l'objectif de cette source
enabled	Booléen	---	0	Permet d'activer/désactiver la source.

Remarque

Si vous activer/desactiver la source depuis l'interface (page d'accueil), le fichier .cfg sera mis à jour.

Options de mélange des sources

```
# #
# MERGE OPTIONS #
# #

# Behaviour of mixing of sources
#
# Default : all => Merge with all source
# ... : dont_merge_with_other => Don't merge with other source
# ... : only_with_syncui => Merge only with syncui
# ... : all_without_syncui => Merge with all source without syncui
#
# merge_mode all

# New element calculation
#
# Default : authorized ( Authorized the calculation of new items )
# ... : never_new ( Do not compute new items )
#
# compute_new_element authorized

# Element difference calculation
#
# Default : authorized ( Authorized the calculation of differences )
# ... : never_difference ( Do not compute differences )
#
# compute_element_difference authorized

# Specific treatment with elements that are no longer present in the source
#
# Default : disabled
# ... : delete_if_missing_in_all_sources
# ... : delete_if_missing_in_this_source
#
# compute_deleted_element_not_in_source disabled
```

Mode de mélange des sources

Option qui définit le mode de mélange entre les hôtes récupérés de cette source et les hôtes des autres sources (voir la page [Voir la configuration du module \(synchronize-collector-vmware \)](#)).

```
# Source merge mode
# - all : Merge with all source ( Default )
# - dont_merge_with_other : Don't merge with other source
# - only_with_syncui : Merge only with syncui
# - all_without_syncui : Merge with all source without syncui
# merge_mode all
```

Nom	Type	Unité	Défaut	Commentaire				
merg e_mo de	Enumé ration	---	all	Ce paramètre peut avoir les valeurs suivantes :				
				<table border="1"><thead><tr><th>Nom</th><th>Description</th></tr></thead><tbody><tr><td>dont_merg</td><td>Les éléments récoltés par cette source seront importés, mais pas fusionnés avec les autres sources.</td></tr></tbody></table>	Nom	Description	dont_merg	Les éléments récoltés par cette source seront importés, mais pas fusionnés avec les autres sources.
				Nom	Description			
dont_merg	Les éléments récoltés par cette source seront importés, mais pas fusionnés avec les autres sources.							

e_with_oth er	S'il existe le même élément importé par une autre source, l'élément ne sera pas importé.
only_with_ syncui	Si un même élément existe déjà dans "Syncui", ils seront tous les deux fusionnés pour n'en donner qu'un.
all_without_ _syncui	Si un même élément est importé par plusieurs sources (à part "Syncui"), alors les éléments seront fusionnés pour n'en donner qu'un.
all	Si un même élément est importé par plusieurs sources, alors les éléments seront fusionnés pour en donner qu'un.



À noter: Syncui est une source (interne à shinken) qui importe tous les éléments de l'interface de configuration. Chaque fois qu'une source est importée, elle l'est aussi.

Détecter les nouveaux éléments

Les éléments proposés par la source sont soumis au calcul des nouveautés (Voir la page [Modules de Sources \(imports \)](#) et de [Taggers \(qualification \)](#)).

```
# Detecting new elements
# - authorized ( Default )
# - never_new
# compute_new_element authorized
```

Nom	Type	Unité	Défaut	Commentaire						
compute_ne w_element	Énumér ation	---	authoriz ed	Ce paramètre peut avoir les valeurs suivantes : <table border="1" data-bbox="586 894 1466 1066"> <thead> <tr> <th>Nom</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>authori zed</td> <td>L'option est 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.</td> </tr> <tr> <td>never_ new</td> <td>L'option est 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.</td> </tr> </tbody> </table>	Nom	Description	authori zed	L'option est 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	L'option est 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.
Nom	Description									
authori zed	L'option est 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	L'option est 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

```
# Calculate the differences
# - authorized ( Default )
# - never_difference
# compute_element_difference authorized
```

Nom	Type	Unité	Défaut	Commentaire						
compute_eleme nt_difference	Enumér ation	---	authoriz ed	Ce paramètre peut avoir les valeurs suivantes : <table border="1" data-bbox="630 1499 1466 1694"> <thead> <tr> <th>Nom</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>authori zed</td> <td>L'option est 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.</td> </tr> <tr> <td>never_ differe nce</td> <td>L'option est désactivée les éléments de la source n'apparaîtront pas en tant que différence, même s'ils possèdent des différences par rapport à l'espace de données Staging.</td> </tr> </tbody> </table>	Nom	Description	authori zed	L'option est 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_ differe nce	L'option est désactivée les éléments de la source n'apparaîtront pas en tant que différence, même s'ils possèdent des différences par rapport à l'espace de données Staging.
Nom	Description									
authori zed	L'option est 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_ differe nce	L'option est désactivée les éléments de la source n'apparaîtront pas en tant que différence, même s'ils possèdent des différences par rapport à l'espace de données Staging.									

Détecter les éléments qui ne sont plus présents dans la source

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



Cette option ne peut être activée que si cette source met directement dans l'espace "Staging" ou "Production" ses éléments importés.

```
# Detect elements that are no longer present in the source
# - disabled ( Default )
# - delete_if_missing_in_all_sources
# - delete_if_missing_in_this_source
# compute_deleted_element_not_in_source disabled
```

Nom	Type	Unité	Défaut	Commentaire								
compute_deleted_element_not_in_source	Enumération	---	disabled	<p>Ce paramètre peut avoir les valeurs suivantes :</p> <table border="1"> <thead> <tr> <th>Nom</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>disabled</td> <td>L'option est 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.</td> </tr> <tr> <td>delete_if_missing_in_all_sources</td> <td>L'option est désactivée les éléments de la source n'apparaîtront pas en tant que différence, même s'ils possèdent des différences par rapport à l'espace de données Staging.</td> </tr> <tr> <td>delete_if_missing_in_this_source</td> <td>Les éléments qui ne sont plus présents dans cette source seront proposés à la suppression.</td> </tr> </tbody> </table>	Nom	Description	disabled	L'option est 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.	delete_if_missing_in_all_sources	L'option est désactivée les éléments de la source n'apparaîtront pas en tant que différence, même s'ils possèdent des différences par rapport à l'espace de données Staging.	delete_if_missing_in_this_source	Les éléments qui ne sont plus présents dans cette source seront proposés à la suppression.
Nom	Description											
disabled	L'option est 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.											
delete_if_missing_in_all_sources	L'option est désactivée les éléments de la source n'apparaîtront pas en tant que différence, même s'ils possèdent des différences par rapport à l'espace de données Staging.											
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 d'import des éléments

```
# #
# PUT IN OPTIONS #
# #

# Targeted spaces to put elements
# - source_space : Available for staging ( Default )
# - staging : Saved in staging
# - production : Saved in staging
# : Saved in production
# - production_and_reload_arbiter : Saved in staging
# : Saved in production
# : Available to monitoring
# put_in source_space

# User responsible of import
# accepted name : Only admin shinken name
# >>> admin ( Default )
# put_in_user admin
```

Mettre dans l'espace

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

Nom	Type	Unité	Défaut	Commentaire										
put_in	Enumération	---	source_space	<p>Ce paramètre peut avoir les valeurs suivantes :</p> <table border="1"> <thead> <tr> <th>Nom</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>source_space</td> <td>Les éléments sont sauvegardés dans l'espace de données "Sources". Cet espace sera comparé avec l'espace de données "Staging" afin de calculer les nouveautés et les différences.</td> </tr> <tr> <td>staging</td> <td>Les éléments sont sauvegardés dans les espaces de données "Espace de calcul des Sources" et "Staging".</td> </tr> <tr> <td>production</td> <td>Les éléments sont sauvegardés dans les espaces de données "Espace de calcul des Sources", "Staging" et "Production".</td> </tr> <tr> <td>production_and_reload_arbiter</td> <td>Les éléments sont sauvegardés dans les espaces de données "Espace de calcul des Sources", "Staging" et "Production". L'Arbiter est redémarré après l'import afin de soumettre les changements à la supervision.</td> </tr> </tbody> </table>	Nom	Description	source_space	Les éléments sont sauvegardés dans l'espace de données "Sources". Cet espace sera comparé 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 après l'import afin de soumettre les changements à la supervision.
Nom	Description													
source_space	Les éléments sont sauvegardés dans l'espace de données "Sources". Cet espace sera comparé 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 après l'import 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	Type	Unité	Défaut	Commentaire
put_in_user	Texte	---	shinken-core	La valeur de ce paramètre est mise à jour avec l'utilisateur ayant fait la dernière modification sur les éléments en "Nouveau" et les éléments en "Différence".

Interval d'import et ordre de la source

```
...
#=====  

# import_interval: in minutes, what is the schedule import interval for this source.  

# note: 0 = don't schedule this source, will run only if an administrator launch it from the interface  

import_interval 5  

# order: source order for a source imply if a source is before an another source when  

# merging data  

order 1  

...
```

Il est possible de définir un intervalle d'import afin que la source s'importe régulièrement.

Nom	Type	Unité	Défaut	Commentaire
import_interval	Entier positif	minute	0	Délai écoulé entre les imports automatiques de la source. Si 0, l'import de la source ne sera jamais exécuté automatiquement. (uniquement manuellement)
order	Entier positif	---	2	L'ordre de la source dans l'interface de configuration (<i>A un impact dans la fusion des données lors des imports de sources</i>). Voir la page Le mélange des sources & les clés de synchronisation (sync-key) pour plus d'information au sujet des fusions. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">Remarque <i>Si vous changez l'ordre depuis l'interface (page d'accueil), le fichier .cfg sera mis à jour.</i></div>

Emplacement des fichiers de configuration

```
...
#=====  

# Location of the source definition file. This file will be used to define where the .cfg files are to be imported.
```

```

cfg_path /etc/shinken-user/source-data/source-data-cfg-file-
import-example/definition-cfg-file-import-example.cfg
...

```

Afin de récupérer les éléments Shinken, contenus dans des fichiers .cfg, il faut spécifier un unique fichier de définition de la source qui contiendra la liste des chemins vers les fichiers à importer (regarder le chapitre [Configuration du fichier de définition des données de la source](#) pour plus de détail).

Nom	Type	Unité	Défaut	Commentaire
cfg_path	Texte	---	---	Emplacement du fichier de définition de la source. Ce fichier servira à définir où sont les fichiers .cfg à importer.

Clés de synchronisation (sync_key)

```

...
##### Synchronization keys #####
# The list of properties to be used as sync_keys in addition to the item name. Properties not managed by
Shinken can be added here.
# properties_used_as_synckey address
...


```

Si le paramètre `properties_used_as_synckey` :

- N'est pas défini,
 - sa valeur par défaut est à `address`. Tout élément ayant la propriété `address` aura la valeur de cette propriété ajoutée dans les clés de synchronisation.
- Est vide,
 - aucune propriété ne sera ajoutée dans les clés de synchronisation.
- Est redéfini à `display_name`
 - seule la valeur de cette propriété sera ajoutée dans les clés de synchronisation.
 - Remarque : plusieurs propriétés peuvent être définies comme clef de synchronisation (cela dépendra de vos besoins)

```
properties_used_as_synckey host_name, display_name, address
```

Défini la liste des propriétés qui seront utilisées pour générer les clés de synchronisation.

 **À noter** : On ne peut pas supprimer le nom et le SE_UUID, mais on peut les compléter.

Nom	Type	Unité	Défaut	Commentaire
properties_used_as_synckey	Texte	---	address	Définit la liste de propriétés qui seront utilisées en plus du nom et du SE_UUID de l'élément pour générer les clés de synchronisation (sync_key). Ce paramètre est optionnel. Si ce paramètre n'est pas présent, sa valeur par défaut vaut propriété "address". S'il est défini à vide, la propriété "address" ne sera pas utilisée comme synckey.

Ajout des SE_UUID dans les fichiers de configuration

```

...
##### SE_UUID in cfg file #####
# With this option if item in cfg hasn't a SE_UUID, the source will ask the synchronizer to search a
match in staging or in working area. If found, it will be inserted in the file.
# update_cfg_with_staging_se_uuid 0

# With this option if item in cfg hasn't a SE_UUID, the source will create it and write in the file
# create_and_write_SEUUID_in_file 0
...

```

Si un élément importé dans un des fichiers .cfg ne contient pas de SE_UUID, Shinken peut lui en générer un ou le chercher depuis les éléments existant dans staging. Il est alors possible de faire en sorte que cet SE_UUID soit retranscrit dans le fichier .cfg grâce à ce paramètres.

Nom	Type	Unité	Défaut	Commentaire
update_cfg_with_staging_se_uuid	Booléen	---	0	Ce paramètre permet de retrouver le SE_UUID de l'élément s'il a déjà été importé ou s'il se mélange avec un autre élément. Si un élément existe en staging, le SE_UUID est ajouté dans le fichier .cfg. Cela permet de modifier le nom ou les autres propriétés constituant les sync_keys dans le .cfg, sans perdre le mélange avec l'élément de staging. <ul style="list-style-type: none"> 0 : Ne recherche pas le SE_UUID pour l'écrire dans le fichier .cfg 1 : Recherche le SE_UUID des éléments importés depuis staging et les écrits dans les fichiers .cfg
create_and_write_SEUUID_in_file	Booléen	---	0	Ce paramètre crée un SE_UUID pour chaque élément importé par la source, avant que les éléments ne soient mélangés. Cela empêche les éléments d'être mélangés avec d'autres éléments. Le SE_UUID est ajouté dans le fichier .cfg. <ul style="list-style-type: none"> 0 : Ne crée pas le SE_UUID 1 : Crée et ajoute les SE_UUID des éléments importés dans les fichiers .cfg

Propriétés non récupérées

```
...
##### Properties not stored #####
# Properties which can be defined in the items from the source but which Shinken will not import.
# not_stored_properties
...
```

Il est possible de définir des propriétés que la source ne devra pas récupérer.

Dans le cas de cette source, il se peut que vous ne soyez pas l'auteur des fichiers cfg et que vous ne vouliez pas importer certaines propriétés.

Nom	Type	Unité	Défaut	Commentaire
not_stored_properties	Texte	---	---	Empêche la récupération de certaines propriétés des éléments récoltés

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

Définition de Données globales

Les données globales peuvent être configurées dans ***/etc/shinken/resource.d/***.

- Ce dossier contient des fichiers définissant des données globales pour différents sujets (*email, mysql, oracle, ...*) accessibles par vos objets globalement dans Shinken.
- Vous trouverez plus de détail sur l'utilisation des données globales dans le chapitre [Les Variables \(Remplacement dynamique de contenu - Anciennement les Macros \)](#)

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/source-data/source-data-cfg-sample*** 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/module-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 données globales ne sont pris en compte **qu'après le rechargement du Synchronizer**.