

# shinken-sla-load-archive - Chargement des données SLA extraites dans une archive externe

## Contexte

Ce document décrit comment importer ses propres fichiers CFG à travers sa propre source.

### Important

Attention, nous vous conseillons de créer votre propre source. En effet, la source Nagios (et Shinken) sont des sources qui sont placées dans le répertoire central Shinken (/etc/shinken) et peuvent être soumises à des modifications lors de mises à jour. Par contre, la création de nouvelles sources personnalisées se fera dans le répertoire dédié aux utilisateurs (/etc/shinken-user) et ce répertoire sera donc traité avec grande vigilance par le moteur de mise à jour, car incluant vos données utilisateur.

Voici les étapes à suivre afin de créer une source qui vous permettra d'importer vos propres CFG:

- [Création d'une source](#)
- [Création du répertoire de données de la source](#)
- [Configuration du démon Synchronizer pour déclarer la source](#)

### Sur cette page

- [Concept](#)
- [Options](#)
  - [Option de paramétrage du chargement de l'archive](#)
  - [Options de connexion à la base MongoDB](#)
    - [Options génériques](#)
    - [Options de connexion SSH](#)
    - [Options d'authentification](#)
    - [Options SSL/TLS](#)

## Ce qui est déjà disponible dans l'installation de Shinken

Pour vous rendre la vie plus facile, quelques tâches de configuration ont déjà été faites:

- a. Le type de module "cfg-file-import" est déjà prêt à être utilisé,
- b. La possibilité d'utiliser la source pré-configurée "cfg-file-nagios" pour tester rapidement un import de cfg.

## Utiliser la source pré-configurée cfg-file-nagios

**Attention, encore une fois, pour une mise en production sur le long terme, il est préférable que vous utilisiez votre propre source, passez alors sur le chapitre suivant.**

Pour un essai d'import de fichier cfg, l'installation ou la mise à jour de Shinken va mettre en place la source cfg-file-nagios déjà configurée :

- `cfg-file-nagios`. Vous pouvez la voir dans la page des sources.

Cette source utilise 2 types de configuration de fichiers

- **Les fichiers de définition de la source et de ses modules** : `/etc/shinken/sources/cfg-file-nagios.cfg`
- **Les fichiers de configuration pour l'import de vos éléments via vos fichiers cfg** : `/etc/shinken-user/source-data/source-data-cfg-nagios/definition-source-data-cfg-nagios.cfg`
  - Ce fichier est pré-configuré pour pointer vers votre dossier de configuration nagios. Il vous suffit de saisir le chemin correspondant pour importer facilement et rapidement votre configuration nagios.

Il vous suffira alors d'activer la source pour importer vos éléments définis dans le fichier `nagios.cfg`.

## Créer et utiliser sa propre source et module

### Contexte

Le fait d'avoir plusieurs sources distinctes peut vous aider si vous avez un grand nombre de fichiers `cfg` et que vous souhaitez pouvoir choisir et séparer les imports. Par exemple, vous avez des fichiers `cfg` relatifs à l'organisation de Paris, et d'autres à l'organisation de Bordeaux (Hôtes, Utilisateurs, ..).

A un instant donné, vous pouvez avoir de besoin d'importer les fichiers de Bordeaux uniquement. Ce sera chose possible avec les deux sources distinctes.

Ce chapitre va vous permettre de configurer votre propre source depuis la source exemple : `cfg-file-sample`

? Unknown Attachment



A chaque fois que vous avez besoin de personnaliser Shinken, le répertoire à utiliser est `/etc/shinken-user/`

## Création d'une source

Placez vous à présent dans le répertoire des sources Shinken (dans `/etc/shinken/sources/`) et copiez la source Sample existante (qui a le format d'une source standard sur laquelle vous pouvez vous baser pour créer une source personnalisée) :

```
cd /etc/shinken/sources/  
cp cfg-file-sample.cfg cfg-file-ma-source.cfg
```

### ? Unknown Attachment

Modifiez le fichier `cfg-file-ma-source.cfg`

Supprimez les 4 lignes allant de

**# Shinken Enterprise**

à

**# End of Shinken Enterprise part**

Modifiez

```
source_name      cfg-file-sample  
  cfg_path      /etc/shinken-user/source-data/source-data-cfg-sample/definition-source-data-cfg-  
sample.cfg
```


avec

```
source_name      cfg-file-ma-source  
  cfg_path      /etc/shinken-user/source-data/source-data-cfg-ma-source/definition-source-data-cfg-  
ma-source.cfg
```

(ce chemin sera celui du répertoire de vos fichiers `cfg` dans le `shinken-user`)

En ce qui concerne les paramètres des sources, voici leurs descriptions :

Propriété	Exemple	Description
<code>source_name</code>	<code>cfg-file-ma-source</code>	Nom de la source affichée dans l' <b>UI de configuration</b> en page d'accueil. <b>Doit être unique</b> et d'une <b>longueur inférieure à 40 caractères</b> , sans quoi le Synchronizer ne démarrera pas.
<code>order</code>	2	L'ordre de la source sur la page, qui a un impact dans la fusion des données lors des imports de sources. Voir la page du <a href="#">Synchronizer</a> pour plus d'information au sujet des fusions. Cette propriété est éditée directement si vous changez l'ordre depuis l'interface.

import_interval	0	<p>La source sera exécutée automatiquement toutes les <i>import_interval</i> minutes.</p> <ul style="list-style-type: none"> <li>• Si 0, la source ne sera jamais exécutée automatiquement.</li> </ul> <div style="border: 1px solid green; padding: 10px; margin-top: 10px;"> <p> <b>Conseil</b></p> <p>Pour vos tests initiaux, nous vous conseillons de positionner la valeur à 0. Le bouton <b>Force import</b> de la page principale vous permettra de faire l'import manuellement.</p> <div style="text-align: center; margin-top: 10px;"> <span style="background-color: #f0f0f0; padding: 5px 10px; border: 1px solid #ccc;">? Unknown Attachment</span> </div> </div>
module-type	cfg-file-import	Définition du type de module utilisé par la source.
enabled	1	Activer ou désactiver la source. Cette propriété est éditée directement si vous utiliser le bouton Activer de l'interface.
description	Cette source permet de charger mes fichiers cfg relatifs aux nouvelles commandes	Description de la source
properties_used_as_sync_key	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 champ "address". S'il est défini à vide, le champ "address" ne sera pas utilisé comme synckey.
update_cfg_with_staging_se_uuid	1	<p>S'il est activé, ce paramètre permet récupérer l'identifiant SE_UUID d'un élément du Synchronizer ( de Staging ) et de l'écrire pour cette élément dans le fichier .cfg de votre collecteur. Il faut que l'élément existe déjà dans l'espace Staging.</p> <p>Note: pour cela il faut que l'utilisateur Shinken est les droits d'écriture sur les fichiers .cfg.</p>

## Création du répertoire de données de la source

Pour créer son propre répertoire dans le répertoire des sources de shinken-user :

```
cd /etc/shinken-user/source-data
cp -r source-data-cfg-sample/ source-data-cfg-ma-source/
mv source-data-cfg-ma-source/definition-source-data-cfg-sample.cfg source-data-cfg-ma-source/definition-source-data-cfg-ma-source.cfg
```

Rajouter les droits sur le répertoire :

```
cd /etc/shinken-user/source-data
chown -R shinken:shinken source-data-cfg-ma-source/
```



### Astuce

Une fois le répertoire créé, les fichiers cfg pourront alors être placés directement dans le répertoire source-data-cfg-ma-source/elements ou source-data-cfg-ma-source/packs  
 Dans tous les cas, tous les fichiers cfg dans le répertoire et sous répertoires de source-data-cfg-ma-source seront traités.

## Configuration du démon Synchronizer pour déclarer la source

## ? Unknown Attachment

Modifiez le fichier `/etc/shinken/synchronizers/synchronizer-master.cfg`

A la fin de la ligne "sources", rajoutez votre source.

Exemple :

```
sources          syncui , cfg-file-shinken , active-dir-example , sync-vmware , cfg-file-nagios , discovery ,
openldap-example , cfg-file-ma-source
```

Redémarrez le synchronizer pour qu'il prenne acte de la nouvelle source :

```
service shinken-synchronizer restart
```

## Affichage de la source dans l'UI de configuration

### ? Unknown Attachment

Une fois le démon redémarré, votre nouvelle source personnalisée doit apparaître :

Son paramétrage sera alors en place : son état (activée ou désactivée), son ordre, sa fréquence en automatique ou en manuel.

## Import des objets

Depuis la page d'accueil de l'interface de configuration, vous pouvez alors utiliser le bouton "**Force**

**import**" en cliquant sur le bouton

### ? Unknown Attachment

Dans le panel des éléments, vos nouveaux éléments doivent apparaître. Vous pourrez alors les importer dans votre base.

Depuis une page spécifique de source vous pouvez également appuyer sur le Bouton

### ? Unknown Attachment

, (il ne sera disponible que si la source est active). Une pop-up sera alors affichée jusqu'à la fin de l'import, ensuite la page sera rechargée pour être mise à jour.

## Précisions techniques

### Clés de synchronisation

Les clés utilisées pour la synchronisation sont des propriétés des objets importés par les sources.

- Le fonctionnement et l'utilité des clés de synchronisation sont décrits de manière plus détaillée dans la page de documentation dédiée: [Précisions techniques sur le fonctionnement de l'import des sources](#) ( **Il est important que vous lisiez cette page avant d'aller plus loin dans ce chapitre** ).

Le module d'import de fichier `cfg` utilise le paramètre `properties_used_as_synckey` de son fichier de configuration pour lister les propriétés de l'objet qui seront accrochées à un élément importé en tant que clés de synchronisation.

- Si un élément importé a une ou plusieurs propriétés remplies lors de l'import, le Synchronizer les rajoutera à la liste des valeurs de synchronisation de cet objet.
- 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 son "`address`" ajouté dans les clés de synchronisation.*
  - est vide,
    - aucune propriété ne sera ajoutée dans les clés de synchronisation.
  - est redéfini à `display_name`
    - seul 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
```

- Si dans une même source, deux éléments importés ont toutes leurs clés de synchronisation sont communes, les deux éléments seront fusionnés.  
Par exemple, pour des clés de synchronisation sur le nom et l'adresse :  
Si Hôte 1 et Hôte 2 ont le même nom et la même adresse, c'est deux élément seront fusionnées.  
Par contre si Hôte 1 et Hôte 2 ont le même nom mais pas la même adresse, ces éléments ne seront pas fusionnés, car toutes les clés ne sont pas communes



Remarque: le Synchronizer ajoute automatiquement le nom de l'élément comme clé de synchronisation.