# Le mélange des sources & les clés de synchronisation (sync-key)

**Sommaire**


- Le mélange des sources
  - Exemple d'un mélange de source
- Les clés de synchronisation
  - Définition
    - Exemple de fonctionnement des clés de synchronisation
  - Sources permettant de choisir ses clés de synchronisation
  - Comment choisir ses clés de synchronisation
  - Cas du _SE_UUID
    - Exemple de génération du couple _SE_UUID / _SE_UUID_HASH par code
  - Voir les clés de synchronisation
- Modification du mode de mélange des sources
  - Mode de mélange : "Toutes"
  - Mode de mélange : "Tout sauf Syncui"
  - Mode de mélange : "Ne fusionne pas avec les autres"
  - Mode de mélange : "Seulement avec "Syncui"


## Le mélange des sources

Après la collecte des données d'une source, elle propose les éléments Shinken qu'elle a pu constituer à l'étape suivante :  le mélange des sources ( *voir la page* Modules de Sources ( imports ) et de Taggers ( qualification ) ).

- Le mélange des sources permet de fusionner les données collectées par les différentes sources si elles concernent un même élément ( *exemple : un hôte défini dans une source et les droits d'accès à cet hôte dans une autre source* ).
- Les données collectées sont fusionnées en fonction des **clés de synchronisation** définies sur chaque source. permettent de faire le lien entre les  choisir quelle définition va être fusionnée avec quelle autre définition.

- Règle de mélange :
  - Si une définition possède au moins une clé de synchronisation en commun avec une autre définition, alors ces 2 définitions fusionnent.
  - Les éléments d'une même source possédant au moins une clé de synchronisation en commun doivent avoir toutes leurs clés de synchronisation identiques, sinon ces éléments seront en erreur
    ( *Exemple pour un hôte de la source cfg-file : même nom et même adresse* ).
  - Les propriétés utilisées pour récupérer les clés de synchronisation dépendent du type de l'élément et de la source.
    - Les pages des sources listent les clés utilisées par défaut.

Pour chaque élément, le Synchronizer va :

1. Regarder si d'autres éléments ont des clés de synchronisation en commun avec lui.
   a. Si oui, il va récupérer toutes les propriétés des éléments en question.
2. Générer un nouvel élément.
3. Donner à l'élément généré toutes les propriétés récupérées ainsi que celles de l'élément d'origine.

L'élément généré est celui qui sera envoyé dans l'étape suivante.

En cas de conflit entre des éléments ( *deux sources importent le même élément, mais avec des informations différentes* ), on compare les priorités des sources et on garde les informations de la source ayant la priorité la plus élevée.

La priorité d'une source est indiquée par son numéro dans la colonne "Ordre" dans l'interface ( *définis par le paramètre `"order"` du fichier de configuration de la source se trouvant dans* ***/etc/shinken/sources*** ).

⚠️ Plus ce numéro est **bas** et plus la priorité de la source est **élevée**. ( *une source avec un ordre 2 sera prioritaire à une source ordre 5* ) ⚠️

Une source peut renvoyer un champ dont le nom se termine par "[FORCE]" :

```
members[FORCE]
```

La valeur de ce champ ne sera pas fusionnée à celle des autres champs, mais l'écrasera.

Si une autre source force également le même champ, on gardera les valeurs de celui dont la source a la priorité la plus élevée ( *voir la page* Forcer la valeur des noms des éléments et des propriétés de type liste (comme la propriété des modèles hérités) ).

## Exemple d'un mélange de source

Exemple avec 2 sources :

- *Ma source d'import par fichier* : source de type Collecteur de type cfg-file-import ( format Shinken ou nagios )
- *Discovery* :  source de type Collecteur de type discovery-import ( Scan NMAP )

Les 2 sources ont leurs clés de synchronisation configurées par défaut, c'est-à-dire :

Pour les hôtes de la source "*Ma source d'import par fichier*" :

- host_name ( *le nom de l'hôte* )
- _SE_UUID
- address

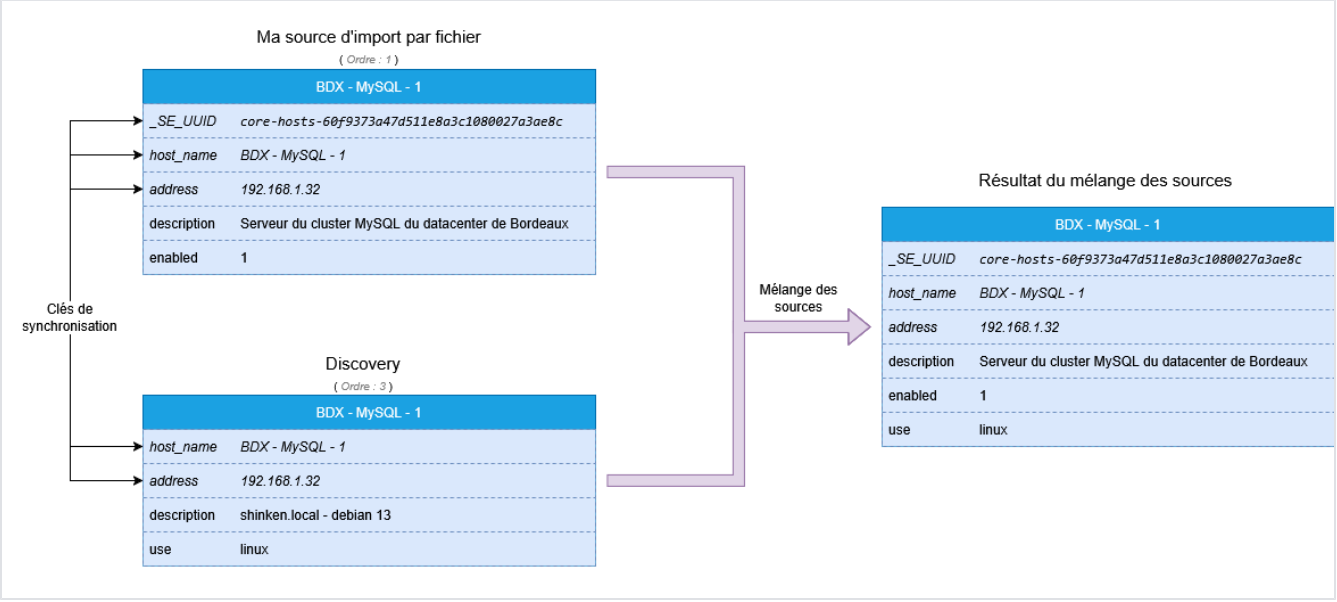
Pour les hôtes de la source *Discovery* :

- host_name ( *le nom de l'hôte* )
- address

Les 2 sources donnent les définitions suivantes :

- Ma source d'import par fichier
  - _SE_UUID  core-hosts-60f9373a47d511e8a3c1080027a3ae8c
  - host_name  BDX - MySQL - 1
  - address  192.168.1.32
  - description  Serveur du cluster MySQL du datacenter de Bordeaux
  - enabled  1
- Discovery
  - host_name  BDX - MySQL - 1
  - address  192.168.1.32
  - description  shinken.local - debian 13
  - use  linux

Le résultat de leur mélange sera le suivant :




Lorsque la même propriété est définie dans plusieurs définitions, la valeur de l'hôte prise en compte sera celle dont la source a le plus petit numéro d'ordre.

Dans l'exemple, on voit que la propriété commune ( *description) a* pour valeur la première valeur définie ( *"Serveur du cluster MySQL du datacenter de Bordeaux"* ), car la source "*Ma source d'import par fichier*" a un ordre plus petit que la source "*Discovery*".

# Les clés de synchronisation

## Définition

La liste des clés de synchronisation permet de déterminer quels éléments, provenant de sources différentes, vont être mélangés ensemble.

Une clé de synchronisation peut être la valeur de n'importe quelle :

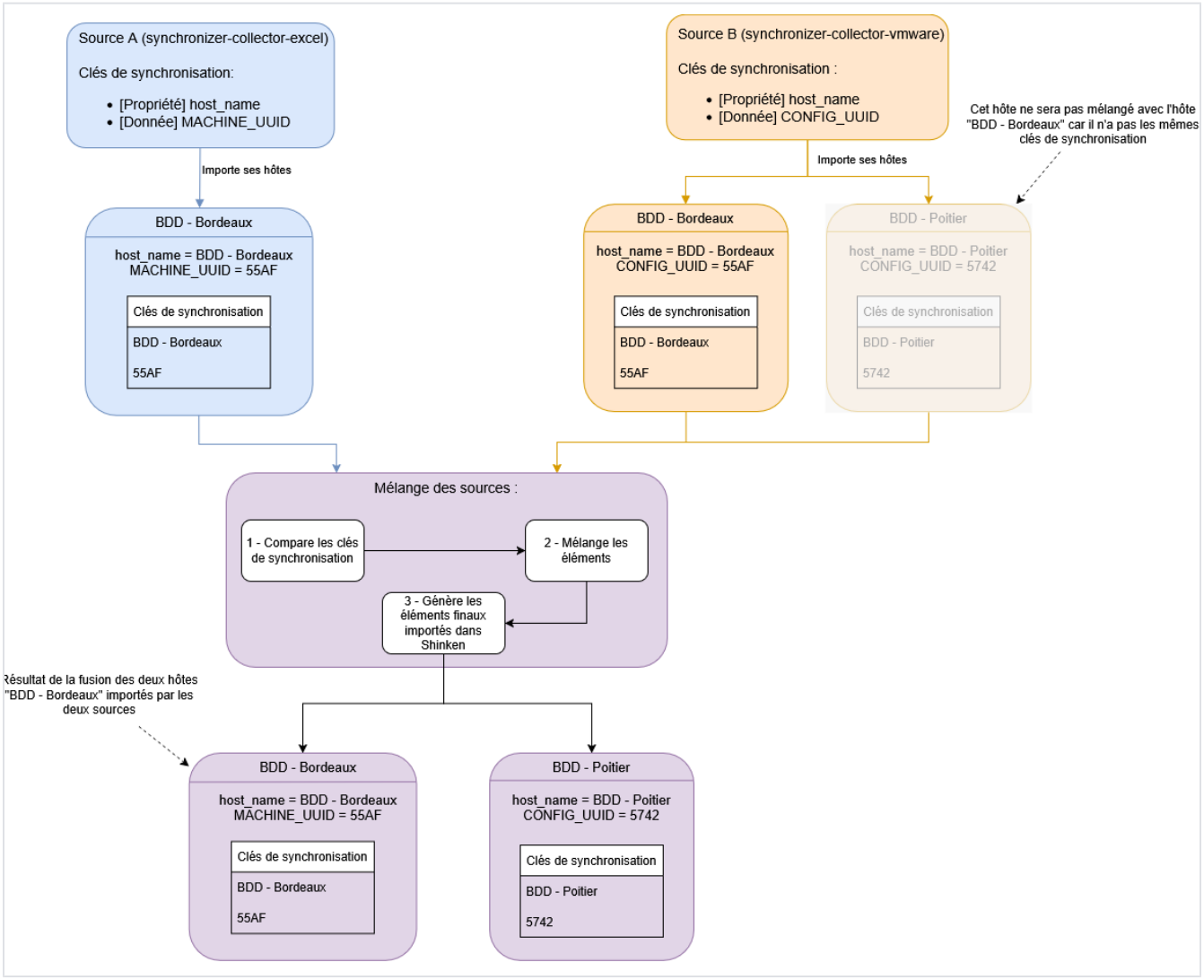
- propriété,
- ou donnée.

Le nom de l'élément et son _SE_UUID seront toujours utilisés comme clé de synchronisation si la source est capable de fournir ces propriétés.

> La source Discovery n'est pas capable de fournir un _SE_UUID. ( *voir la page* Collecteur de type ( discovery-import ) - Import depuis un scan réseau )

> Il est conseillé de ne pas importer les _SE_UUID d'un autre Synchronizer avec la source synchronizer-collector-linker ( *voir la page* Collecteur de type ( synchronizer-collector-linker ) - Import depuis un autre Synchronizer ).
>
> Cela pourrait créer des conflits avec les éléments déjà importés.

### Exemple de fonctionnement des clés de synchronisation




## Sources permettant de choisir ses clés de synchronisation

Il est possible de choisir les clés de synchronisation pour les sources de type suivant :

- Synchronizer-collector-excel ( *voir la page* Voir la configuration du module ( synchronizer-collector-excel ) )
- Synchronizer-collector-vmware ( *voir la page* Voir la configuration du module ( synchronizer-collector-vmware ) )

- ldap-import ( *voir la page* Voir la configuration du module ( ldap-import : mode openldap ) )
- cfg-file-import ( *uniquement pour les hôtes, voir la page Voir la configuration du module ( cfg-file-import )* )

## Comment choisir ses clés de synchronisation

Quand la source permet de choisir ses clés de synchronisation, une clé de synchronisation peut être la valeur de n'importe quelle :

- propriété,
- ou donnée.

Il est conseillé d'utiliser des clés permettant de différencier les éléments : il faut éviter les clés qui ne retournent que peu de valeurs différentes ( *comme les propriétés n'ayant pour valeurs possibles que "Vrai" ou "Faux"* )

Par exemple :

- La valeur de la propriété " enabled " ( *"Activé" dans l'Interface de Configuration* ) est une **mauvaise** clé de synchronisation.
- La valeur de la propriété "address" ( *"Addresse" dans l'Interface de Configuration* ) est une **bonne** clé de synchronisation pour un Hôte, car il est cohérent que deux hôtes ayant la même adresse ( *IP ou DNS* ) soient le même élément.

Pour trouver la clé d'import d'une propriété, il est possible de se rendre dans l'interface de configuration, de cliquer sur le bouton d'aide d'une propriété **( 1 )** :

Il suffit de copier la clé depuis l'interface **( 2 )** :

## Cas 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-hosts-60f9373a47d511e8a3c1080027a3ae8c
      _SE_UUID_HASH        7883f1c2c623f7001d70c02503ce198d
   # 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-hosts-6454ba8a47d511e897f8080027a3ae8c
      _SE_UUID_HASH        7b991a6b08e648842e1a7c7bc28ee140
   # 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.

La data _SE_UUID_HASH permet de simplement vérifier que la data _SE_UUID n'a été générée correctement et n'a pas été corrompue par une typo d'un utilisateur en éditant son fichier.

À noter : elle n'a qu'une valeur d'information pour l'auteur du fichier, le Synchronizer ne refusera pas l'objet si la valeur est vide ou incorrecte.

Un couple _SE_UUID / _SE_UUID_HASH à la forme suivante :

_SE_UUID core- **type_element** - **identifiant_uuid1_unique**

_SE_UUID_HASH **hash_md5_du_se_uuid**

- **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 :
    - clusters
    - clustertpls
    - hosts
    - hosttpls
    - hostgroups
    - serviceshosts
    - serviceshosttpls
    - servicesclusters
    - servicesclustertpls
    - servicetpls
    - contacts
    - contacttpls
    - contactgroups
    - escalations
    - notificationways
    - commands
    - businessimpactmodulations
    - macromodulations
    - resultmodulations
    - timeperiods
- **identifiant_uuid1_unique** : Chaine de caractères ( *alphanumériques* ) au format **UUID1** permettant d'identifier de manière unique l'élément.
- **hash_md5_du_se_uuid** : Hash MD5 de la valeur du se_uuid
  - par exemple:
    - pour un hôte, en ayant généré un uuid1="60f9373a47d511e8a3c1080027a3ae8c"
      - on calcule donc se_uuid_hash = md5("core-hosts-60f9373a47d511e8a3c1080027a3ae8c") => "7883f1c2c623f7001d70c02503ce198d"

```
_SE_UUID            core-hosts-60f9373a47d511e8a3c1080027a3ae8c
_SE_UUID_HASH       7883f1c2c623f7001d70c02503ce198d
```

### Exemple de génération du couple _SE_UUID / _SE_UUID_HASH par code

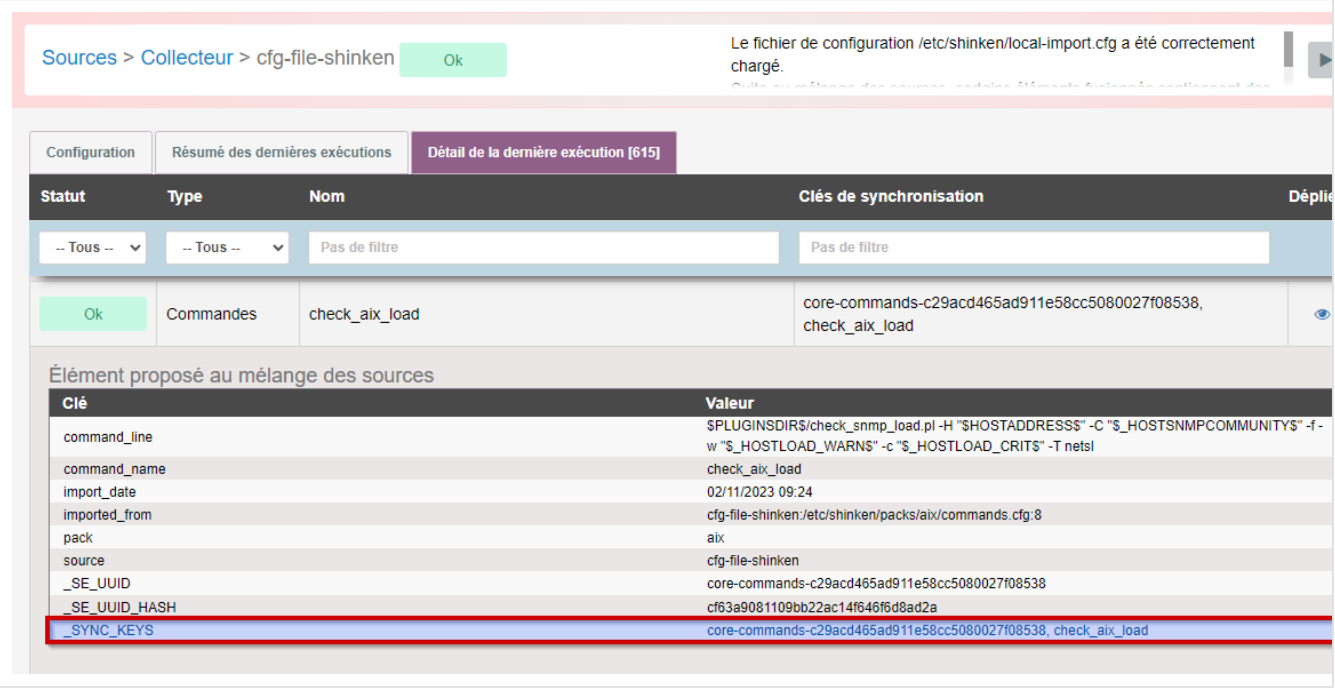
L'**UUID1** et **MD5** sont normés, et vont donc donner des résultats similaires suivants les langages utilisés. Par exemple en python, on peut les générer de cette manière :

**Exemple de génération des UUID1 et MD5 en python pour un hôte**

```
import uuid
import hashlib

# uuid1 generation as string
s_uuid = uuid.uuid1().hex

# generate se_uuid as string
se_uuid = u'core-hosts-%s' % s_uuid

# generate the se_uuid_hash as string, based on the se_uuid string
se_uuid_hash = hashlib.md5(se_uuid).hexdigest()
```

## Voir les clés de synchronisation

Les clés de synchronisation sont visibles depuis l'onglet "Détail de la dernière exécution" dans les pages de gestions de la source.




# Modification du 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.

- La valeur par défaut est : **all**
- Les valeurs possibles sont :

| Nom des valeurs dans l'interface | Noms des valeurs dans le fichier | Description |
|---|---|---|
| Ne fusionne pas avec les autres | `dont_merge_with_other` | Les éléments récoltés par cette source seront importés, mais pas fusionnés avec les autres sources.<br><br>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 provoqueront un conflit.<br><br>ⓘ À utiliser si la source donne un élément complet et qui ne doit pas être modifié ni par l'interface de configuration ni dpar d'autres sources |
| Seulement avec "Syncui" | `only_with_syncui` | Si un élément importé par cette source est aussi récolté par "Syncui", ils seront fusionnés pour ne donner qu'un élément.<br><br>ⓘ À utiliser si la source donne un élément complet et qu'on veut le modifier via l'interface de configuration. |
| Tout sauf Syncui | `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.<br><br>ⓘ À utiliser si la source donne un élément à compléter avec d'autres sources, mais qui ne doit pas être modifié par l'interface de configuration. |
| Toutes | `all` | Si un même élément est importé par plusieurs sources, alors les éléments seront fusionnés pour n'en donner qu'un.<br><br>ⓘ Cas par défaut : les éléments de cette source seront complétés avec les autres sources et l'interface de configuration. |

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

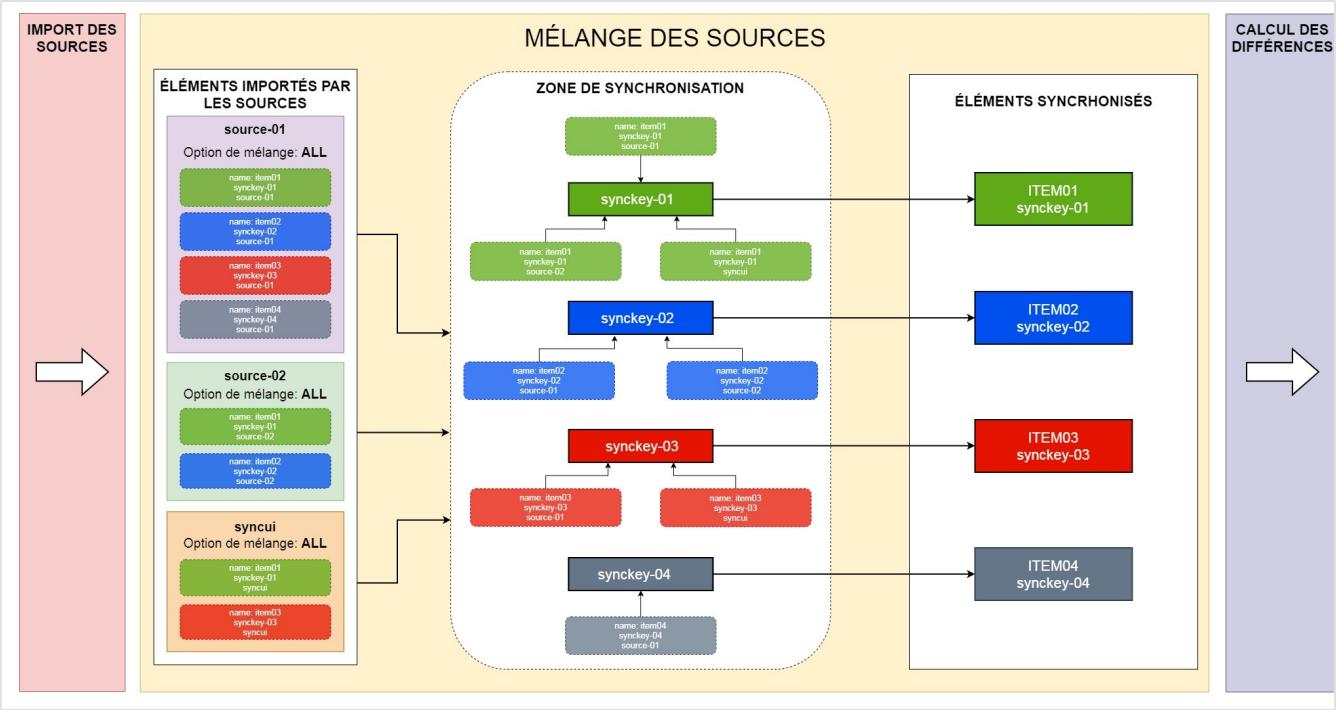
Chaque fois qu'une source est importée, elle l'est aussi. Si une source ne se mélange pas avec Syncui, les modifications faites dans l'interface de configuration sur les propriétés qui n'ont pas été importées de cette source apparaîtront en différence.

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

## Mode de mélange : "Toutes"

Les éléments de la source avec le mode de mélange "Toutes" ( *all* ) seront fusionnées avec les éléments trouvés lors du dernier import des autres sources. Y compris Syncui, soit les dernières modifications de l'interface de configuration.

Un exemple de mélange de sources avec l'option "Toutes".

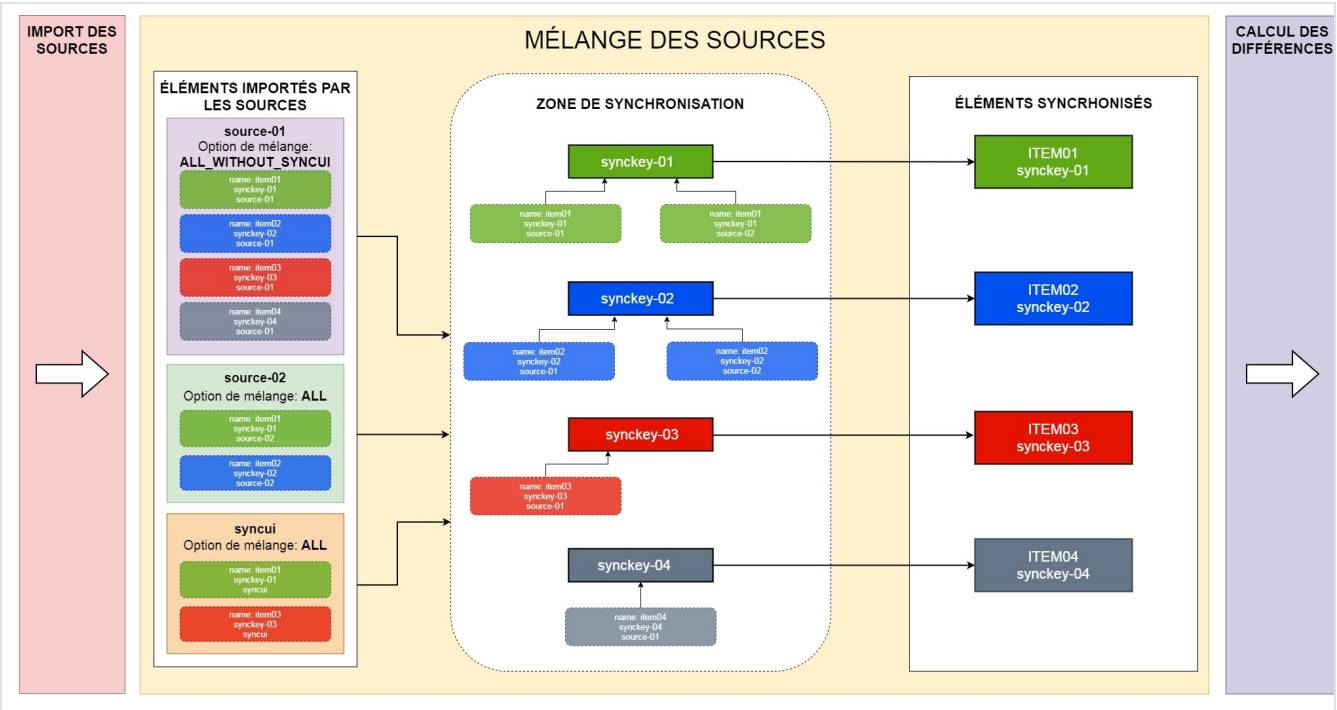



## Mode de mélange : "Tout sauf Syncui"

Les éléments de la source avec le mode de mélange "Tout sauf Syncui" ( *all_without_syncui* ) seront fusionnés avec les éléments des autres sources sauf avec la source Syncui.

- Les modifications faites dans l'interface de configuration sur les propriétés qui n'ont pas été importées de cette source apparaîtront en différence.

Un exemple de mélange de sources avec l'option "Tout sauf Syncui"



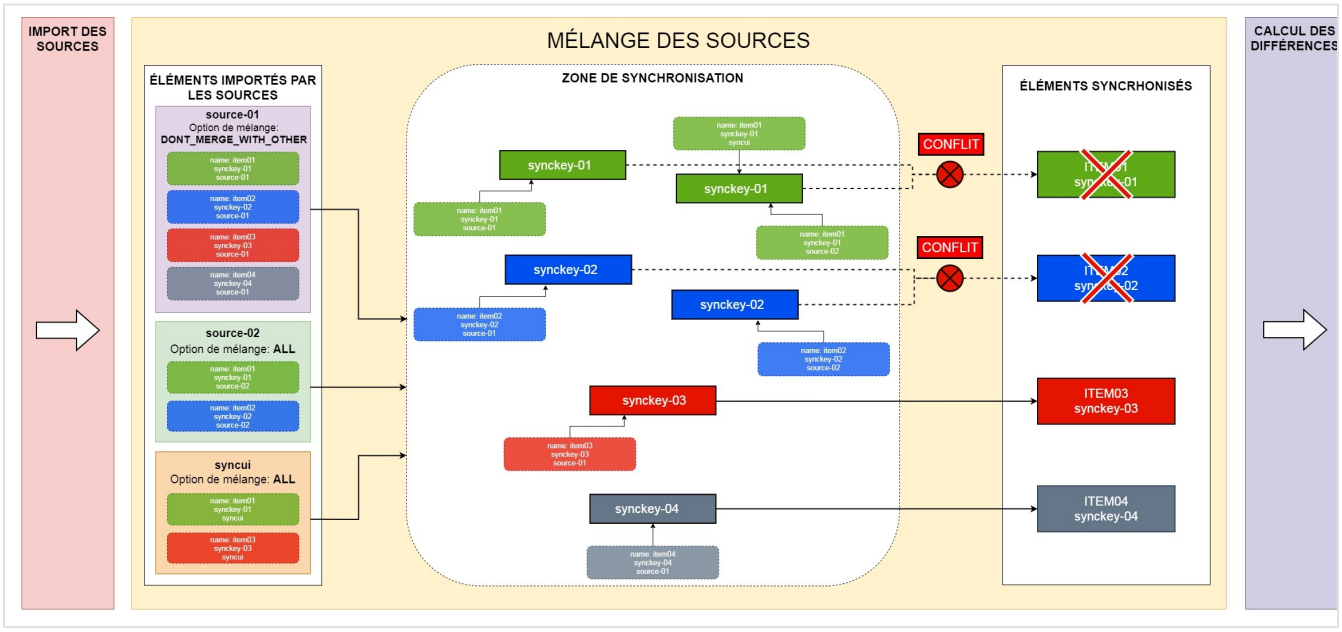

Quand la source Syncui provoque un conflit, elle n'est pas mélangée avec les autres sources.

## Mode de mélange : "Ne fusionne pas avec les autres"

Les éléments de la source avec le mode de mélange "Ne fusionne pas avec les autres" ( *dont_merge_with_other* ) ne seront pas fusionnés avec les éléments des autres sources y compris la source Syncui.

- Les modifications faites dans l'interface de configuration sur les propriétés qui n'ont pas été importées de cette source apparaîtront en différence.
- Et les éléments des autres sources ne pourront pas fusionner avec les éléments de cette source. 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 pas être importés et provoqueront un conflit.

Un exemple de mélange de sources avec l'option "Ne fusionne pas avec les autres".




## Mode de mélange : "Seulement avec "Syncui"

Les éléments de la source avec le mode de mélange "Seulement avec "Syncui" ( *only_with_syncui* ) seront fusionnés uniquement avec la source Syncui.

- Les éléments des autres sources ne pourront pas fusionner avec les éléments de cette source. 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 provoqueront un conflit.

Un exemple de mélange de sources avec l'option "Seulement avec "Syncui".