

# Widget Graphique

## Sommaire

[Description](#)  
[Utilisation](#)  
[Configuration](#)  
[Affichage des points isolés](#)

## Présentation

Le module architecture-export effectue trois étapes types d'actions distinctes :

- Modification de la configuration de NagVis afin de permettre la communication entre NagVis et Shinken
- Envoi de l'architecture aux différents récepteurs ( autres installation de Shinken )
- Création des cartes d'une architecture Shinken pour NagVis

Chaque action est retranscrite dans les logs de l'Arbiter.

## Les logs de modification de la configuration de NagVis



Tous les logs de modification/lecture sont écrits deux fois. Ceci est dû au fait que nous fournissons deux installations NagVis distinctes, chacune d'entre elles est concernée par ces modifications/lectures :

nagvis-shinken-architecture, qui est celui sur lequel nous mettons à disposition des cartes générées automatiquement

nagvis, qui lui ne contient aucune carte générée par Shinken

## Lecture de la configuration du module [ READING CONFIGURATION ]

Au démarrage du module, celui-ci va commencer par lire sa propre configuration afin de vérifier que les informations sont bonnes (voir [HIDDEN - 001 - Module architecture-export](#))

Il est ainsi possible de voir toutes les étapes de la lecture de la configuration.

### Exemple

Voici un exemple d'une lecture de configuration du module ayant réussie :

```
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ READING CONFIGURATION ] Broker  
[ broker ] found from parameter [ architecture_export__broker_connection__broker_name ]  
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ READING CONFIGURATION ] WebUI  
module [ WebUI ] found from parameter [ architecture_export__broker_connection__broker_webui ]  
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ READING CONFIGURATION ] The  
WebUI address is [ localhost ], we will use its public address [ http://192.168.1.119:7767 ]  
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ READING CONFIGURATION ]  
Livestatus module [ Livestatus ] found from parameter [  
architecture_export__broker_connection__broker_livestatus ]  
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ READING CONFIGURATION ] The  
Livestatus address is [ localhost ], we will use its public address [ 192.168.1.119:50000 ]
```

### Cas d'erreurs

Si le Broker renseigné dans le paramètre **architecture\_export\_\_broker\_connection\_\_broker\_name** est introuvable, l'édition du fichier de configuration de NagVis sera annulée et ce message apparaîtra :

```
[AAAA-MM-DD HH:MM:SS] ERROR : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ READING CONFIGURATION ] The  
Broker [ broker ] set with parameter [ architecture_export__broker_connection__broker_name ] do not exists.  
Abort edition of the nagvis config file
```

Si le paramètre **architecture\_export\_\_broker\_connection\_\_broker\_webui\_communication\_type** est à "module" et que le module WebUI renseigné dans le paramètre **architecture\_export\_\_broker\_connection\_\_broker\_webui\_target** est introuvable, l'édition de la configuration de la communication entre NagVis et le module WebUI sera annulée et ce message apparaîtra :



Pour plus d'informations sur ces paramètres, voir [HIDDEN - 001 - Module architecture-export](#)

```
[AAAA-MM-DD HH:MM:SS] ERROR : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ READING CONFIGURATION ] The WebUI module [ WebUI ] set with parameter [ architecture_export__broker_connection__broker_webui_target ] do not exists on the Broker [ broker-master ], we will not change the configuration of the communication between NagVis and the WebUI
```

Si le module Livestatus renseigné dans le paramètre **architecture\_export\_\_broker\_connection\_\_broker\_livestatus** est introuvable, l'édition de la configuration de la communication entre NagVis et le module Livestatus sera annulée et ce message apparaîtra :

```
[AAAA-MM-DD HH:MM:SS] ERROR : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ READING CONFIGURATION ] The Livestatus module [ Livestatus ] set with parameter [ architecture_export__broker_connection__broker_livestatus ] do not exists on the Broker [ broker-master ], we will not change the configuration of the communication between NagVis and the Livestatus
```

## Edition/Lecture de la configuration de NagVis [ CONFIGURATION ]

Après la lecture de la configuration du module, celui-ci va être face à deux options :

- Si la configuration est bonne, la configuration de NagVis sera systématiquement mise à jour.
- Sinon, il va simplement lire la configuration de NagVis afin de connaître son état actuel

### Exemples

Voici un exemple d'une modification de la configuration de NagVis ayant réussie :

```

[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ] UPDATING connection between NagVis and Shinken modules
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ] Editing /etc/shinken/external/nagvis/etc/nagvis.ini.php file
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ] - Setting WebUI connection to allow redirection on NagVis maps
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ] => "shinken_auth_protocol" parameter set to [ http ]
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ] => "shinken_auth_port" parameter set to [ 7767 ]
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ] => "shinken_auth_address" parameter set to [ 192.168.1.119 ]
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ] => "hosturl" parameter set to [ http://192.168.1.119:7767/detail-by-name/
[host_name] ]
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ] => "serviceurl" parameter set to [ http://192.168.1.119:7767/detail-by-name/
[host_name]/checks/[service_description] ]
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ] - Setting Livestatus connection to get Shinken objects status
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ] => "socket" parameter set to [ socket="tcp:192.168.1.119:50000" ]
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ] File /etc/shinken/external/nagvis/etc/nagvis.ini.php edited
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
UPDATING connection between NagVis and Shinken modules
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
Edit /opt/nagvis/etc/nagvis.ini.php file
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
- Setting WebUI connection to allow redirection on NagVis maps
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
] => "shinken_auth_protocol" parameter set to [ http ]
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
] => "shinken_auth_port" parameter set to [ 7767 ]
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
] => "shinken_auth_address" parameter set to [ 192.168.1.119 ]
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
] => "hosturl" parameter set to [ http://192.168.1.119:7767/detail-by-name/[host_name] ]
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
] => "serviceurl" parameter set to [ http://192.168.1.119:7767/detail-by-name/[host_name]/checks/
[service_description] ]
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
- Setting Livestatus connection to get Shinken objects status
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
] => "socket" parameter set to [ socket="tcp:192.168.1.119:50000" ]
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
File /opt/nagvis/etc/nagvis.ini.php edited

```

Si une erreur de configuration est présente, l'architecture-export lira simplement la configuration de NagVis afin d'afficher son état actuel :

```

[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ CONFIGURATION ]
=====
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ CONFIGURATION ] DUE TO
CONFIGURATION ERROR, The connection between NagVis and Shinken modules will NOT be updated
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ CONFIGURATION ]
=====
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ] Reading /etc/shinken/external/nagvis/etc/nagvis.ini.php file
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ]      - No WebUI redirection configuration found, we will keep previous NagVis configuration.
Redirection to WebUI on NagVis maps may not work properly
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ]      => "shinken_auth_protocol" parameter not changed. It is still [ http ]
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ]      => "shinken_auth_port" parameter not changed. It is still [ 7767 ]
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ]      => "shinken_auth_address" parameter not changed. It is still [ 192.168.1.119 ]
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ]      => "hosturl" parameter not changed. It is still [ http://192.168.1.119:7767/detail-
by-name/[host_name] ]
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ]      => "serviceurl" parameter not changed. It is still [ http://192.168.1.119:7767
/detail-by-name/[host_name]/checks/[service_description] ]
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ]      - No Livestatus module found, we will keep previous NagVis configuration. Status of
Shinken elements on NagVis maps may not work properly
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis-shinken-architecture ]
[ CONFIGURATION ]      => "socket" parameter not changed. It is still [ socket="tcp:192.168.1.119:50000" ]
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ] The
connection between NagVis and Shinken modules will not be changed
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
Read /opt/nagvis/etc/nagvis.ini.php file
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
- No WebUI redirection configuration found, we will keep previous NagVis configuration. Redirection to WebUI
on NagVis maps may not work properly
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
]      => "shinken_auth_protocol" parameter not changed. It is still [ http ]
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
]      => "shinken_auth_port" parameter not changed. It is still [ 7767 ]
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
]      => "shinken_auth_address" parameter not changed. It is still [ 192.168.1.119 ]
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
]      => "hosturl" parameter not changed. It is still [ http://192.168.1.119:7767/detail-by-name/
[host_name] ]
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
]      => "serviceurl" parameter not changed. It is still [ http://192.168.1.119:7767/detail-by-name/
[host_name]/checks/[service_description] ]
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
- No Livestatus module found, we will keep previous NagVis configuration. Status of Shinken elements on
NagVis maps may not work properly
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ nagvis ] [ CONFIGURATION ]
]      => "socket" parameter not changed. It is still [ socket="tcp:192.168.1.119:50000" ]
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ CONFIGURATION ]
=====

```

## Les logs d'envoi/réception d'architecture [ NOM\_DE\_L'ARCHITECTURE ]

Un module architecture-export peut envoyer et recevoir une architecture.  
Chaque architecture reçue sera cartographiée sur le module l'ayant reçu.

### Envoi de son architecture

Au démarrage, l'architecture-export va envoyer son architecture à tous les "recipients" configurés dans le module.

Les logs montrent alors tous les envois effectués.

### Exemple

```
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ ARCHITECTURE_NAME ] Sending architecture to 2 recipient(s)
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ ARCHITECTURE_NAME ] Sending architecture to [ http://127.0.0.1:7780 ]
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ ARCHITECTURE_NAME ] Sending architecture to [ http://192.168.0.15:7780 ]
```

## Cas d'erreurs

Si une adresse ne commence pas par 'http' ou 'https' cette erreur apparaîtra et l'envoi de l'architecture à cette adresse sera annulée :

```
[AAAA-MM-DD HH:MM:SS] ERROR : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ Shinken ] External export module address "127.0.0.1" has an incorrect format ( does not start with "http://" or "https://" )
```

Si le port d'une adresse n'a pas été spécifié, cette erreur apparaîtra et l'envoi de l'architecture à cette adresse sera annulée:

```
[AAAA-MM-DD HH:MM:SS] ERROR : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ Shinken ] No port specified in the external export module address "127.0.0.1"
```

## Réception d'une architecture

Lors de la réception d'une architecture, nous la sauvegardons temporairement dans un fichier json afin de préserver son état. Ceci permet au redémarrage de l'Arbiter de connaître les architectures présentes sur son module.

### Exemple

```
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ ARCHITECTURE_NAME ] [ 127.0.0.1 ] ask me to map its architecture
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ ARCHITECTURE_NAME ] The Architecture have been received, we will make its maps
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ ARCHITECTURE_NAME ] Save broks contents /var/lib/shinken/architecture_export_received.json
```

## Les logs de création de carte NagVis [ NOM\_DE\_L'ARCHITECTURE ] [ MAP GENERATOR ]

### Récupération des hôtes existants sur le listener-shinken

Avant tout, il nous faut récupérer les hôtes existant sur le listener-shinken étant lié à l'architecture que nous allons cartographier. Nous affichons alors des logs expliquant ce qui a été trouvé :

### Exemple

Voici un exemple de récupération d'hôtes auprès du listener-shinken

```
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP GENERATOR ] - Retrieving host(s) corresponding to this architecture from the Synchronizer ( via listener-shinken )
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP GENERATOR ] => Found [ Shinken - ALL - 192.168.1.119 ]
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP GENERATOR ] => Found [ Shinken - BORDEAUX - 192.168.1.119 ]
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP GENERATOR ] => Found [ Shinken - GRENOBLE - 192.168.1.119 ]
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP GENERATOR ] => Found [ Shinken - LYON - 192.168.1.119 ]
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP GENERATOR ] => Found 4 host(s)
```

## Cas d'erreurs

Si l'architecture-export essaie de communiquer avec le listener-shinken en HTTPS alors qu'il ne l'est pas, une erreur apparaîtra et la création de la map sera abandonnée. Pour régler ce problème, il faut modifier la configuration du module (voir [HIDDEN - 001 - Module architecture-export](#)).

```
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      - Retrieving host(s) corresponding to this architecture from the Synchronizer ( via listener-
shinken )
[AAAA-MM-DD HH:MM:SS] ERROR     : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => Tried to request the listener with HTTPS, but it failed. Verify if your listener-
shinken is on HTTPS
```

Si les identifiants de connexion au listener-shinken sont incorrects, une erreur apparaîtra et la création de la map sera abandonnée. Pour régler ce problème, il faut modifier la configuration du module (voir [HIDDEN - 001 - Module architecture-export](#)).

```
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      - Retrieving host(s) corresponding to this architecture from the Synchronizer ( via listener-
shinken )
[AAAA-MM-DD HH:MM:SS] ERROR     : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => Bad login and password for the listener
```

## Création de la liste des hôtes à partir de l'architecture reçue

Une fois les hôtes déjà disponible sur le listener-shinken récupérés, nous devons en modifier/créer pour être en accord avec l'architecture reçue.

### Exemple

Voici l'exemple d'une génération d'un hôte de l'architecture reçue :

```
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      - Building host(s) from architecture description received
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => Host [ Shinken - ALL - localhost ] modified
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => Host [ Shinken - Paris - localhost ] generated
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => 2 host(s) generated
```

## Envoi des hôtes au listener-shinken

Lorsque tous les hôtes liés à l'architecture sont générés, nous les envoyons au listener-shinken.

### Exemple

Voici un exemple de l'envoi d'un hôte vers le listener-shinken :

```
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      - Sending host(s) to Synchronizer ( via listener-shinken )
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => Sending [ Shinken - ALL - localhost ]
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => Sending [ Shinken - Paris - localhost ]
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => 2 host(s) sent
```

## Cas d'erreurs

Si l'architecture-export essaie de communiquer avec le listener-shinken en HTTPS alors qu'il ne l'est pas, une erreur apparaîtra et la création de la map sera abandonnée. Pour régler ce problème, il faut modifier la configuration du module (voir [HIDDEN - 001 - Module architecture-export](#)).

```
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      - Sending host(s) to Synchronizer ( via listener-shinken )
[AAAA-MM-DD HH:MM:SS] ERROR     : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => Tried to request the listener with HTTPS, but it failed. Verify if your listener-
shinken is on HTTPS
```

Si les identifiants de connexion au listener-shinken sont incorrect, une erreur apparaîtra et la création de la map sera abandonnée. Pour régler ce problème, il faut modifier la configuration du module (voir [HIDDEN - 001 - Module architecture-export](#)).

```
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      - Sending host(s) to Synchronizer ( via listener-shinken )
[AAAA-MM-DD HH:MM:SS] ERROR      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => Bad login and password for the listener
```

## Désactivation des hôtes obsolètes

Des hôtes, trouvés dans le listener-shinken, liés à l'architecture reçue peuvent être obsolètes suite à une modification de l'architecture. Nous les désactivons alors du listener-shinken.

### Exemple

Voici un exemple d'une désactivation d'hôtes obsolètes

```
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      - Disabling old host(s) on Synchronizer ( via listener-shinken )
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => Old host [ Shinken - BORDEAUX - 192.168.1.119 ] disabled on synchronizer
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => Old host [ Shinken - GRENOBLE - 192.168.1.119 ] disabled on synchronizer
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => Old host [ Shinken - LYON - 192.168.1.119 ] disabled on synchronizer
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => 3 host(s) disabled
```

## Suppression des images des cartes NagVis obsolètes

Avant de créer les nouvelles cartes, il faut supprimer les anciennes afin de ne pas avoir de doublons et de ne pas charger le disque.

Les cartes sont découpées en deux parties :

- Le fond, qui est une image générée par l'architecture-export
- Le fichier de configuration, qui est généré par l'architecture-export et équivaut aux positions des éléments Shinken dans la carte

Nous commençons par supprimer les fonds.

### Exemple

Voici un exemple d'une suppression de fonds de cartes NagVis

```
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      - Removing obsolete NagVis map background(s)
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => [ /etc/shinken/external/nagvis/share/userfiles/images/maps/background_[Shinken]
_Royaume:_All2021519148-elbca43a-cdac-4411-9691-870a625b911b.png ] removed
[AAAA-MM-DD HH:MM:SS] INFO      : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => [ /etc/shinken/external/nagvis/share/userfiles/images/maps/background_[Shinken]
_Architecture_Shinken2021519148-elbca43a-cdac-4411-9691-870a625b911b.png ] removed
```

## Suppression des fichiers de configuration des cartes NagVis obsolètes

Après avoir supprimé les fonds des cartes, il faut supprimer les fichiers de configuration.

### Exemple

Voici un exemple d'une suppression de fichiers de configuration NagVis

```
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ] - Removing obsolete NagVis map configuration(s)
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ] => [ /etc/shinken/external/nagvis/etc/maps/shinken_global-elbca43a-cdac-4411-9691-
870a625b911b.cfg ] removed
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ] => [ /etc/shinken/external/nagvis/etc/maps/shinken_architecture-elbca43a-cdac-4411-9691-
870a625b911b.cfg ] removed
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ] => [ /etc/shinken/external/nagvis/etc/maps/All-elbca43a-cdac-4411-9691-870a625b911b.cfg
] removed
```

## Calcul de la taille des royaumes pour les cartes NagVis

Les cartes NagVis sont sous forme de blocs imbriqués. Pour pouvoir les créer il faut alors assigner une taille de bloc à chaque royaume. Pour se faire, nous calculons leur taille selon leur contenu :

- Taille initiale du royaume
- Taille des sous-royaumes
- Taille des hôtes
  - Nombre de checks

### Exemple

Voici un exemple d'un calcul de la taille réussi

```
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ] - Computing architecture objects size
```

## Tri des royaumes par taille

nous trions les royaumes selon le paramètre `map_realm_layout` défini dans la configuration du module (voir [HIDDEN - 001 - Module architecture-export](#)).

### Exemples

Voici un exemple d'un tri par taille

```
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ] - Sorting realms by size
```

Et un exemple d'un tri par nom

```
[AAAA-MM-DD HH:MM:SS] INFO : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ] - Sorting realms by name
```

## Génération des cartes NagVis

C'est à cette étape que les cartes NagVis sont générées. Il y a au total 4 types de fichiers :

- Un arbre représentant tous les royaumes,
- Une carte détaillée de tous les royaumes,
- Une carte détaillée par royaume,
- Un fichier de configuration de rotation, permettant de faire défiler les cartes NagVis d'une même architecture,

Chaque création de fichier est affiché dans les logs.

### Exemple

Voici un exemple de génération ayant réussie

```
[AAAA-MM-DD HH:MM:SS] INFO    : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      - Generating Nagvis maps
[AAAA-MM-DD HH:MM:SS] INFO    : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => Generating realms tree NagVis map
[AAAA-MM-DD HH:MM:SS] INFO    : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => Generating detailed architecture NagVis map
[AAAA-MM-DD HH:MM:SS] INFO    : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => Generating NagVis map for realm [All]
[AAAA-MM-DD HH:MM:SS] INFO    : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => Generating NagVis map for realm [Paris]
[AAAA-MM-DD HH:MM:SS] INFO    : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => Generating rotation configuration for NagVis maps
```

## Sauvegarde des hôtes listener-shinken dans un fichier de rétention

A chaque réception d'architecture, le module sauvegarde l'état des hôtes du listener-shinken dans un fichier.

Ce permet de n'envoyer de demande de creation / modification d'hôte que si il y a des nouveautés.

### Exemple

Voici un exemple de sauvegarde de rétention ayant réussie

```
[AAAA-MM-DD HH:MM:SS] INFO    : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      - Saving host(s) sent to listener in retention file
[AAAA-MM-DD HH:MM:SS] INFO    : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => listener hosts mapping saved in /var/lib/shinken/architecture_export_hosts_mapping.
json
```

### Cas d'erreurs

Si l'architecture-export n'a pas les droits d'écriture sur le fichier de rétention, cette erreur apparaîtra et la sauvegarde sera abandonnée

```
[AAAA-MM-DD HH:MM:SS] INFO    : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      - Saving host(s) sent to listener in retention file
[AAAA-MM-DD HH:MM:SS] ERROR   : [ NOM_DE_L'ARBITER ] [ ARCHITECTURE-EXPORT ] [ NOM_DE_L'ARCHITECTURE ] [ MAP
GENERATOR ]      => Permission denied on /var/lib/shinken/architecture_export_hosts_mapping.json
```