

# Configuration avancée ( shinken.cfg )

## Sommaire

- Paramètres de données de Performance
  - Option de gestion des données de performance
- Paramètres avancés de planification
  - Hôte orphelin/Service Check Option
- Réglage des performances
  - Hôtes en cache/Service Check Horizon
- Paramètres de Flapping
  - Activer la détection
  - Seuil bas par défaut ( check / hôte / cluster )
  - Seuil haut par défaut ( check / hôte / cluster )
- Diverses commandes en timeout
- Rafraîchissement des checks
  - Option de rafraîchissement des Hôtes/Service
  - Intervalle de rafraîchissement
  - Option de notification
  - Option de commande externe
- Paramètres de planification
  - Option d'exécution d'un hôte/service
  - Option d'acceptation de check/service passifs
  - Option du gestionnaire d'événements
  - Option de log des notifications
  - Option de log du gestionnaire d'événements
  - Option de log de commandes externes
  - Option de log des checks passifs
  - Temps d'intervalle
- Autres
  - Caractères illégaux dans le nom d'objet
  - Temps d'attente minimum avant la création du statut données manquantes
    - Au démarrage
    - En temps normal

## Paramètres de données de Performance

### Option de gestion des données de performance

Format:

```
process_performance_data=<0/1>
```

Cette valeur détermine si Shinken Enterprise va gérer des données de performance .

- 0 = ne gère pas de données de performance
- 1 = gère les données de performance (défaut)

Exemple:

```
process_performance_data=1
```

## Paramètres avancés de planification

### Hôte orphelin/Service Check Option

Format:

```
check_for_orphaned_services=<0/1>  
check_for_orphaned_hosts=<0/1>
```

Cette option permet d'activer ou désactiver les checks sur les hôtes/checks orphelins. Les checks orphelins sont ceux lancés par le Poller mais qui n'ont jamais renvoyés de résultats depuis longtemps. Sachant qu'ils n'ont rien renvoyé, ils ne sont pas replanifiés dans la file d'événements.

Si cette option est activée, et Shinken Enterprise ne trouve aucun résultat pour un check en particulier, un message d'erreur sera loggué et le check sera replanifié. Si vous voyez des checks qui ne sont jamais replanifiés, activez cette option et regardez si vous voyez des messages d'erreur loggués à propos de services orphelins.

- 0 = ne pas vérifier les services orphelins
- 1 = vérifier les services orphelins (défaut)

Exemple:

```
check_for_orphaned_services=1
check_for_orphaned_hosts=1
```

## Réglage des performances

### Hôtes en cache/Service Check Horizon

Format:

```
cached_host_check_horizon=<seconds>
cached_service_check_horizon=<seconds>
```

Cette option détermine le temps maximum ( *en secondes* ) pendant lequel le dernier état d'un check est considéré comme son état courant. L'état géré en cache (venant d'hôtes/check ayant été exécutés plus récemment que le temps spécifié par cette valeur) peut grandement améliorer la performance des checks.

Si cette valeur est trop élevée, cela peut se traduire par un état (temporaire) inapproprié. Une valeur plus basse peut générer un pic de performance .

Utiliser la valeur 0 si vous souhaitez désactiver la mise en cache des hôtes/checks.

Exemple:

```
cached_host_check_horizon=0
cached_service_check_horizon=0
```

## Paramètres de Flapping

### Activer la détection

Format:

```
enable_flap_detection=<0/1>
```

Cette option détermine si Shinken Enterprise va tenter de détecter les hôtes et checks en état "flapping".

L'état "flapping" apparaît quand un hôte ou un check passent d'un état à un autre très fréquemment. Quand Shinken Enterprise détecte un hôte ou un check dans cet état, il supprimera temporairement les notifications jusqu'à la fin de l'état "flapping".

Vous trouverez plus d'information sur la détection du flapping dans le paragraphe [Elements en Flapping](#).

- 0 = ne pas activer la détection du flapping
- 1 = activer la détection du flapping (défaut)

Exemple:

```
enable_flap_detection=1
```

### Seuil bas par défaut ( check / hôte / cluster )

Format:

```
low_service_flap_threshold=<percent>
low_host_flap_threshold=<percent>
```

Cette option est utilisée pour définir la valeur par défaut du seuil bas de détection du flapping. Pour plus d'information, suivre ce lien [élément en flapping](#).

Exemple:

```
low_service_flap_threshold=25.0
low_host_flap_threshold=25.0
```

### Seuil haut par défaut ( check / hôte / cluster )

Format:

```
high_service_flap_threshold=<percent>
high_host_flap_threshold=<percent>
```

Cette option est utilisée pour définir la valeur par défaut du seuil haut de détection du flapping. Pour plus d'information, suivre ce lien [élément en flapping](#).

Exemple:

```
high_service_flap_threshold=50.0
high_host_flap_threshold=50.0
```

## Diverses commandes en timeout

Format:

```
event_handler_timeout=<seconds> # default: 30s  
notification_timeout=<seconds> # default: 30s
```

Il s'agit du nombre maximum en secondes pour lancer le [gestionnaire d'événements](#) et les notifications. Si une commande dépasse cette limite de temps, elle sera supprimée et une alerte sera remontée.

Il y a souvent une large confusion sur ce que fait réellement cette option. Elle a été conçue pour être utilisée en dernier recours comme un mécanisme de suppression d'une commande qui se comporte mal dans le temps. Elle devrait être définie à une valeur élevée (comme 60 secondes ou plus pour une notification), de façon à ce que chaque gestionnaire d'événements puisse terminer d'exécuter sa tâche dans les temps.

Exemple:

```
event_handler_timeout=60  
notification_timeout=60
```

## Rafraîchissement des checks

### Option de rafraîchissement des Hôtes/Service

Format:

```
check_service_freshness=<0/1>  
check_host_freshness=<0/1>
```

Cette option détermine si Shinken Enterprise va vérifier régulièrement le rafraîchissement des hôtes/services. Activer cette option est très utile pour permettre de vérifier que les checks passifs se déroulent bien dans le temps ( [voir la page Mode actif et mode passif](#) ).

- 0 = ne pas vérifier le rafraîchissement des hôtes/services
- 1 = vérifier le rafraîchissement des hôtes/services (défaut)

Exemple:

```
check_service_freshness=0  
check_host_freshness=0
```

### Intervalle de rafraîchissement

Format:

```
service_freshness_check_interval=<seconds>  
host_freshness_check_interval=<seconds>
```

Ce paramètre définit la fréquence (en secondes) à laquelle Shinken Enterprise va vérifier le rafraîchissement d'un hôte/service ( [voir la page Mode actif et mode passif](#) ).

Exemple:

```
service_freshness_check_interval=60  
host_freshness_check_interval=60
```

### Option de notification

Format:

```
enable_notifications=<0/1>
```

Cette option détermine si Shinken Enterprise va envoyer des notifications ( [voir la page Logique de notification](#) ). Si cette option est désactivée, Shinken Enterprise n'enverra aucune notification pour aucun des hôtes/services.

Les valeurs sont les suivantes :

- 0 = Désactive les notifications
- 1 = Active les notifications (défaut)

Exemple:

```
enable_notifications=1
```

### Option de commande externe

Format:

```
check_external_commands=<0/1>
```

Cette option détermine si Shinken Enterprise va vérifier le fichier de commande externe pour les commandes devant être exécutées depuis le démon Arbiter.

- 0 = ne pas vérifier les commandes externes
- 1 = vérifier les commandes externes (défaut)

Exemple:

```
check_external_commands=1
```

## Paramètres de planification

### Option d'exécution d'un hôte/service

Format:

```
execute_service_checks=<0/1>  
execute_host_checks=<0/1>
```

Cette option détermine si Shinken Enterprise va exécuter les checks.

- 0 = ne pas exécuter les checks
- 1 = exécuter les checks (défaut)

Exemple:

```
execute_service_checks=1  
execute_host_checks=1
```

### Option d'acceptation de check/service passifs

Format:

```
accept_passive_service_checks=<0/1>  
accept_passive_host_checks=<0/1>
```

Cette option détermine si Shinken Enterprise acceptera les hôtes/checks passifs. Si cette option est désactivée, Shinken Enterprise n'acceptera aucun check passif.

- 0 = N'accepte pas les checks passifs
- 1 = Accepte les checks passifs

Exemple:

```
accept_passive_service_checks=1  
accept_passive_host_checks=1
```

### Option du gestionnaire d'événements

Format:

```
enable_event_handlers=<0/1>
```

Cette option détermine si Shinken Enterprise va lancer le gestionnaire d'événements.

- 0 = Désactive le gestionnaire d'événement
- 1 = Active le gestionnaire d'événement (défaut)

Exemple:

```
enable_event_handlers=1
```

### Option de log des notifications

Format:

```
log_notifications=<0/1>
```

Cette variable détermine si les messages de notification sont remontés. Utilisez cette option pour garder une trace des notifications de vos contacts.

- 0 = ne pas remonter de notifications

- 1 = remonté des notifications

Exemple:

***log\_notifications=1***

## Option de log du gestionnaire d'événements

Format:

***log\_event\_handlers=<0/1>***

Cette variable détermine si vous allez remonté les événements. Le gestionnaire d'événements est une commande optionnelle qui peut être lancée dès que l'état d'un hôte ou d'un service change. Il peut être très utile de les logger lorsque vous êtes en mode debug, ou lorsque vous essayez pour la première fois les scripts du gestionnaire d'événements.

- 0 = ne log pas d'événements
- 1 = log des événements

Exemple:

***log\_event\_handlers=1***

## Option de log de commandes externes

Format:

***log\_external\_commands=<0/1>***

Cette variable détermine si Shinken Enterprise va remonter les commandes externes reçues .

- 0 =ne logue pas les commandes externes
- 1 = loggue les commandes externes (défaut)

Exemple:

***log\_external\_commands=1***

## Option de log des checks passifs

Format:

***log\_passive\_checks=<0/1>***

Cette variable détermine si Shinken Enterprise va logger les checks passifs reçus des commandes externes.

- 0 = Don't log passive checks
- 1 = Log passive checks (default)

Exemple:

***log\_passive\_checks=1***

## Temps d'intervalle

Format:

***interval\_length=<seconds>***

Il s'agit du nombre de secondes par "unité d'intervalle" utilisé pour calculer le temps dans la liste de replanification, renotifications, etc. " les unités d'intervalle" sont utilisées dans le fichier de configuration de l'objet pour déterminer à quelle fréquence lancer un service, renotifier un contact, etc...

La valeur par défaut est à 60 secondes, ce qui signifie qu'une "unité d'intervalle" définie à 1 correspond à 60 secondes.

Exemple:

***interval\_length=60***

## Autres

### Caractères illégaux dans le nom d'objet

Format:

***illegal\_object\_name\_chars=<chars...>***

Cette option autorise l'utilisation de caractères spéciaux qui ne sont pas autorisés dans le nom des hôtes, services.... Shinken Enterprise peut autoriser tout type de caractères, mais nous recommandons de ne pas utiliser ceux spécifiés plus haut dans l'exemple.

Exemple:

```
illegal_object_name_chars=^-!$%^&*"'|'<>?,()=
```

## Temps d'attente minimum avant la création du statut données manquantes

### Au démarrage

Nom	Type	Unité	Défaut	Commentaire
<code>minimal_time_before_an_element_become_missing_data_at_startup</code>	entier positif	seconde	600	<p>Cette option est utile principalement pour les configurations imposantes ou pour les infrastructures souffrant d'une forte latence réseau.</p> <ul style="list-style-type: none"><li>En effet, si le chargement d'une configuration prend trop de temps, le statut de certains éléments pourrait être considéré comme manquant, alors que Shinken n'a pas fini de transmettre la configuration au Broker</li><li>La valeur minimale ne devrait pas être inférieure à celle de <b><code>minimal_time_before_an_element_become_missing_data</code></b></li></ul>

### En temps normal

Nom	Type	Unité	Défaut	Commentaire
<code>minimal_time_before_an_element_become_missing_data</code>	entier positif	seconde	60	<p>Cette option est utile principalement pour les configurations imposantes ou pour les infrastructures souffrant d'une forte latence réseau.</p> <ul style="list-style-type: none"><li>En effet, si l'intervalle entre les vérifications est court, et si les échanges entre les Schedulers, les Pollers et les Brokers prennent trop de temps, le statut de certains éléments pourrait être considéré comme manquant alors qu'il est en cours d'acheminement. Les checks seront alors dans le statut "<i>Données manquantes</i>"</li><li>Avec cette option, le mécanisme de détection des données manquantes ne commencera qu'après le délai défini dans cette option, laissant le temps aux résultats de check d'arriver jusqu'au Broker.</li></ul>