



## Onglet générique

- **Check Name**
  - Cette propriété est utilisée pour définir le nom du check. Les caractères suivants sont interdits : ~!%^&\*"'|<>?,()=+  
2 checks associés sur le même hôte ne peuvent pas avoir le même nom.
- **Check Templates to inherit**
  - Les modèles de templates qui sont attachés à l'hôte pour hériter de leurs propriétés et leurs données.
    - L'ordre des templates est important.
    - Une valeur définie dans 2 modèles vaudra la valeur du modèle le plus proche du début de la liste des modèles.
    - Si vous définissez une valeur pour une propriété ou une donnée, elle sera prioritaire sur celle héritée de modèles.
- **Apply on Host Groups**
  - Cette propriété est utilisée pour attacher ce check sur un ou plusieurs groupes d'hôtes.  
Utiliser les virgules pour spécifier plusieurs hôtes.
- **Apply on Hosts**
  - Cette propriété est utilisée pour attacher ce check sur un ou plusieurs hôtes.  
Utiliser le menu déroulant pour spécifier un ou plusieurs hôtes.
- **External URL**
  - Cette propriété est utilisée pour définir une URL qui peut être utilisée pour fournir plus d'information sur cet hôte.  
N'importe quelle URL valide peut être mise, commençant par `_http://_`  
Cette URL sera disponible dans l'interface de visualisation dans le volet détail présentant l'hôte.  
Cela peut être pratique si vous désirez faire une description écrite détaillée de l'hôte, une procédure de reprise sur panne, ... qui sera visible pour les autres membres de l'équipe.
- **Enabled**
  - Permettre que cet élément soit pris en compte lors du passage de Staging en Production.
- **Pack**
  - Cette propriété permet de définir dans quel pack se trouve le check.

## Onglet données

Cet onglet définit des données qui pourront être utilisées par des [Macros](#), notamment à l'utilisation de la [Commande](#) de check. Consulter ces pages pour plus d'information.

## Onglet monitoring

- **Planned DOWNTIME period**
  - Cette propriété permet de définir une période de temps durant laquelle l'hôte sera en maintenance, et en sortira dès que la période de temps sera fini.  
En maintenance, les checks de l'hôte seront toujours exécutés (donc il n'y aura pas de manques dans les métriques ou l'historique de l'hôte), mais les notifications et les commandes du gestionnaire d'évènement ne seront pas lancées.
- **Monitoring Period**
  - Cette propriété permet de choisir la période de supervision où sont faites les vérifications de l'état de l'hôte par Shinken.
- **Check Command Name**
  - Cette propriété permet de spécifier la commande qui est employée pour vérifier si le check est dans un statut en OK ou CRITIQUE.  
Si vous laissez vide cette propriété, la commande de contrôle par défaut sera employée.
- **Max Check Attempts**
  - Cette propriété permet de définir le nombre de tentative(s) où Shinken recommencera la commande de vérification d'un check si son résultat est autre chose que le statut OK  
Mettre cette valeur à 1 impliquera que Shinken générera une alerte dès qu'il n'est plus en état OK, sans réessayer  
Si vous ne désirez pas que Shinken fasse de vérification de statut de ce check, laissez cette propriété à 1, mais la commande de vérification doit être vide
- **Check Interval**
  - Cette propriété permet de définir le temps entre 2 vérifications de l'état du check.
- **Retry Interval**
  - Cette propriété permet de définir le temps d'attente avant de relancer la commande de vérification du check après un premier état différent de OK.  
Ce nombre est en minutes.  
Une fois que le nombre de relances de commandes de vérifications a atteint la valeur `max_check_attempts`, le dernier état récupéré est validé et la cadence des vérifications redevient à la valeur normale, définie dans la propriété **Intervalle de Check**
- **Check Running Timeout**
  - Cette propriété permet de définir la valeur en secondes du temps maximum permis pour le lancement d'une commande de supervision. Passé ce délai, la commande sera terminée. Si le check ou la commande définit également ce paramètre, leur valeur sera prioritaire par rapport à celle de l'hôte.  
Aucune valeur ou une valeur a -1, indique qu'il n'y a pas de Temps maximum défini.  
Si ni l'hôte, ni le check, ni la commande ne définissent ce paramètre, alors la valeur définie globalement pour Shinken (via le fichier `shinken.cfg` et le paramètre `check_running_timeout`) sera utilisée à la place.
- **Active checks enabled**

- Cette propriété permet de définir si Shinken ordonnance et lances la commande de vérification pour déterminer l'état de ce check. Ses valeurs peuvent être:
  - True: Shinken va ordonner et lancer la commande de vérification
  - False: Shinken ne va pas ordonner et lancer la commande de vérification
- **Passive checks enabled**
  - Cette propriété permet de définir si Shinken accepte les résultats de commandes de checks depuis des outils externes pour cet hôte. Les résultats externes seront reçu via le daemon Receiver de Shinken. Les valeurs possibles sont:
    - True: accepte les résultats de commandes de checks externes
    - False: n'accepte pas ces résultats
- **Duplicate for each**
  - Cette propriété est utilisé pour générer plusieurs checks dans l'arbitre avec une seule définition de check dans le Synchronizer. Shinken comprends cette propriété comme : Crée un check pour chaque Valeur de cette propriété. Le nom du check doit contenir un \$KEY\$. En créant un cheque pour chaque valeur, Shinken remplacera le \$KEY\$ par cette Valeur. \$KEY\$ est aussi utilisable dans la check command si nécessaire, pour son exécution. Des paramètres peuvent fournis aux checks (appelé \$VALUE\$ si il y en a qu'un ou \$VALUE\_x\_\$, avec x valant de 1 à n).
- **Duplicate for each default arguments**
  - Vous pouvez fournir plusieurs arguments pour chaque valeur définie dans la donnée de l'hôte utilisé par la fonctionnalité Duplicate for Each. Mais si la liste des arguments devient grande et plusieurs fois la même, il est inutile de les répéter.

## Onglet notifications

- **Notifications Enabled**
  - Cette propriété permet de définir si Shinken va envoyer des notifications aux utilisateurs listés comme cibles ( Utilisateurs ou Groupes d'utilisateurs) sur ce check. Les notifications sont envoyées:
    - Vrai: Shinken va envoyer les notifications
    - Faux: Shinken n'enverra pas les notifications
- **Users**
  - Cette propriété est utilisée pour définir quels sont les utilisateurs à notifier.
- **User groups**
  - Cette propriété est utilisée pour définir quels sont les groupes d'utilisateurs à notifier.
- **Inverval between notifications**
  - Cette propriété permet de définir les intervalles de temps entre les notifications. Cette valeur est en minutes. Note: la valeur 0 sur cette propriété signifie que seule la première notification sera envoyée.
- **Notification time period**
  - Cette propriété permet de définir la période de temps durant laquelle les notifications sont autorisées. En dehors de cette période, aucune notification ne sera envoyée. Par défaut, il n'y a pas de période de temps, et donc les notifications ne seront jamais bloquées.
- **States that raise notifications**
  - Cette propriété permet de définir quels états de l'hôte peuvent envoyer des notifications La valeur est une liste d'options séparées par des virgules:
    - d: envoie une notification sur un état CRITIQUE
    - r: envoie une notification lors d'un rétablissement (signifie que l'hôte redevient OK)
    - f: envoie une notification quand l'hôte commence ou sort du contexte FLAPPING
    - s: envoie une notification quand l'hôte entre ou sort d'une période de maintenance
    - n: n'envoyer aucune notification
 Par défaut, toutes les notifications sont autorisées.
- **First notification delay**
  - Cette propriété permet de définir combien de minutes Shinken doit attendre avant d'envoyer la première notification. Ce temps additionnel peut être mis à profit par les utilisateurs pour prendre en compte l'hôte depuis l'interface de visualisation avant que la notification ne soit envoyée. Par défaut la valeur est 0, ceci signifie que la première notification sera envoyée sans attendre.
- **Escalations**
  - Cette propriété permet de lier ce check à une ou plusieurs définitions d'escalade. Si, au bout d'un certain temps, le check n'est toujours pas revenu OK ou pas pris en compte (Contexte ACKNOWLEDGE ou DOWNTIME), la règle d'escalade sera appliquée.

## Onglet avancé

- **Poller Tag**
  - Cette propriété permet de définir le poller\_tag du check. le checks sera exécuté que par les Pollers qui ont cette valeur dans leur paramètre poller\_tags. Par défaut, la valeur de **Tag de Poller** est *non taggé*, donc toutes les Pollers n'ayant aucun de poller\_tag prendront en compte les checks d'un hôte non taggué, car la valeur par défaut pour les Pollers est aussi *non taggé*

- **Business Impact**
  - Cette propriété permet de définir l'importance métier que l'on donne à un check.  
( valeur entre 0 et 5 ) :
    - du moins important ( 0 )
    - au plus important ( 5 )
La valeur par défaut est 2.
- **Volatile**
  - Cette propriété permet de forcer Shinken à envoyer une notification pour chaque retour en erreur d'un check.  
Si non forcé, Shinken va vérifier l'état du check avant de lancer une notification (avec un nombre maximum de confirmation du statut), puis ne va envoyer de notification que toutes les **interval de notifications** (typiquement une fois par jour par défaut).  
Mais si cette option est activée, tous ces systèmes seront désactivés, et une notification sera envoyée pour chaque check retournant un status en erreur.
- **Checking that external received state expiration**
  - Cette propriété permet de définir si Shinken doit vérifier que la date de réception du dernier état reçu des outils externes ne soit pas expirée (trop vieille). Si la date est plus âgée que le seuil d'expiration, alors Shinken va forcer l'exécution de la commande de check.  
Ses valeurs peuvent être :
    - Activé: Shinken va vérifier l'expiration
    - Désactivé: Shinken ne vérifiera pas l'expiration
- **External received state expiration threshold ( x secondes )**
  - Cette propriété permet de définir quand Shinken doit forcer la commande pour check si le dernier état reçu des outils externes est plus âgé que ce seuil.  
Cette valeur est en secondes.
- **Flap Detection Enabled**
  - Cette propriété permet de définir si la détection du Contexte FLAPPING est actif sur cette hôte.  
Peut être :
    - Vrai: active la détection du contexte FLAPPING.
    - Faux: désactive la détection du contexte FLAPPING.
- **Flap Detection Options**
  - Cette propriété permet de définir quel statut d'un check sont pris en compte pour le calcul du % de FLAPPING.  
C'est une combinaison de l'un ou de plusieurs valeurs :
    - o = statut OK
    - d = statuts CRITICAL
    - u = statuts UNREACHABLE
Le séparateur est la virgule.  
**o est obligatoire.**  
Exemple: o,d,u
- **% to entering into FLAPPING Context**
  - Sur les 21 derniers statuts, chaque fois qu'un statut est différent du précédent (de OK a Warning par exemple), le % de FLAPPING augmente. Donc pour 10 changements, cela représentera un % de FLAPPING de 50% et pour 20, cela représentera 100%.  
Si ce % calculé est **supérieur** au % **d'entrée dans le Contexte FLAPPING**, alors le Contexte de check deviendra FLAPPING.
- **% to exit from Flapping Context**
  - Sur les 21 derniers Statuts, chaque fois qu'un statut est différent du précédent (de OK a Warning par exemple), le % de Flapping augmente. Donc 10 changements représentera une % de flapping de 50% et 20 représentera 100%.  
Si ce % calculé est **supérieur** au % **de sortie du Contexte FLAPPING**, alors le Contexte du check ne sera plus FLAPPING.
- **Metrics reading and storage**
  - Cette propriété permet de définir si Shinken doit lire et stocker les métriques issues de la commande de vérification.  
Les valeurs possibles sont :
    - Vrai: Shinken va lire et stocker les métriques issues de la commande de vérification de l'hôte
    - Faux: Shinken ne va pas lire et stocker les métriques issues de la commande de vérification de l'hôte
- **Event Handler Enabled**
  - Cette propriété permet de définir si Shinken va lancer une commande (définie par le paramètre **commande de gestionnaire d'évènement**) à des étapes spécifiques du statut de l'hôte :
    - quand l'hôte passe d'un état OK à un état à problème (CRITICAL ou UNKNOWN)
    - à chaque vérification additionnelle, donc un nombre **maximum de confirmation du statut**
Si la propriété est :
    - Vrai: va lancer la commande quand nécessaire
    - Faux: ne va pas lancer cette commande
- **Event Handler command**
  - Cette propriété permet de définir la commande que lancera le gestionnaire d'évènement pour cet hôte.
- **Business impact modulations**
  - Cette propriété permet de définir une ou plusieurs modulations d'impact métier. Les modulations ont une période de temps durant laquelle elles sont actives.  
Pendant cette période, la valeur d'impact métier de l'hôte sera changée par celle de la modulation.
- **Data modulations**
  - Cette propriété permet de définir une ou plusieurs modulations de données. Les modulations ont une période de temps durant laquelle elles sont actives.  
Pendant cette période, les données du check seront changées par celle de la modulation.
- **Definition order**
  - Ordre de définition