

Synchronizer - Les logs d'activité des utilisateurs (authentication et session)

Onglet : Général

? Unknown Attachment

- **Nom**

- *Clé d'import:* host_name

Cette propriété permet de définir le nom utilisé pour identifier le modèle d'hôte.
Les caractères ~!\$%^&*"'|<>?,()=/+ sont interdits dans ce champ.

⊖ Un caractère non autorisé provoquera un compteur erreur

★ ce champ ne peut pas rester vide



Le nom d'hôte ne doit pas exister en tant que nom de cluster. Si tel est le cas, une erreur sera affichée à la sauvegarde.

- **Description**

- *Clé d'import:* display_name

Cette propriété permet de définir une description de l'hôte qui pourra être affichée dans les interfaces web.

Le nom est accessible par l'intermédiaire de la variable de **\$HOSTNAME\$**.

Remarque: Si vous ne spécifiez pas la propriété, la propriété vaudra par défaut le nom de l'hôte.



Pour des raisons de sécurité, les caractères suivants <> & "/ se seront remplacés par des équivalents dans la base de données.

⊖ Un caractère non autorisé provoquera un compteur erreur

- **Adresse**

- *Clé d'import:* address

Cette propriété permet de définir l'adresse de l'hôte. Habituellement, il s'agit de l'adresse IP, bien que cela puisse être n'importe quelle valeur (pour peu qu'elle soit utile pour la détermination du statut de l'hôte).
Vous pouvez utiliser un FQDN pour l'adresse, mais gardez à l'esprit que si le DNS n'est pas disponible, les vérifications ne pourront pas être faites.

Le nom est accessible par l'intermédiaire de la variable de **\$HOSTADDRESS\$**.

Remarque: Si vous ne spécifiez pas une adresse, la propriété vaudra par défaut le nom de l'hôte.

Il y a un risque à utiliser le mécanisme par défaut - Si le DNS n'est pas disponible, la plupart des checks échoueront car les sondes dont ils dépendent ne seront pas capables de résoudre le nom de l'hôte.

- **Modèles d'hôte hérités**

- *Clé d'import:* use

Les modèles d'hôtes qui sont attachés à l'hôte pour hériter de leurs propriétés et leurs données.
L'ordre des modèles est important.

- **Ajouter dans le groupe d'hôtes**

- *Clé d'import:* hostgroups

Cette propriété permet de lister les groupes d'hôtes auxquels cet hôte appartient.

Ajouter un groupe d'hôte à cet hôte a le même effet que de rajouter cet hôte dans le groupe voulu (dans la page d'édition du groupe d'hôte).

L'ordre dans lequel vous listez les groupes d'hôtes n'a pas d'incidence.

Cette propriété gère l'[Héritage additif \(le +\)](#).

- **Royaume**

- *Clé d'import:* realm

Cette propriété permet de définir le royaume où l'hôte se trouvera.

En plaçant un hôte dans un royaume particulier, un des schedulers de ce royaume gèrera cet hôte.

- **Impact métier**

- *Clé d'import:* business_impact

Cette propriété permet de définir l'importance que l'on donne à un cluster pour le business. (valeur numérique entre 0 et 5 dans les fichiers d'imports, mais affichée dans l'interface avec des étoiles ou des dollars suivant le niveau):

- du moins important (0 => *)
- (1 => **)
- (2 => ***)
- (3 => \$\$\$\$)
- (4 => \$\$\$\$\$)
- au plus important (5 => \$\$\$\$\$\$)

La valeur par défaut est 2 (***).

- **Dépendances réseaux**

- *Clé d'import:* parents

Cette propriété permet de définir une liste d'hôtes ou de Cluster des quels l'hôte est dépendant.

Ceci signifie que si toutes les dépendances réseaux sont en état CRITICAL, alors les notifications ne seront pas lancées pour cet hôte et ses checks.

Note: les dépendances réseaux doivent être dans le même royaume que l'hôte.

L'ordre de définition n'a pas d'impact sur la supervision de cet élément.

Cette propriété gère l'[Héritage additif \(le +\)](#).

- **URL externe**

- *Clé d'import:* notes_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://* ou *https://*

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.

🚫 Une URL qui ne commence pas par *http://* ou *https://* provoquera un compteur erreur

- **Liste des URL externes**

- *Clé d'import:* notes_multi_url

Cette propriété est utilisée pour définir plusieurs URL qui peuvent être utilisées pour fournir plus d'informations sur cet hôte.

N'importe quelle URL valide peut être utilisée, commençant par **http://** or **https://**

Chaque URL commence par le nom à afficher, suivi du séparateur '~=' puis du lien.

Par exemple pour créer un lien hypertexte vers <http://www.shinken-solutions.com>, avec un texte d'URL qui dit 'Bienvenue à Shinken Solutions!', vous devez écrire ceci:

Bienvenue à Shinken Solutions!~=<http://www.shinken-solutions.com>

Pour rajouter plusieurs URL, séparez chaque entrée par '~#~'

Par exemple pour créer trois URL, vous devez écrire ceci:

Nom 1~=<http://lien1.com>~#~Nom 2~=<http://lien2.com>~#~Nom 3~=<http://lien3.com>

Important

Dans cette propriété, des données (macro) de cet hôte peuvent être utilisées dans l'URL.

- La référence à ces données se fait via le principe de [LES DONNEES \(Remplacement dynamique de contenu - Anciennement les MACROS \)](#).
- Pour intégrer une donnée dans l'URL, il suffit de rajouté \$_HOSTDATA1\$
 - HOSTUID

Les données (macros) qui peuvent être utilisées sont les suivantes:

- Données globales
- Données accrochées à l'hôte présentes dans l'onglet Données
- et les propriétés (données statiques) suivantes:
 - HOSTUID
 - HOSTNAME
 - HOSTDISPLAYNAME
 - HOSTALIAS
 - HOSTADDRESS
 - HOSTREALM
 - HOSTADDRESS

- Remarque: Le nom à afficher est optionnel. (mais retournera un warning dans l'interface)

Ces URL seront disponibles dans l'interface de visualisation dans le volet détail présentant l'hôte et également dans la liste sur la colonne "URL externe"

Cela peut être pratique si vous désirez faire une description écrite détaillée de l'hôte, une procédure de reprise sur panne, ou toute autre information qui sera visible pour les autres membres de l'équipe.

 Une URL qui ne commence pas par `http://` ou `https://` provoquera un compteur erreur

- **Activé**

- *Clé d'import:* enabled

Permettre que cet élément soit pris en compte lors du passage de Staging en Production.

Pour les modèles uniquement :

- **Modèle visible pour les Administrateurs de SI?**

- *Clé d'import:* for_all_users

Cette propriété permet de cacher ce modèle d'hôte aux Administrateurs de SI.

Cela peut-être intéressant pour diminuer le nombre de modèle visible pour simplification, ou par sécurité.


Onglet : Données

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



Si vous donnez un nom de donnée protégée à l'une de vos données, vous ne pourrez plus modifier ce nom par la suite.

Cette modification est interdite afin d'éviter qu'une donnée protégée devienne visible, en changeant son nom.

 Unknown Attachment

Onglet : Droits de l'utilisateur

Il est possible pour chaque hôte, de définir qui peut voir, éditer, ou encore recevoir les notifications. Le fonctionnement de ce mécanisme est expliqué dans la page [Droits d'accès à un hôte](#). Ces propriétés gèrent l'[Héritage additif \(le +\)](#).

 Unknown Attachment

Ces 6 propriétés se synchronisent les unes avec les autres, pour éviter d'avoir des droits aberrants (comme par exemple avoir le droit de d'éditer l'élément mais sans avoir droit de le voir.

Onglet : Supervision

? Unknown Attachment

- **Période de supervision**
 - Clé d'import: `check_period`

Cette propriété permet de choisir la période de supervision où sont faites les vérifications de l'état de l'hôte par Shinken.
- **Commande de vérification**
 - Clé d'import: `check_command`

Cette propriété permet de spécifier la commande qui est employée pour vérifier si l'hôte est dans un statut en OK ou CRITIQUE. Si vous laissez vide cette propriété, la commande de contrôle par défaut sera employée.
- **Nb maximum de tentatives de confirmation du statut de l'hôte**
 - Clé d'import: `max_check_attempts`

Cette propriété permet de définir le nombre de tentative(s) où Shinken recommencera la commande de vérification d'un hôte 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 vérifie le statut de cette hôte, laissez cette propriété à 1, mais laissez vide la commande de vérification.

⊖ Une valeur non numérique provoquera un compteur erreur
- **Temps maximum d'exécution d'un check**
 - Clé d'import: `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.

⊖ Une valeur non numérique provoquera un compteur erreur
- **Seuil d'alerte de l'utilisation CPU (sec)**
 - Nombre de secondes que peut consommer une commande avant d'apparaître en Warning dans le check 'Scheduler - Performance'
La valeur sera récupérée dans l'ordre, par la commande, par l'élément Check, par l'élément Hôte, ou par la valeur par défaut (`shinken.cfg:warning_threshold_cpu_usage`)

⊖ Une valeur non numérique provoquera un compteur erreur
- **Intervalle entre les vérifications (x minutes)**
 - Clé d'import: `check_interval`

Cette propriété permet de définir le temps entre 2 vérifications de l'état de l'hôte.

⊖ Une valeur non numérique provoquera un compteur erreur
- **Intervalle de nouvelles tentatives de confirmation d'état (x minutes)**
 - Clé d'import: `retry_interval`

Cette propriété permet de définir le temps d'attente avant de relancer la commande de vérification de l'hôte 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**

⊖ Une valeur non numérique provoquera un compteur erreur
- **Les commandes de vérifications sont ordonnancées et lancées par Shinken**
 - Clé d'import: `active_checks_enabled`

Cette propriété permet de définir si Shinken ordonnance et lance les commandes de vérifications pour déterminer l'état de cet hôte. Ses valeurs peuvent être:

 - True: Shinken va ordonnancer et lancer des commandes de vérifications

- False: Shinken ne va pas ordonnancer et lancer les commandes de vérifications
- **Shinken accepte les états reçus depuis des outils externes pour cet hôte**
 - Clé d'import: `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
- **Période de maintenance planifiée**
 - 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.

Onglet : Checks

Cet onglet résume quels sont les Checks qui seront appliqués à l'hôte, et leur provenance (venant de quel modèle d'hôte, ou directement appliqué à l'hôte).

Les modèles sont listés par ordre d'héritage (dans l'exemple `mssql` et `ms-sql-availability`)

Cet onglet permet également de vérifier la duplication par la fonctionnalité permettant de [Dupliquer pour chaque valeur de la Donnée de l'hôte](#)

Ici, dans l'image, un seul check utilise des Duplicate Foreach.

Cette page permet également de surcharger les valeurs du checks.

Enfin elle permet d'essayer les checks depuis leur configuration actuelle afin de vérifier leur résultat.

? Unknown Attachment

Essayer les checks

Les boutons dans la colonne [Essayer ce check] permettent d'évaluer ou d'essayer directement l'exécution d'un check, avec la résolution de ses données.

Le premier bouton (roue crantée) permet de simuler la résolution des arguments, sans l'exécution du check. Cette évaluation vous permettra d'afficher la commande avec les données interprétées.

Le second bouton (roue crantée + icône play) permet d'évaluer et de simuler son exécution depuis la plateforme de configuration (synchronizer). Cette exécution n'utilisera pas vos pollers. Vous pouvez donc utiliser ce bouton pour tester votre commande sans affecter vos serveurs pollers en production.

Le tableau récapitulatif présente les données récupérées, et le résultat de la commande en prenant en compte les éventuelles modulations.

Le troisième bouton (icône play) permet d'évaluer et de simuler son exécution directement sur les pollers, comme lors de l'exécution normale sur votre architecture Shinken. Vous pouvez donc utiliser ce bouton pour tester votre commande sur votre environnement de production.

Le tableau récapitulatif présente les données récupérées, et le résultat de la commande en prenant en compte les éventuelles modulations.

Le dernier bouton (croix) permet de cacher le bloc d'essai du check.

? Unknown Attachment

? Unknown Attachment



Afin de tester au mieux vos checks, si une erreur survient pendant l'essai du check, celle-ci vous sera affichée à la place des résultats.

Les checks cachés

Certains checks peuvent être affichés en grisé avec le libellé 'Caché'.

Cette situation se produit lorsque deux checks ayant le même nom sont attachés sur deux modèles d'hôte hérités ou bien directement sur l'hôte. Dans ce cas le check attaché sur le modèle d'hôte apparaissant en amont dans l'ordre d'héritage sera actif, l'autre sera caché.

Inverser l'ordre d'héritage de ces modèles d'hôte inversera également le statut (caché / actif) des checks.

? Unknown Attachment

Onglet : Notifications

? Unknown Attachment

- **Notifications activées**

- *Clé d'import:* notifications_enabled

Cette propriété permet de définir si Shinken va envoyer des notifications aux utilisateurs listés comme destinataires (Utilisateurs ou Groupes d'utilisateurs) sur ce hôte.

Les notifications sont envoyées:

- Premièrement après la fin de la période de nouvelle tentative, une fois que l'état CRITICAL est confirmé.
- ensuite après chaque **intervalle de notification**, exprimé en minutes
- quand l'hôte redevient OK

Les valeurs possibles sont:

- Vrai: Shinken va envoyer les notifications
- Faux: Shinken n'enverra pas les notifications

- **Intervalle de temps entre les notifications (x minutes)**

- *Clé d'import:* notification_interval

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.

⊖ Une valeur non numérique provoquera un compteur erreur

- **Période de temps de notification**

- *Clé d'import:* notification_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.

- **Options de notification**

- *Clé d'import:* notification_options

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
- u: envoie une notification sur un état UNREACHABLE
- 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.

⊖ Une option non existante provoquera un compteur erreur

- **Décalage de la première notification**

- Clé d'import: 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.

⊖ Une valeur non numérique provoquera un compteur erreur

- **Escalades**

- Clé d'import: escalations

Cette propriété permet de lier cet hôte à une ou plusieurs définitions d'escalade.

Si, au bout d'un certain temps, l'hôte n'est toujours pas revenu OK ou pas pris en compte (Contexte ACKNOWLEDGE ou DOWNTIME), la règle d'escalade sera appliquée. Cette propriété gère l'[Héritage additif \(le +\)](#).

Onglet : Expert

? Unknown Attachment

- **Tag de Poller**

- Clé d'import: poller_tag

Cette propriété permet de définir le poller_tag de l'hôte.

Tous les checks de l'hôte ne seront exécutés 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 tous les Pollers n'ayant aucun poller_tag prendront en compte les checks d'un hôte non taggé, car la valeur par défaut pour les Pollers est *Aucun*

- **Tag de Reactionner**

- Clé d'import: reactionner_tag

Cette propriété permet de définir le reactionner_tag de l'hôte.

Toutes les notifications sur l'hôte ne seront exécutées que par les Reactionners qui ont cette valeur dans leur paramètre reactionner_tags.

Par défaut, la valeur de **Tag de Reactionner** est **non taggé**, donc les Reactionners n'ayant aucun reactionner_tag prendront en compte les checks d'un hôte non taggé, car la valeur par défaut pour les Reactionners est *Aucun*

- **Lire et stocker les métriques**

- Clé d'import: process_perf_data

Cette propriété permet de définir si Shinken doit lire et stocker les métriques issues de la commande de vérification de l'hôte. 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

- **Détection du FLAPPING activé**

- Clé d'import: flap_detection_enabled

Cette propriété permet de définir si la détection du Contexte FLAPPING est actif sur cet hôte.

Peut être:

- Vrai: active la détection du contexte FLAPPING.
- Faux: désactive la détection du contexte FLAPPING.

- **Flap Detection Options**

- Clé d'import: flap_detection_options

Cette propriété permet de définir quels statuts d'un hôte sont pris en compte pour le calcul du % de FLAPPING.

C'est une combinaison de l'une 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

⊖ Une option non existante provoquera un compteur erreur

- **Sortie du contexte FLAPPING**

- Clé d'import: low_flap_threshold

Sur les 21 derniers statuts, chaque fois qu'un statut est différent du précédent (de OK à Warning par exemple), le % de FLAPPING augmente. Donc 10 changements représenteront un % de flapping de 50% et 20 représenteront 100%.

Si ce % calculé est **inférieur** au % de sortie du Contexte **FLAPPING**, alors le Contexte de l'hôte ne sera plus FLAPPING.

- **Entrée du contexte FLAPPING**

- Clé d'import: high_flap_threshold

Sur les 21 derniers statuts, chaque fois qu'un statut est différent du précédent (de OK à 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 l'hôte deviendra FLAPPING.

Il sortira de ce Contexte quand ce pourcentage calculé sera **inférieur** au % de sortie du Contexte **FLAPPING**.

- **Gestionnaire d'événement activé**

- Clé d'import: event_handler_enabled

Cette propriété permet de définir si Shinken va lancer une commande (définie par le paramètre **commande de gestionnaire d'événement**) 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

- **Commande lancée par le gestionnaire d'événements**

- Clé d'import: event_handler

Cette propriété permet de définir la commande que lancera le gestionnaire d'événement pour cet hôte.

- **Vérification que l'état reçu des outils externes ne soit pas expiré**

- Clé d'import: check_freshness

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 d'une commande de check pour cet élément.

Ses valeurs peuvent être:

- Activé: Shinken va vérifier l'expiration
- Désactivé: Shinken ne vérifiera pas l'expiration

- **Seuil d'expiration des états reçus des outils externes (x secondes)**

- Clé d'import: freshness_threshold

Cette propriété permet de définir quand Shinken doit forcer une commande de vérification pour cet hôte si le dernier état reçu des outils externes est plus âgé que ce seuil.

Cette valeur est en secondes.

 Une valeur non numérique provoquera un compteur erreur

- **Modulation d'impact métier**

- Clé d'import: business_impact_modulations

Cette propriété permet de définir une ou plusieurs modulations d'impact métier. Les modulations ont une période temps durant laquelle elles sont actives.

Pendant cette période, la valeur d'impact métier de l'hôte sera changée par celle de la modulation.

Cette propriété gère l'**Héritage additif (le +)**.

- **Modulation de données**

- Clé d'import: macromodulations

Cette propriété permet de définir une ou plusieurs modulations de données. Les modulations ont une période temps durant laquelle elles sont actives.

Pendant cette période, les données de l'hôte seront changées par celle de la modulation.

Cette propriété gère l'**Héritage additif (le +)**.

- **Modulation de résultats**

- Clé d'import: resultmodulations

Cette propriété permet de définir une ou plusieurs modulations de résultats. (maximum 4)

Les modulations de résultats redéfinissent le statut de sortie d'un check, en fonction de son statut initial, d'une période de temps, ou de sa sortie.

Cette propriété gère l'**Héritage additif (le +)**.

- **Check à exclure**

◦ *Clé d'import:* service_excludes

Cette propriété permet d'exclure des checks

Vous pouvez exclure plusieurs checks à l'aide d'une liste de noms séparés par des virgules.