

# Modèle Switch-SNMPv1v2-detailed

## Sommaire

- Contexte
- Sommaire des checks
- Les données
  - Les données communes
  - Les données spécifiques
    - Pour le check "Cpu Switch SNMPv1v2"
    - Pour le check "Memory Switch SNMPv1v2"
  - Les données DFE ( Duplicate Foreach )
    - Utilisation
- Comment appliquer un modèle d'hôte à un hôte

## Contexte

Le modèle **Switch-SNMPv1v2-detailed** permet d'avoir un point de vue plus détaillé sur **chaque interface** du switch, et donner des informations plus précises comme la mémoire RAM disponible et l'état du CPU.

Si le besoin de modifier certains éléments ( *commandes, checks ou modèles d'hôtes* ) se présente, il faut lire la page [Les bonnes pratiques d'utilisation d'un pack livré par Shinken](#) ).

## Sommaire des checks

Nom	Description
Hardware Health Switch SNMPv1v2	Il vérifie le bon fonctionnement physique du matériel de l'appareil ( <i>alimentation, ventilateurs, températures, disques...</i> ).  ( voir la page <a href="#">Hardware Health Switch SNMPv1v2 - Modèle Switch-SNMPv1v2-detailed</a> )
Cpu Switch SNMPv1v2	Il récupère et affiche les informations concernant l'utilisation du/des processeur(s) du switch.  ( voir la page <a href="#">Cpu Switch SNMPv1v2 - Modèle Switch-SNMPv1v2-detailed</a> )
Memory Switch SNMPv1v2	Il récupère et affiche les informations sur l'utilisation mémoire du switch.  ( voir la page <a href="#">Memory Switch SNMPv1v2 - Modèle Switch-SNMPv1v2-detailed</a> )
<b>\$KEY\$</b> : Errors Switch SNMPv1v2  ( <i>exemple</i> : <b>\$KEY\$</b> = <i>Vlan201, Vlan202</i> => <i>Vlan201: Errors Switch SNMPv1v2</i> => <i>Vlan202: Errors Switch SNMPv1v2</i> )	Il récupère et affiche le taux moyen d'erreurs en entrée et en sortie des interfaces.  ( voir la page <a href="#">\$KEY\$ : Errors Switch SNMPv1v2 - Modèle Switch-SNMPv1v2-detailed</a> )  Utilise le mécanisme de DuplicateForEach ( <i>Données SWITCH_INTERFACES</i> )
<b>\$KEY\$</b> : Status Switch SNMPv1v2  ( <i>exemple</i> : <b>\$KEY\$</b> = <i>Vlan201, Vlan202</i> => <i>Vlan201: Status Switch SNMPv1v2</i> => <i>Vlan202: Status Switch SNMPv1v2</i> )	Il récupère et affiche les informations concernant le statut des interfaces réseaux de votre switch.  ( voir la page <a href="#">\$KEY\$ : Status Switch SNMPv1v2 - Modèle Switch-SNMPv1v2-detailed</a> )  Utilise le mécanisme de DuplicateForEach ( <i>Données SWITCH_INTERFACES</i> )

<p><b>\$KEY\$</b> : Usage Switch SNMPv1v2</p> <p>( <i>exemple</i> : <b>\$KEY\$ = Vlan201, Vlan202</b>  =&gt; <i>Vlan201: Usage Switch SNMPv1v2</i>  =&gt; <i>Vlan202: Usage Switch SNMPv1v2</i> )</p>	<p>Il récupère et affiche les informations sur le volume d'utilisation de toutes les interfaces réseaux de votre switch.</p> <p>( voir la page <a href="#">\$KEY\$ : Usage Switch SNMPv1v2 - Modèle Switch-SNMPv1v2-detailed</a> )</p> <p>Utilise le mécanisme de DuplicateForEach ( <i>Données SWITCH_INTERFACES</i> )</p>
---	---

## Les données

### Les données communes

Nom	Modifiable sur	Unité	Défaut	Valeur par défaut à l'installation de Shinken	Description
SNMPCOMMUNITY	l'Hôte ( <i>Onglet Données</i> )	--	\$SNMPCOMMUNITYREAD\$	<b>public</b>	La Communauté SNMP v1/v2 défini sur votre switch : <ul style="list-style-type: none"> <li>En SNMP v1/v2, la communauté est un équivalent à un ID ou à un mot de passe pour se connecter aux équipements.</li> </ul>
SWITCH_TIME_OUT	l'Hôte ( <i>Onglet Données</i> )	secondes	60	<b>60</b>	Variable permettant au check de s'arrêter après un certain temps si une tâche ne s'est pas terminée. <ul style="list-style-type: none"> <li>Permet d'éviter que le programme ne s'exécute indéfiniment et de prévenir des problèmes de performances.</li> </ul> <p>( <i>La valeur doit être supérieure à 3</i> )</p>
SWITCH_PORT	l'Hôte ( <i>Onglet Données</i> )	--	161	<b>161</b>	Variable correspondant au port SNMP sur lequel le démon SNMP de votre équipement écoute.
SWITCH_WORKING_FOLDER	l'Hôte ( <i>Onglet Données</i> )		/var/tmp /check_nwc_health	<b>/var/tmp /check_nwc_health</b>	Dossier dans lequel la sonde stockera ses fichiers de travail



#### Remarque

Par défaut, la donnée SNMPCOMMUNITY prend le contenu de la donnée globale \$SNMPCOMMUNITYREAD\$ dont la valeur est modifiable dans le fichier `/etc/shinken/resource.d/snmp.cfg`,

- voir la page [LES VARIABLES \( Remplacement dynamique de contenu - Anciennement les MACROS \)](#), pour comprendre leur utilisation.

La modification des valeurs par défaut présentes dans le fichier du serveur ( `/etc/shinken/resource.d/snmp.cfg` ) nécessite un redémarrage des démons suivants :

- Synchronizer pour le voir dans l'interface,
- Arbiter pour que la modification soit prise en compte en supervision.

Donc, pour changer la valeur de SNMPCOMMUNITY, vous pouvez :

- changer sa valeur dans un hôte,
- changer sa valeur dans le modèle Switch-SNMPv1v2 ( *mais cela impactera tous les hôtes utilisant ce modèle* ).
- modifier le contenu de la donnée globale \$SNMPCOMMUNITYREAD\$ ( *en prenant en compte que **tous les hôtes** utilisant cette donnée seront impactés* ).

### Les données spécifiques

Pour le check "[Cpu Switch SNMPv1v2 - Modèle Switch-SNMPv1v2-detailed](#)"

Nom	Modifiable sur	Unité	Défaut	Valeur par défaut à l'installation de Shinken	Description
SWITCH_CPU_LOAD_CRIT	l'Hôte ( Onglet Données )	Pourcentage	90	90	Valeur de charge à partir de laquelle le check passe en CRITIQUE.
SWITCH_CPU_LOAD_WARN	l'Hôte ( Onglet Données )	Pourcentage	80	80	Valeur de charge à partir de laquelle le check passe en AVERTISSEMENT.

### Pour le check "Memory Switch SNMPv1v2 - Modèle Switch-SNMPv1v2-detailed"

Nom	Modifiable sur	Unité	Défaut	Valeur par défaut à l'installation de Shinken	Description
SWITCH_MEMORY_USAGE_CRIT	l'Hôte ( Onglet Données )	Pourcentage	90	90	Valeur de charge à partir de laquelle le check passe en CRITIQUE.
SWITCH_MEMORY_USAGE_WARN	l'Hôte ( Onglet Données )	Pourcentage	80	80	Valeur de charge à partir de laquelle le check passe en AVERTISSEMENT.

### Les données DFE ( Duplicate Foreach )

Nom	Modifiable sur	Unités	Défaut	Valeur par défaut à l'installation de Shinken	Description
SWITCH_INTERFACES	l'Hôte ( Onglet Données )	---	port1, port2	port1,port2	Nom de ou des interfaces ( séparés par une virgule ), exemple "Vlan01, Vlan02,Vlan03".  Check(s) impacté(s) : <ul style="list-style-type: none"> <li>• \$KEY\$ : Errors Switch SNMPv1v2 ( voir la page \$KEY\$ : Errors Switch SNMPv1v2 - Modèle Switch-SNMPv1v2-detailed ).</li> <li>• \$KEY\$ : Status Switch SNMPv1v2 ( voir la page \$KEY\$ : Status Switch SNMPv1v2 - Modèle Switch-SNMPv1v2-detailed ).</li> <li>• \$KEY\$ : Usage Switch SNMPv1v2 ( voir la page \$KEY\$ : Usage Switch SNMPv1v2 - Modèle Switch-SNMPv1v2-detailed ).</li> </ul>

### Utilisation

Il vous faudra saisir les noms des interfaces à surveiller :

- Dans l'exemple ci-dessous, le modèle liste **par défaut** les interfaces appelées **port1** et **port2**.

Locale	
Locale [ 0 ]	
Venant des modèles	
Switch-SNMPv1v2-detailed [ 8 ]	
(Duplicate Foreach)	
Switch-SNMPv1v2-detailed [ 1 ]	Utilisé par les checks [ 3 / 3 ]
Nom	Valeur
? SWITCH_INTERFACES	port1,port2 [ Dans le modèle Switch-SNMPv1v2-detailed ]

- Changer cette liste avec les noms qui concernent votre équipement :
  - Par exemple Eth1, Eth2, Eth3, pour avoir les checks surveillant ces interfaces.

Locale	
Locale [ 0 ]	
Venant des modèles	
Switch-SNMPv1v2-detailed [ 8 ]	
(Duplicate Foreach)	
Switch-SNMPv1v2-detailed [ 1 / 1 ]	Utilisé par les checks [ 3 / 3 ]

Nom	Valeur
SWITCH_INTERFACES	Eth1,Eth2,Eth3

Pour cette donnée,

- La **virgule** sert de séparateur.
- Vous pouvez mettre n'importe quelle chaîne de caractères.
  - Si vous voulez surveiller les interfaces réseaux non continues, comme les Eth1, Eth2, Eth3 et Eth9, Eth10, il vous suffit d'en faire la liste :

Eth1,Eth2,Eth3,Eth9,Eth10

### Astuce

Si vous voulez générer une liste de 256 ports de la forme Eth0,Eth1, Eth2, ... Eth255, le faire à la main serait très fastidieux!

Nous avons donné la possibilité de générer **AUTOMATIQUEMENT** les nombres :

- La syntaxe **[nombre1-nombre2]** permet de générer plusieurs valeurs
- donc pour avoir notre liste, saisissez **Eth[0-255]**

Syntaxe	Résultats	Commentaire
<b>Eth[5-9]</b>	Eth5,Eth6,Eth7,Eth8,Eth9	Pour une liste d'interfaces continues
<b>Eth[5-9], Eth [60-65]</b>	Eth5,Eth6,Eth7,Eth8,Eth9,Eth60,Eth61,Eth62,Eth63,Eth64,Eth65	Pour faire des sauts entre plusieurs listes d'interfaces continues
<b>Eth[2-3][0-5]</b>	Eth20,Eth21,Eth22,Eth23,Eth24,Eth25,Eth30,Eth31,Eth32,Eth33,Eth34,Eth35	Pour faire un saut régulier entre des listes d'interfaces

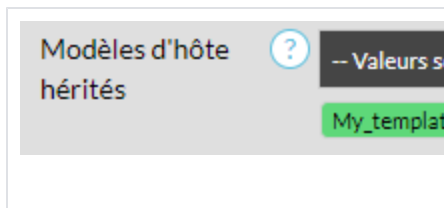
La génération de la liste des valeurs souhaitées se fera par l'Arbiter après la mise en production, elle sera visible uniquement dans l'interface de visualisation, du coté de l'interface de production la syntaxe du DFE ne seras pas interprétée.

## Comment appliquer un modèle d'hôte à un hôte

### Application du modèle via l'interface de Configuration

Dans l'interface de Configuration :

- créer ou éditer un hôte ( voir la page [Editer un Hôte](#) ),
- ajouter le modèle "My\_template" dans la propriété "Modèles d'hôte hérités" à l'aide du menu déroulant.



## Application du modèle via un collecteur d'import de fichiers au format .cfg

Dans votre fichier de définition de vos éléments à importer via votre collecteur :

- créer ou éditer la définition de votre hôte,
- ajouter la valeur "**My\_template**" ( *selon vos besoins* ), dans la propriété "**use**",
- importer le contenu du fichier via un collecteur de type "cfg-file-import" ( voir la page [Collecteur de type \( cfg-file-import \) - Import depuis des fichiers au format .cfg](#) ).

```
define host {  
    host_name    My_host  
    use          My_template  
}
```