


```
$(ma_valeur_par_défaut_1)$$$(ma_valeur_par_défaut_2)$
```

Le check peut donc utiliser, dans sa configuration, **\$KEY\$**, et **\$VALUE1\$** jusqu'à **\$VALUE16\$**.

Ces Variables seront remplacées de façon spécifique à chaque copie.

Plus d'information sur les Variables [ici](#) avec le cas spécifique du "Duplicate for each".

L'hôte

Pour que l'hôte duplique ses checks, il faut renseigner la valeur de la donnée "Duplicate for each". Ce champ est obligatoire, en cas d'oubli la donnée apparaît en erreur et aucun check n'est généré.

The screenshot shows the Nagios XI configuration page for host 'srv-oracle-2'. A modal error message is displayed, stating 'Syntaxe Invalide' and 'Erreurs: 1. La clé N° 1 est vide.' Below the error, a tooltip explains the expected format: 'Chaque valeur Duplicate Foreach est présentée sous la forme suivante: CLE\$(ARG1)\$(ARG2)\$...\$(ARGn)\$ (CLE obligatoire, arguments facultatifs). Les valeurs sont séparées entre elles par des virgules.'

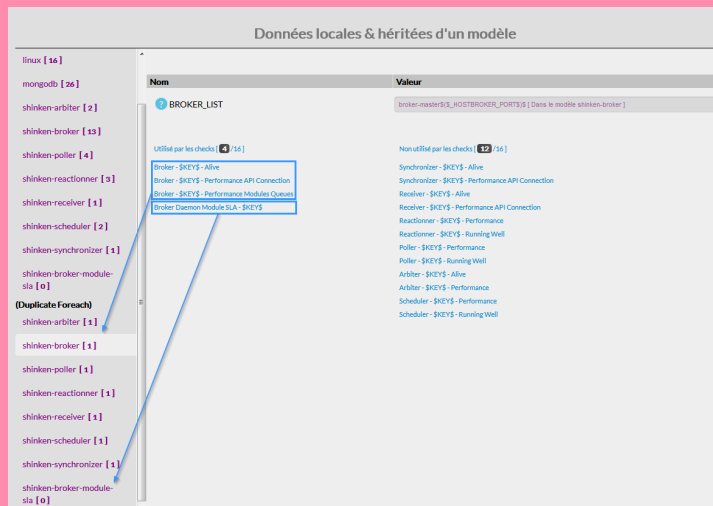
Un hôte ayant des checks à dupliquer qui lui sont directement attachés ou provenant de modèles hérités, doit obligatoirement fournir des valeurs dans sa donnée de Duplicate for each.

Lors de l'édition, chaque clé peut être spécifiée avec ses valeurs sur une ligne séparée.

The screenshot shows the 'Duplicate for each' field in the configuration page. The field contains the following text: 'ok, warning\$(Shinken)\$, critical\$(Shinken Entreprise)\$(testing)\$'. The interface also shows a table with columns 'Nom' and 'Valeur'.

Nom	Valeur
STATUS	ok, warning\$(Shinken)\$, critical\$(Shinken Entreprise)\$(testing)\$

Lorsque plusieurs checks utilisent la même donnée Duplicate For Each (ex: *BROKER_LIST* dans le screen ci-dessous), celle-ci n'apparaît qu'une fois et sa valeur est partagée pour tous les checks.



Syntaxe de la valeur "Duplicate for each":

La valeur de la donnée duplicate for each a une syntaxe précise:

```
ma_cle1$(mon_arg1_cle1)$$(mon_arg2_cle1)$
```

On définit le nom de la clé et les arguments sont passés entre \$(et)\$ de manière consécutive. Les arguments sont facultatifs (s'ils ne sont pas renseignés, les arguments par défaut pourront être utilisés).

Pour renseigner plusieurs clés, on reprend la même syntaxe séparée par des virgules:

```
Exemple 1 : ma_cle1$(mon_arg1_cle1)$$(mon_arg2_cle1)$, ma_cle2$(mon_arg1_cle2)$$(mon_arg2_cle2)$,
ma_cle3$(mon_arg1_cle3)$$(mon_arg2_cle3)$
Exemple 2 : ma_cle1, ma_cle2$(mon_arg1_cle2)$$(mon_arg2_cle2)$, ma_cle3$(mon_arg1_cle3)$
```

Dès que la saisie du champ est terminée, le nombre de checks de l'hôte est mis à jour.

Exemple de donnée "Duplicate for each":

```
ok,warning$(1)$$(Dummy WARNING)$,
critical$(2)$$(Dummy CRITICAL)$,unknown$(3)$$(Dummy UNKNOWN)$
```

Dans cet exemple, la première valeur/clé "ok" n'a aucun argument défini, la valeur par défaut des arguments sera alors utilisée, à savoir, 2 arguments dans notre exemple :

```
$(0)$$(Dummy OK)$
```

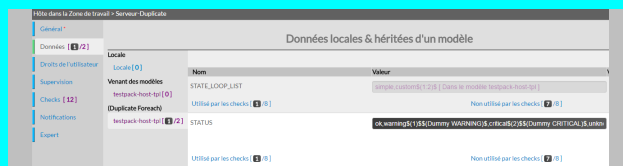
Pour la valeur/clé "warning", cette clé a deux arguments "1" et "Dummy WARNING".

Plus de détails sont dans la prochaine section "Les valeurs".

L'onglet check permet de vérifier le résultat de la duplication.

Les checks obtenus ont des configurations identiques. Pour pouvoir leur donner des données spécifiques, il faudra utiliser les valeurs.

On voit ici que le check a bien été dupliqué pour les 4 valeurs de données "Duplicate for each".



Pour chaque duplication, les arguments sont bien récupérés pour le passage dans la commande. (Via les Variables spéciales \$VALUE\$ qui correspondent aux différents \$ARGn\$)



Les valeurs

Pour que chaque check puisse exécuter une vérification différente sur l'hôte, il est possible de paramétrer spécifiquement chaque copie. Pour cela, l'hôte peut passer des arguments de duplication à chaque copie sous la forme de valeurs dans la donnée de "Duplicate for each".

Par exemple, pour une commande qui demande deux arguments, la syntaxe de la donnée de "Duplicate for each" est la suivante :

```
red$(IndianRed)$$(CD5C5C)$,green$(ForestGreen)$,blue
```

Remarque: Les arguments manquants seront remplacés par les valeurs par défaut.

Check	Variables	Valeur
Check-Duplicate-red	\$KEY\$	red
	\$VALUE1\$	IndianRed
	\$VALUE2\$	CD5C5C
Check-Duplicate-green	\$KEY\$	green
	\$VALUE1\$	ForestGreen
	\$VALUE2\$	default_value2
Check-Duplicate-blue	\$KEY\$	blue
	\$VALUE1\$	default_value1
	\$VALUE2\$	default_value2

Syntaxe des fichiers de configuration

Voici dans notre exemple, la définition d'un check duplicate for each qui va s'appliquer à un modèle d'hôte:

```
define service {
  service_description check-dummy-$KEY$
  register 0
  host_name testpack-host-tpl
  use check-dummy-tpl
  duplicate_foreach _STATUS
  default_value $(0)$$(Dummy OK)$
}

define host {
  name testpack-host-tpl
  check_command check-host-alive
  check_interval 1
  retry_interval 1
  register 0
  _STATUS ok,warning$(1)$$(Dummy WARNING)$,critical$(2)$$(Dummy CRITICAL)$,unknown$(3)$$(Dummy UNKNOWN)$
}
```