

zephir-client - Scénario #35858

Contenu modifié zephir.eol (rétro-portage 2.7)

29/03/2024 09:39 - Tom Ricci

Statut:	Nouveau	Début:	29/03/2024
Priorité:	Normal	Echéance:	29/03/2024
Assigné à:	Benjamin Bohard	% réalisé:	0%
Catégorie:		Temps estimé:	0.00 heure
Version cible:	Carnet Cadoles	Temps passé:	0.00 heure

Description

Bonjour,

Les clés des dictionnaires dans le fichier `/var/lib/zephir/conf/UAI/ID/variables.eol` ne sont pas ordonnées de la même manière que sur le Zéphir, ce qui donne un MD5 différent, et affiche le message « contenu modifié zephir.eol » alors que les valeurs de configuration sont les mêmes.

La modification suivante règle le problème (adaptation du code pour la 2.8.0 et suivante, cf [#34954](#))

```
359,366c359,372
```

```
<         with open(var_eol, 'w') as f_var:
<             for key, value in sorted(conf.items()):
<                 if key not in ['mode_zephir', '__version__'] and isinstance(va
lue, dict) and 'val' in value:
<                     if type(value['val']) in [list, dict, tuple]:
<                         converted_value = to_bytes(value['val'])
<                     else:
<                         converted_value = value['val']
<                     f_var.write('{}:{}\n'.format(key, converted_value))
---
>         for var_name in var_names:
>             if var_name not in ('mode_zephir', '__version__'):
>                 # test supplémentaires au cas où d'autres 'fausses variables'
>                 # que __version__ seraient ajoutées.
>                 # cf : https://dev-eole.ac-dijon.fr/issues/10548
>                 if type(conf[var_name]) == dict and 'val' in conf[var_name]:
>                     val = conf[var_name].get('val')
>                     if isinstance(val, dict):
>                         val = {str(k): str(v) for k, v in sorted(val.items(), ke
y=lambda x: x[0])}
>                     if isinstance(val, list) or isinstance(val, tuple):
>                         val = to_bytes(val)
>                     var_data = "{0}:{1}\n".format(var_name, val)
>                     f_var.write(var_data) #.encode(charset))
>         f_var.close()
```

Serait-il possible s'il vous plait d'intégrer cette modification dans une future version du paquet 2.7 ?

D'avance merci, bien cordialement,

Tom.

Sous-tâches:

Tâche # 35883: Étude

À valider

Tâche # 35890: Exposer la version du Zéphir

À valider

Tâche # 35891: Conditionner le formatage du fichier de variables à la version du serveu...

À valider

Demandes liées:

Lié à zephir-client - Scénario #35868: Corrections client Zéphir à rétro-port...

Nouveau

01/10/2022

01/01/2024

Historique

#1 - 05/04/2024 17:19 - Joël Cuissinat

- *Tâche parente mis à #35868*

#2 - 10/04/2024 11:50 - Benjamin Bohard

Un test sur les VM ne permet pas de déterminer ce que change le rétro-portage dans l'ordre du dictionnaire. La différence majeure entre les deux codes repose sur le moment du tri dans le cas des dictionnaires.

En 2.7, le cas des dictionnaires est traité dans la fonction `to_bytes`, avec un tri après conversion :

```
    if isinstance(objet, dict):
        dico={}
        for cle in objet:
            dico[to_bytes(cle)] = to_bytes(objet[cle])
        return '{{{}}}'.format(', '.join(['{}: {}'.format(el[0], el[1]) for el in sorted(dico.items())
]))
```

En 2.8+, la fonction `to_bytes` n'est pas utilisée (pas de problème de représentation des chaînes du type `u"`) et le tri a lieu avant conversion.

La fonction était utilisée globalement pour traiter les dictionnaires, les listes, les tuples mais également les chaînes unicode.

On perd également la récursion mais les cas où elle est utile ne sont peut-être pas fréquents (existants ?).

#3 - 10/04/2024 13:39 - Tom Ricci

Bonjour Benjamin,

Afin d'aider, voici un exemple de différences dans le fichier `variable.eol` entre celui généré par le Zéphir 2.7 et celui de l'Amon 2.7 :

```
# diff variables.eol variables.eol.amon
35c35
< netmask_admin_eth0: {'1': '0.0.0.0', '0': '255.255.255.248'}
---
> netmask_admin_eth0: {'0': '255.255.255.248', '1': '0.0.0.0'}
38c38
< netmask_ssh_eth0: {'1': '0.0.0.0', '0': '255.255.255.248'}
---
> netmask_ssh_eth0: {'0': '255.255.255.248', '1': '0.0.0.0'}
```

Il est donc possible de reproduire sur vos VM de mettre plusieurs sous-réseaux dans la « Adresse IP réseau autorisée pour les connexions SSH » ou « Adresse IP réseau autorisée pour administrer le serveur », par exemple.

Je n'étais pas allé jusqu'à la fonction `to_bytes`, j'ai simplement calqué le code 2.8 en adaptant pour gérer les chaînes unicodes (on voit au passage que la ligne 358 « `f_var = file(var_eol, 'w')` » a elle aussi été précédemment rétro-portée et fait doublon avec le « `open` » de la ligne suivante).

Bien cordialement,

Tom.

#4 - 10/04/2024 14:12 - Benjamin Bohard

Je me demande si la raison de la différence de tri dans ce cas précis n'est pas davantage liée au serveur Zéphir : le tri est bien effectué côté client depuis la version 2.7.2 mais pas sur zéphir (la modification n'avait été faite qu'à partir de la version 2.8.0).

La modification suivante est intégrée dans Zéphir à partir de 2.8.0 :

```
--- a/python/zephir/backend/lib_backend.py
+++ b/python/zephir/backend/lib_backend.py
@@ -971,7 +971,10 @@ class Serveur:
     # que ___version___ seraient ajoutées.
     # cf : https://dev-eole.ac-dijon.fr/issues/10548
     if type(conf_orig[var_name]) == dict and 'val' in conf_orig[var_name]:
-         var_data = "{0}:{1}\n".format(var_name, conf_orig[var_name].get('val'))
+         val = conf_orig[var_name].get('val')
+         if isinstance(val, dict):
+             val = {k: v for k, v in sorted(val.items(), key=lambda x: x[0])}
+         var_data = "{0}:{1}\n".format(var_name, val)
         f_var.write(var_data)
     f_var.close()
 fics=[]
```

Avec le couple module enregistré 2.7.2 et Zéphir 2.7.2, il y effectivement une incohérence dans le traitement.

#5 - 11/04/2024 10:30 - Joël Cuissinat

- *Tâche parente #35868 supprimé*

#6 - 11/04/2024 10:32 - Joël Cuissinat

- *Tracker changé de Tâche à Scénario*

- *Echéance mis à 29/03/2024*

- *Release mis à Carnet de produit (Cadoles)*

- *Points de scénarios mis à 2.0*

+1 point pour ajout appel RPC

#7 - 11/04/2024 10:33 - Joël Cuissinat

- Lié à Scénario #35868: Corrections client Zéphir à rétro-porter en 2.7 ajouté

#8 - 11/04/2024 13:29 - Benjamin Bohard

- Assigné à mis à Benjamin Bohard

Fichiers

zephirservice.py

25,4 ko

29/03/2024

Tom Ricci