Comprenez les taqueuses de mémoire de commande de subversion

Contenu

Introduction

Ce document décrit des commandes utiles de subversion (svn) pour la suite de stratégie de Cisco (CPS).

Conditions préalables

Exigences

Cisco recommande que vous ayez la connaissance du système d'exploitation Linux.

[Composants utilisés](#)

Ce document n'est pas des restrictes au logiciel et aux versions de matériel spécifiques.

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est vivant, assurez-vous que vous comprenez l'impact potentiel de n'importe quelle commande.

[Informations générales](#)

La subversion est un référentiel ce code source de pistes. Chaque version d'une configuration est numérotée et enregistrée en historique de référentiel de subversion. Par conséquent, il est possible de retourner à n'importe quelle version d'une configuration. Le builder de stratégie n'a pas une manière de faire ceci par l'intermédiaire de l'interface utilisateur graphique (GUI), mais en employant les outils ligne de commande de subversion, n'importe quelle version de la configuration peut être faite à la révision en cours.

Commandes utiles de Svn

Ce sont des commandes utilisées pour gérer le svn sur pcrfclient01. Ce ne sont pas toutes les

commandes, mais il affiche ceux typiquement utilisés avec des exemples.

Cette commande montre une liste du repos disponible sur le système.

```
svn ls http://pcrfclient01/repos
```

Voici le passage, la configuration et le repos de starhub_configuration_prod.

```
[root@pcrfclient01 ~]# svn ls http://pcrfclient01/repos/
configuration/
run/
starhub_configuration_prod/
[root@pcrfclient01 ~]#
```

Regardez le log pour le passage de repo de svn pour voir la version

```
svn log http://pcrfclient01/repos/run
```

Ceci affiche un log des informations de repo de passage. Vous pouvez voir que c'est r345 et il y a un commentaire.

```
[root@pcrfclient01 ~]# svn log http://pcrfclient01/repos/run
------------------------------------------------------------------------
r345 | broadhop | 2014-12-08 12:22:48 -0700 (Mon, 08 Dec 2014) | 1 line

removed more "-CO3" from sessionmgr name
------------------------------------------------------------------------
```

Regardez le log pour que tout les repos de svn trouve des versions préalables

```
svn log http://pcrfclient01/repos
```

```
[root@pcrfclient01 ~]# svn log http://pcrfclient01/repos | more
------------------------------------------------------------------------
r345 | broadhop | 2014-12-08 12:22:48 -0700 (Mon, 08 Dec 2014) | 1 line

removed more "-CO3" from sessionmgr name
------------------------------------------------------------------------
r344 | broadhop | 2014-12-08 12:22:32 -0700 (Mon, 08 Dec 2014) | 1 line

removed more "-CO3" from sessionmgr name
------------------------------------------------------------------------
r343 | broadhop | 2014-12-08 12:21:59 -0700 (Mon, 08 Dec 2014) | 1 line

removed more "-CO3" from sessionmgr name
------------------------------------------------------------------------
```

Exportez la base de données de svn de référentiel de passage.

```
svn export http://pcrfclient01/repos/run run_config
```

Ceci exporte la base de données s'exécutante actuellement de svn au run_config de répertoire local.

```
[root@pcrfclient01 ~]# svn export http://pcrfclient01/repos/run run_config
A    run_config
A    run_config/Service-default-_nVmEMLW-EeOaLenhDJbTLQ.xmi
A    run_config/ServiceOption-850M_PREM_PP-_PU4DQNXVEeORFc2I8BVpkA.xmi
```

```
[root@pcrfclient01 ~]# ls
rs.init.sh  run_config  tony
[root@pcrfclient01 ~]#
```

Exportez une base de données de svn d'un référentiel autre que le passage.

```
svn export -r 343 http://pcrfclient01/repos/configuration export_config
```

Ceci exporte la base de données r343 du repo de configuration au répertoire local d'export_config.

```
[root@pcrfclient01 ~]# svn export -r 343 http://pcrfclient01/repos/configuration ex
port_config
A    export_config
A    export_config/ServiceOption-default-_nixTcCdEEeGYKLIkCB773Q.xmi
```

```
Exported revision 343.
[root@pcrfclient01 ~]# ls
export_config  rs.init.sh  run_config
[root@pcrfclient01 ~]#
```

Importez une base de données de svn

```
svn import exported_data http://pcrfclient01/repos/configuration_import_12062014 -m 'import
description'
```

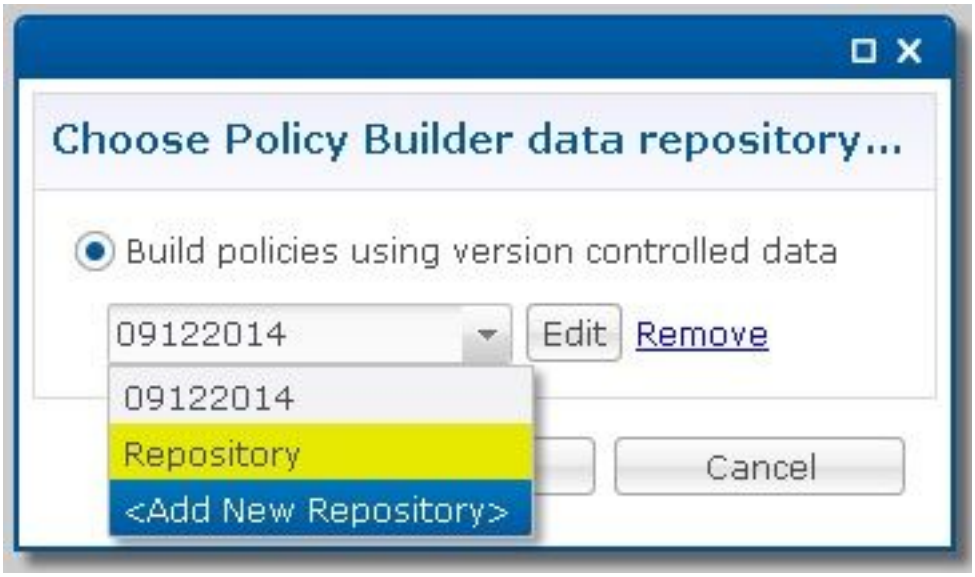
Ceci importe une base de données de svn enregistrée dans l'exported_data de répertoire local et la met dans un repo nommé configuration_import_12062014.

```
[root@pcrfclient01 ~]# svn import export_config http://pcrfclient01/repos/configura
tion_import_12062014 -m 'importing new repo 12062014'
Adding         export_config/ServiceOption-default-_nixTcCdEEeGYKLIkCB773Q.xmi
Adding         export_config/RadiusAAAServerSettings-_siCA4D48EeG2AZ4zmbGbxg.xmi
```

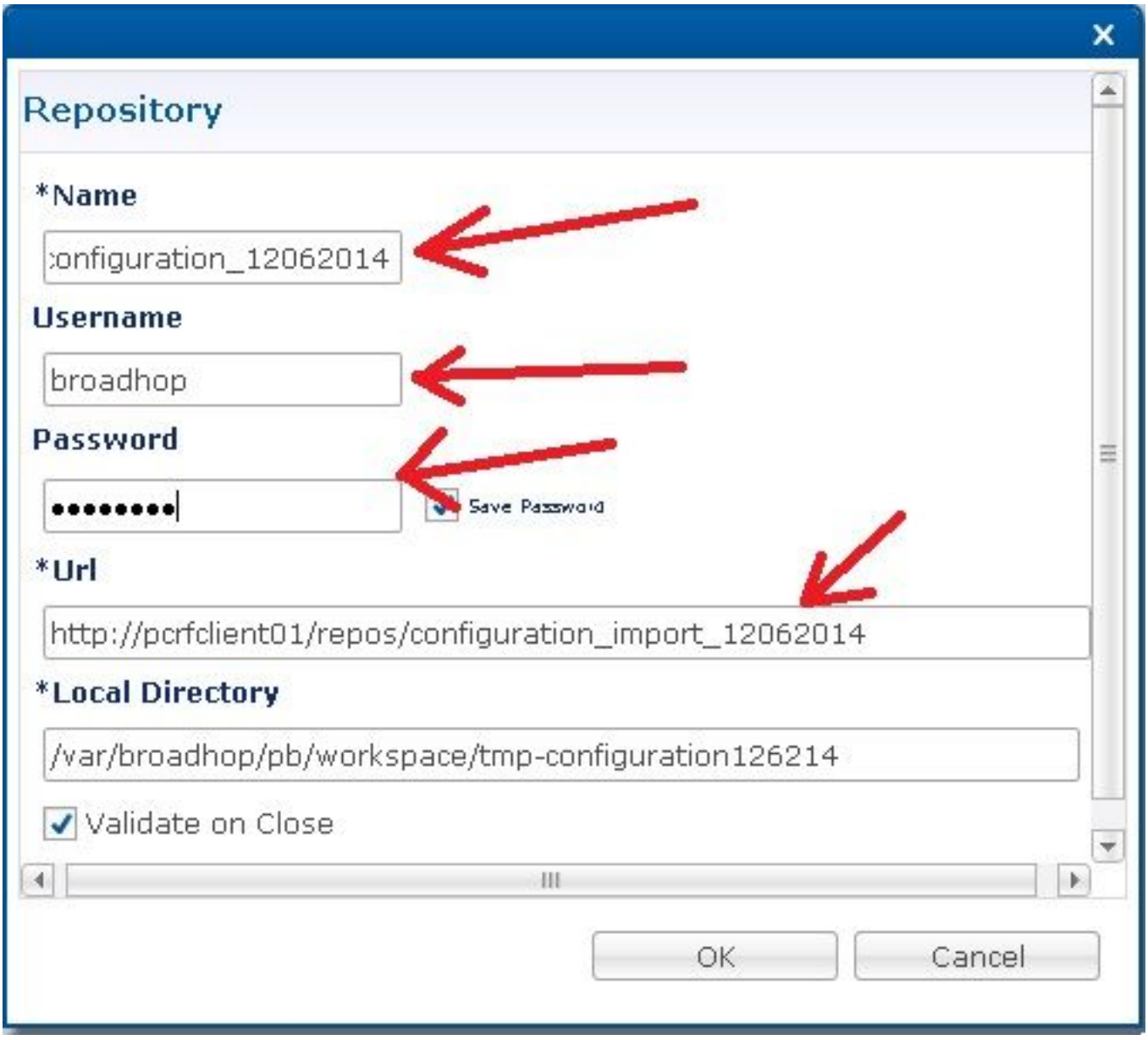
```
Adding         export_config/ConfiguredBlueprint-00819999-70ea-4a36-80f6-2f2287f510
3f-11820.xmi

Committed revision 346.
```

Une fois que la base de données est importée vous pouvez l'éditer et l'utiliser dans le builder de stratégie par ces étapes.

Étape 1. Créez un nouveau repo dans le builder de stratégie.



Étape 2. Changez le nom à quelque chose qui identifie la base de données que vous avez importé, avez placé l'URL au même nom que le répertoire vous a importé la configuration dans et a placé le nom d'utilisateur et mot de passe.



Étape 3. Cliquez ensuite sur **OK**. Maintenant vous pouvez employer le PB pour éditer cet

ensemble de stratégies importées et pour les éditer au CPS.

.