

# Copier un sous-arbre d'un dépôt Subversion vers un autre dépôt

Stéphane Bortzmeyer  
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 7 Mai 2008

<http://www.bortzmeyer.org/copy-subversion-subtree.html>

---

Le VCS Subversion s'est largement imposé comme la référence en matière de VCS centralisé, c'est-à-dire reposant sur un **dépôt** unique de données. Contrairement à son prédécesseur CVS, Subversion permet, par exemple, de déplacer facilement un fichier ou un répertoire à l'intérieur du dépôt, sans perdre l'historique. Mais ce déplacement n'est pas possible entre dépôts différents. Il existe une manipulation pour le faire mais elle nécessite quelques efforts.

Pour Subversion, le monde se réduit à un dépôt à la fois. Si on a un dépôt B (<https://svn.nic.fr/ReD-AFNIC/Etudes> dans l'exemple ci-dessous) et qu'on souhaite y importer des fichiers venus d'un dépôt A (<svn+ssh://bortzmeyer@foobar.example.com/home/bortzmeyer/AFNIC-Subversion> dans l'exemple ci-dessous), **en gardant leur historique** (donc, sans utiliser les fonctions d'importation classiques), la simple copie ne marche pas :

```
% svn copy svn+ssh://bortzmeyer@foobar.example.com/home/bortzmeyer/AFNIC-Subversion \  
    https://svn.nic.fr/ReD-AFNIC/Etudes  
svn: Source and dest appear not to be in the same repository (src: 'svn+ssh://bortzmeyer@foobar.example.com/home
```

La manipulation exacte est possible mais plus complexe. Il faut :

- Copier le dépôt A sous forme d'un fichier texte. Il faut opérer sur le serveur (on accède directement aux fichiers) avec la commande `svnadmin:svnadmindump/home/bortzmeyer/AFNIC-Subversion>svn.dump`
- Filtrer le fichier texte de "*dump*" avec la commande `svndumpfilter` pour ne garder que le sous-arbre qui nous intéresse. Attention, le chemin à donner en paramètre est un chemin absolu dans le dépôt (ici, on garde trois sous-arbres, `anycast`, `EPP` et `ddOS-February-2007`). La commande utilisée est `svndumpfilterinclude/RD/anycast/RD/EPP/RD/ddOS-February-2007<svn.dump>svn-filtered.dump`

- Selon l'arborescence utilisée dans l'ancien et dans le nouveau dépôt, il peut être nécessaire d'éditer le fichier de "dump" pour changer les noms des répertoires (les champs `Node-path` et, s'il y en a des copies, par exemple des branches, les champs `Node-copyfrom-path`). Par exemple, si le fichier "dump" contient `Node-path: IETF/Behave/tests-socket`, et qu'on charge ce fichier dans le répertoire `/ReD-AFNIC`, le fichier résultant sera en `/ReD-AFNIC/IETF/Behave/tests-socket`. Ce changement peut se faire avec la fonction de rechercher / remplacer d'un éditeur ordinaire (les fichiers de "dump" de Subversion sont de simples fichiers texte) ou bien avec un outil comme `sed` ou `perl`.
- De même, il peut être nécessaire de créer le répertoire où sera accroché le sous-arbre copié. Cette création peut se faire via une copie de travail (`mkdir` puis `svncommit`) ou bien directement dans le dépôt avec `svnmkdirURL`.
- Copier le fichier résultant du filtrage sur le nouveau serveur (là encore, il faut pouvoir accéder aux fichiers) avec `svnadmin:svnadminload--parent-dir/ReD-AFNIC/home/Subversion-Repository< dump`. L'option `--parent-dir` permet d'indiquer à quel endroit on accroche le sous-arbre. Cette commande peut échouer partiellement et il est donc prudent de la tenter d'abord sur une copie du dépôt.

Les deux commandes `svnadmin` nécessitent d'avoir les droits pour accéder aux fichiers du dépôt Subversion, en lecture pour la première étape et en écriture pour la seconde. Par exemple, si le dépôt final n'est accessible qu'en HTTP et que les données sont propriétaires de l'utilisateur sous le nom duquel tourne le serveur HTTP (cet utilisateur est `www-data` par défaut sur une Debian), il faudra exécuter la commande sous l'identité de `www-data` par exemple avec `sudo: sudo-uwww-datassvnadminload`.  
...

Si `svnadminload` échoue avec un message du genre :

```
<<< Started new transaction, based on original revision 4
svnadmin: File not found: transaction '92-1', path '/ReD-AFNIC/Etudes/two'
* adding path : ReD-AFNIC/Etudes/two ... %
```

c'est qu'un répertoire manque. Il faut, comme indiqué ci-dessus, changer les noms dans le fichier de "dump" ou bien créer le répertoire. C'est le point le plus délicat de la manipulation.

Une fois copié l'ancien dépôt, il est prudent de faire un `svndelete` sur cet ancien dépôt, pour éviter de continuer à l'utiliser par accident.

Quelques avertissements, enfin :

- Si des fusions ou d'autres opérations qui dépendent du numéro de révision sont présentes dans l'ancien dépôt, elles ne seront pas forcément copiées correctement.
- Les numéros des révisions ne sont pas conservés (puisque'ils sont uniques par dépôt), mais les dates le sont. Résultat, l'ordre des révisions dans le nouveau dépôt n'est plus l'ordre chronologique. Bref, pour résumer, cette manipulation n'est pas idéale. Avec Subversion, il vaut mieux réfléchir avant de mettre les fichiers dans des dépôts séparés, copier entre dépôts restera toujours un pis-aller !

Merci beaucoup à Kevin Grover pour ses explications détaillées sur la manipulation à faire, merci aussi à Les Mikesell pour des détails utiles.

Blair Zajac me suggère qu'on peut aussi utiliser la propriété `svn:externals` de Subversion pour « copier » l'ancien dépôt sans réellement le fusionner (pas tout à fait ce dont j'avais besoin mais cette technique peut être utile dans d'autres cas).

Pour tester différentes idées, j'ai écrit un script shell (en ligne sur <http://www.bortzmeyer.org/files/test-copy-subversion.sh>) qui réalise cette manipulation sur des dépôts « bidons ».