

Introduction à SVN pour les utilisateurs de CVS

Guillaume DUC

`<guillaume.duc@enst-bretagne.fr>`

Département Informatique
École Nationale Supérieure des Télécommunications de Bretagne

22 février 2006

Plan

1 Introduction

2 Utilisation

- Commandes de base
- Propriétés
- Branchements et marquages
- Notes

3 Administration

- Création d'un dépôt
- Configuration de l'accès au dépôt
- Administration de routine
- Hooks

4 Conclusion

Plan

1 Introduction

2 Utilisation

- Commandes de base
- Propriétés
- Branchements et marquages
- Notes

3 Administration

- Création d'un dépôt
- Configuration de l'accès au dépôt
- Administration de routine
- Hooks

4 Conclusion

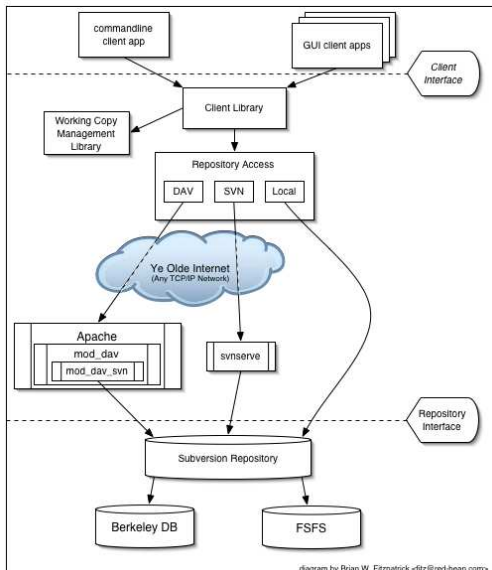
Subversion (SVN)

- Système de gestion de révision
 - Libre (licence de type Apache)
 - Centralisé (contrairement à des systèmes comme Darcs, Arch, *etc.*)
 - Conçu comme un remplaçant à CVS
 - Très répandu (bientôt proposé sur SourceForge, utilisé par le projet KDE, *etc.*)

Améliorations par rapport à CVS

- Suivi des versions pour les répertoires
- Gestion propre des suppressions, copies et déplacements
- Mises à jour atomiques
- Méta-données versionnées
- Plusieurs méthodes d'accès au dépôt (*repository*)
- Gestion automatique des fichiers textes et binaires
- Branchements et marquages (*tags*) efficaces
- Nombreuses opérations en mode déconnecté

Architecture



Méthodes d'accès au dépôt et URL

- Plusieurs méthodes d'accès au dépôt sont disponibles :
 - Local (`file:///path/to/repos`)
 - Protocole svn (`svn://serveur/repos`) potentiellement encapsulé au dessus de SSH (`svn+ssh://serveur/repos`)
 - WebDAV (`http://serveur/repos` ou `https://serveur/repos`)

Plan

1 Introduction

2 Utilisation

- Commandes de base
- Propriétés
- Branchements et marquages
- Notes

3 Administration

- Création d'un dépôt
- Configuration de l'accès au dépôt
- Administration de routine
- Hooks

4 Conclusion

Principes

- SVN est centralisé : un dépôt central stocke toutes les données des différentes versions (comme dans CVS)
- Un développeur récupère une copie de travail (`checkout`), effectue des modifications, récupère les mises à jour faites par les autres développeurs (`update`) puis envoie ses propres modifications au dépôt (`commit`)

Plan

- 1 Introduction
- 2 Utilisation
 - **Commandes de base**
 - Propriétés
 - Branchements et marquages
 - Notes
- 3 Administration
 - Création d'un dépôt
 - Configuration de l'accès au dépôt
 - Administration de routine
 - Hooks
- 4 Conclusion

checkout (co)

- `svn co [-r REV] URL [PATH]`
- Récupère une révision donnée (par défaut la dernière) du dépôt et crée une copie de travail locale
- `REV` : révision à récupérer (numéro, date, etc.)
- `URL` : URL d'accès au dépôt (possibilité de ne récupérer qu'une partie du dépôt)
- `PATH` : chemin de stockage de la copie locale (par défaut un sous répertoire au nom du dépôt)

update

- `svn update [PATH]`
- Met à jour la partie de la copie de travail situé sous le répertoire actuel (ou passé en argument) par rapport à la dernière révision stockée dans le dépôt
- Résultats : A (Added), D (Deleted), U (Updated), G (Merged), C (Conflict)
- Conflits possibles si les modifications apportées à la copie de travail et au dépôt ne sont pas fusionnables (voir traitement des conflits plus loin)

status (hors ligne)

- `svn status [-u] [PATH]`
- Affiche le statut des fichiers et des répertoires de la copie de travail locale par rapport à la dernière révision récupérée
- Première colonne : A (Added), C (Conflicted), D (Deleted), G (Merged), I (Ignored), M (Modified), R (Remplaced), ? (non versionné), ! (manquant)
- Deuxième colonne : Statut des propriétés (C (Conflicted), M (Modified))
- Troisième colonne : Informations de verrouillage
- Quatrième colonne : + (Ajout avec historique)
- `-u` : Interroge le serveur pour connaître le statut de la copie de travail par rapport à la dernière révision du dépôt

commit (ci)

- `svn commit [-m MESSAGE] [PATH]`
- Envoie les modifications faites à la copie de travail vers le dépôt
- `-m MESSAGE` : Message de commit (s'il n'est pas spécifié, un éditeur s'ouvre pour le saisir)
- La commande peut échouer si la version sur laquelle on a travaillé est trop ancienne (update nécessaire avant le commit)

Manipulation sur la copie de travail

- `svn add PATH` : Ajoute un fichier ou un répertoire (l'ajout en question ne se fait que lors du prochain commit)
- `svn delete PATH` : Supprime un fichier ou un répertoire
- `svn copy SRC DST` : Copie un fichier ou un répertoire (SRC et DST peuvent être soit un chemin dans une copie de travail, soit une URL)
- `svn move SRC DST` : Déplace un fichier ou un répertoire
- `svn mkdir PATH` : Crée un répertoire et l'ajoute dans la liste des objets versionnés
- `svn revert PATH` : Annule des modifications non encore committées (modification, ajout, suppression)
- Toutes ces modifications sont pleinement versionnées dans svn (contrairement à CVS)

Résolution des conflits

- En cas de conflit, svn crée trois fichiers temporaires
 - `foo.mine`
 - `foo.rOLDREV`
 - `foo.rNEWREV`
- De plus si le fichier est un fichier texte, il place des marqueurs de conflit (à la CVS) directement dans celui-ci pour indiquer les différences
- Aucun commit ne peut avoir lieu avant la résolution manuelle du conflit
- Une fois le conflit résolu (le résultat doit être placé dans le fichier `foo`) : `svn resolved foo` supprime les fichiers temporaires et indique à svn que tout est bon pour le prochain commit

Récupération d'informations

- `svn log PATH` : Affiche tout l'historique des modifications d'un fichier
- `svn diff PATH` : Affiche les changements locaux apportés au fichier (hors ligne)
- `svn diff -r REV PATH` : Affiche les changements apportés au fichier depuis la révision REV
- `svn diff -r OLD:NEW PATH` : Affiche les changements apportés au fichier entre les révisions OLD et NEW
- `svn blame PATH` : Indique pour chaque ligne d'un fichier, le numéro de la dernière révision où elle a été modifiée et l'auteur de cette dernière modification

Plan

- 1 Introduction
- 2 **Utilisation**
 - Commandes de base
 - **Propriétés**
 - Branchements et marquages
 - Notes
- 3 Administration
 - Création d'un dépôt
 - Configuration de l'accès au dépôt
 - Administration de routine
 - Hooks
- 4 Conclusion

Propriétés

- Une propriété est un couple (PROPNAME, PROPVAL)
- Elles permettent de stocker des meta-données utiles

Manipulation

- `svn propset PROPNAME [PROPVAL|-F VALFILE] PATH` : Définit une propriété sur un fichier ou un répertoire
- `svn propedit PROPNAME PATH` : Édite le contenu d'une propriété
- `svn proplist PATH` : Liste les propriétés définies sur un fichier
- `svn propget PROPNAME PATH` : Récupère la valeur d'une propriété
- `svn propdel PROPNAME PATH` : Supprime une propriété

Propriétés spéciales

- `svn:ignore` : Permet d'ignorer certains fichiers (comme les `.o`, les fichiers de sauvegarde, etc.), certains fichiers étant déjà exclus par défaut
- `svn:keywords` : Permet d'utiliser certains mots clé spéciaux à la CVS (comme `Id`)
- `svn:mime-type` : permet de définir le type MIME du fichier (utilisé pour déterminer si le fichier est binaire ou pas et pour envoyer le bon type au client WebDAV)
- Et d'autres propriétés spéciales qui permettent de contrôler s'il est exécutable ou pas, comment traiter les caractères de fin de ligne, etc.

Plan

1 Introduction

2 Utilisation

- Commandes de base
- Propriétés
- **Branchements et marquages**
- Notes

3 Administration

- Création d'un dépôt
- Configuration de l'accès au dépôt
- Administration de routine
- Hooks

4 Conclusion

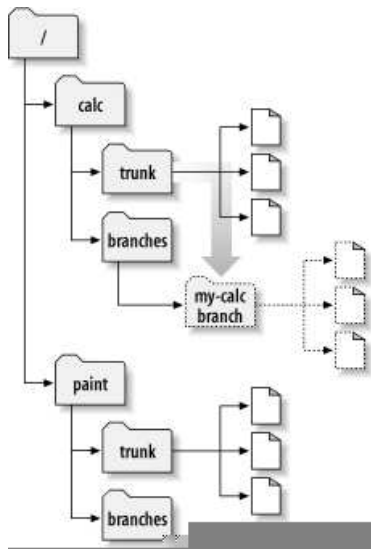
Exemple d'utilisation des branches

- Les développements sont effectués dans le dépôt
- À la sortie de la version 1.0, le développeur crée une branche pour identifier la version
- Les développements se poursuivent dans la branche principale
- Un bug est découvert dans la version 1.0. Le développeur ne peut pas se contenter de le corriger dans la version de développement car celle-ci n'est pas encore prête
- Le développeur corrige le bug dans la branche 1.0, sort une version 1.1 et peut continuer le développement dans la branche principale

Branchements et marquages

- svn n'a pas à proprement parler de concept de branches ou de tags
- Ce ne sont que des copies d'une partie du dépôt à un moment donné
- Il est "recommandé" de créer trois répertoires à la racine du dépôt pour clarifier l'organisation : `trunk` (qui est la branche de développement), `branches` (qui va contenir les différentes branches) et `tags` (qui va contenir les tags)

Exemple



Manipulation

- `svn copy trunk branches/new_branch` : crée une nouvelle branche ou un nouveau tag (fonctionne aussi avec des URL : `svn copy http://serv/repos/trunk http://serv/repos/tags/v1.0`)
- `svn co http://serveur/repos/branches/new_branch` : récupère une branche particulière
- La commande `svn merge` permet de fusionner deux branches (voir la doc...)

Remarque

- L'opération de copie est “gratuite” : la copie physique d'un fichier au niveau du dépôt n'est réalisée que lors de la première modification du fichier original ou de la copie

Plan

1 Introduction

2 Utilisation

- Commandes de base
- Propriétés
- Branchements et marquages
- **Notes**

3 Administration

- Création d'un dépôt
- Configuration de l'accès au dépôt
- Administration de routine
- Hooks

4 Conclusion

Notes

- Pour pouvoir effectuer certaines opérations hors ligne, svn garde une copie de chaque fichier tel qu'il était lors du dernier checkout ou de la dernière mise à jour, ce qui peut engendrer quelques problèmes d'espace disque...
- Afin de ne pas redemander le mot de passe (le cas échéant) à chaque connexion au serveur, svn le stocke (par défaut) dans le répertoire `/.subversion` en clair. Pour éviter ce stockage, il faut utiliser l'argument `--no-auth-cache` à chaque utilisation d'une commande svn utilisant le dépôt

Plan

1 Introduction

2 Utilisation

- Commandes de base
- Propriétés
- Branchements et marquages
- Notes

3 Administration

- Création d'un dépôt
- Configuration de l'accès au dépôt
- Administration de routine
- Hooks

4 Conclusion

Plan

1 Introduction

2 Utilisation

- Commandes de base
- Propriétés
- Branchements et marquages
- Notes

3 Administration

- **Création d'un dépôt**
- Configuration de l'accès au dépôt
- Administration de routine
- Hooks

4 Conclusion

Création d'un repository

- Quelque soit la méthode d'accès qui sera utilisée ultérieurement pour accéder au dépôt :
- `svnadmin create --fs-type FS_TYPE PATH`
- Crée un repository dans le répertoire indiqué
- Il convient ensuite de configurer l'accès à ce dépôt en fonction du mode d'accès choisi (local, svnserve, Apache)

Type de dépôt

- Berkley DB (bdb)
 - Scalable
 - Checkout de la dernière révision rapide
 - Plus mûr
- FSFS (fsfs)
 - Moins sensibles aux problèmes (corruption de la BDB possible dans certains cas)
 - Stockage indépendant de la plate-forme
 - Utilisable au dessus de NFS
 - Va devenir le choix par défaut

Plan

1 Introduction

2 Utilisation

- Commandes de base
- Propriétés
- Branchements et marquages
- Notes

3 Administration

- Création d'un dépôt
- **Configuration de l'accès au dépôt**
- Administration de routine
- Hooks

4 Conclusion

Accès local

- URL de la forme : `file:///path/to/repos`
- Pour l'accès local, le client doit avoir accès directement au système de fichier contenant le dépôt
- Le contrôle d'accès se fait uniquement via les mécanismes de protection sur le système de fichier (droits Unix classiques ou ACL sur les fichiers du dépôt)
- Plusieurs problèmes :
 - Lecture seule théoriquement impossible (des fichiers de verrouillage et de transaction doivent être créés)
 - Problème de l'umask si plusieurs personnes travaillent sur un dépôt commun (ce qui est souvent le cas...)

svnserve + SSH

- URL de la forme : `svn+ssh://serveur/path/repos`
- Pas de serveur particulier (hormis le serveur SSH)
- Les clients doivent disposer d'un compte sur la machine hébergeant le dépôt (ils doivent au minimum pouvoir exécuter la commande `svnserve`)
- Le contrôle d'accès se fait de la même façon que pour l'accès local, *i.e.* autorisations gérées par le système de fichier, d'où les mêmes problèmes que pour l'accès local

svnserve (tout seul)

- URL de la forme : `svn://serveur/path/repos`
- Serveur léger (équivalent de pserve pour CVS) permettant d'accéder au dépôt à distance via le réseau
- Protocole en clair...
- L'authentification est assurée par svnserve grâce à un fichier de configuration contenant des noms d'utilisateur et des mots de passe
- Les autorisations sont également gérées par un fichier de configuration propre à svnserve (droits : lecture ou lecture et écriture sur tout le dépôt)

WebDAV — Présentation

- URL de la forme : `http://serveur/repos`
- Utilise le protocole WebDAV (RFC 2518) pour gérer les différentes opérations entre le client et le serveur
- Nécessite le serveur Apache 2 (la version 1 ne fonctionne pas avec subversion)
- Les opérations sont gérées par le module Apache `mod_dav_svn`
- Protocole en clair pouvant être encapsulé dans SSL/TLS via les mécanismes standards d'Apache (`https://`)
- Permet l'accès en lecture à la dernière version du dépôt par n'importe quel client WebDAV (Firefox, IE, Nautilus, *etc.*), les dernières versions permettant même de faire de l'auto-commit pour permettre des accès en écriture. . .

WebDAV — Présentation

- L'authentification est assurée par les mécanismes d'authentification d'Apache (offre donc de très nombreuses possibilités : fichiers de mots de passe, base de donnée, LDAP, *etc.*)
- Les autorisations sont gérées :
 - soit à un grain très gros directement par Apache via la directive `limit` (lecture/écriture, lecture seule ou rien sur tout le dépôt)
 - soit à un grain très fin via le module `mod_authz_svn` qui, grâce à un fichier de configuration permet de définir des droits (lecture seule ou lecture/écriture) par utilisateur pour chaque répertoire au sein d'un dépôt

WebDAV — Présentation

- Les fichiers du dépôt doivent pouvoir être accessibles en écriture par l'utilisateur faisant tourner le serveur Apache puisque c'est sous l'identité de ce dernier que les opérations vont être effectuées au niveau du système de fichier

WebDAV — Exemple

```
<Location /repos>  
  SetOutputFilter DEFLATE  
  
  DAV svn  
  SVNPath /path/to/repos  
  
  AuthType Basic  
  AuthName "Subversion repository"  
  AuthUserFile /path/to/htpasswd  
  Require valid-user  
</Location>
```

WebDAV — Exemple 2

```
<Location /repos>
  SetOutputFilter DEFLATE

  DAV svn
  SVNPath /path/to/repos

  AuthType Basic
  AuthName "Subversion repository"
  AuthUserFile /path/to/htpasswd

  <LimitExcept GET PROPFIND OPTIONS REPORT>
    Require valid-user
  </LimitExcept>
</Location>
```

WebDAV — Exemple 3

```
<Location /repos>
  SetOutputFilter DEFLATE

  DAV svn
  SVNPath /path/to/repos

  AuthLDAPUrl ldaps://ldap.enstb.org/ou=people,o=RIRE,dc=enst-bretagne,dc=fr?uid
  AuthType Basic
  AuthName "Subversion repository"
  require group cn=svn-gduc,ou=groups,o=RIRE,dc=enst-bretagne,dc=fr
  AuthLDAPGroupAttribute memberUid
  AuthLDAPGroupAttributeIsDN off
</Location>
```

Plan

1 Introduction

2 Utilisation

- Commandes de base
- Propriétés
- Branchements et marquages
- Notes

3 Administration

- Création d'un dépôt
- Configuration de l'accès au dépôt
- **Administration de routine**
- Hooks

4 Conclusion

Sauvegarde d'un dépôt

- Attention : Il ne faut pas faire de sauvegarde “simple” à chaud d'un dépôt (comme pour les bases de donnée. . .) sous peine d'avoir une sauvegarde incohérente
- Utiliser à la place le script `hot-backup.py` fourni avec subversion (il faut adapter le script à ses besoins)
- Ce script utilise la commande `svnadmin hotcopy`

Quelques commandes d'administration

- `svnadmin recover /path/to/repos` : Répare le dépôt dans le cas d'une transaction non achevée (uniquement pour le format Berkley DB)
- `svnadmin dump /path/to/repos` : Dump le contenu du dépôt (par défaut toutes les révisions)
- `svnadmin load /path/to/repos` : Remplit un dépôt à partir d'un dump

Plan

1 Introduction

2 Utilisation

- Commandes de base
- Propriétés
- Branchements et marquages
- Notes

3 Administration

- Création d'un dépôt
- Configuration de l'accès au dépôt
- Administration de routine
- **Hooks**

4 Conclusion

Hooks

- Scripts situés dans le répertoire `hooks` du dépôt
- Appelés automatiquement à plusieurs moments de la vie d'un dépôt
 - `start-commit` : Avant le début d'une transaction de commit
 - `pre-commit` : Avant un commit (Autorisations d'accès)
 - `post-commit` : Après le commit (Envoi des modifications par mail, sauvegarde à chaud, *etc.*)
 - `pre-revprop-change` : Avant une modification de propriété
 - `post-revprop-change` : Après une modification de propriété

Plan

1 Introduction

2 Utilisation

- Commandes de base
- Propriétés
- Branchements et marquages
- Notes

3 Administration

- Création d'un dépôt
- Configuration de l'accès au dépôt
- Administration de routine
- Hooks

4 Conclusion

Résumé des différences avec CVS

- Numéros de révision globaux
- Gestion du déplacement et des copies de fichiers ou répertoires
- Nombreuses opérations disponibles en mode déconnecté
- Gestion des branches et des tags différente
- Méta-données
- Résolution des conflits
- Gestion automatiques des fichiers binaires

Clients

- psvn (http://xsteve.nit.at/prg/vc_svn/) : Interface svn pour Emacs
- TortoiseSVN (<http://tortoisesvn.tigris.org/>) : Client svn pour Windows, comparable à TortoiseCVS
- Subclipse (<http://subclipse.tigris.org/>) : Plugin svn pour Eclipse

Outils

- `cvs2svn` (<http://cvs2svn.tigris.org/>) : Permet de convertir un dépôt CVS en un dépôt SVN (peut conserver tout l'historique d'un projet, ses branches, ses tags, etc.). Voir la doc :-)
- API et de nombreux bindings (notamment en Python) disponibles pour développer facilement des logiciels autour de subversion

Outils

- Trac (<http://www.edgewall.com/trac/>) : Application Web de gestion de projet logiciel minimaliste
 - Interface à Subversion
 - Gestionnaire de tickets
 - WiKi
 - Timeline

Outils

- GForge (<http://gforge.org>) :
 - Dépôt CVS ou Subversion
 - Contrôle d'accès (RBAC)
 - Listes de diffusion
 - Forums
 - Gestion de tickets
 - Gestionnaire de documents
 - Wiki

Références

- Site web de Subversion : <http://subversion.tigris.org/>
- Version Control with Subversion : <http://svnbook.red-bean.com/>
Excellent livre disponible en ligne sur tous les aspects de l'utilisation et de l'administration de subversion (contient tout ce qui est dans cette présentation et tout ce qui n'y est pas...)

Fin

Merci de votre attention

Des questions ?