

Gestionnaire de versions (Introduction à Subversion)

Laurent Réveillère

Enseirb
Département Télécommunications

`reveillere@enseirb.fr`
`http://uuu.enseirb.fr/~reveille/`

Objectifs de Subversion (svn)

- ◆ Garder un historique des différentes versions des fichiers d'un projet
- ◆ Permettre le retour à une version antérieure quelconque
- ◆ Garder un historique des modifications avec leur nature, leur date, leur auteur, etc.
- ◆ Permettre un accès souple en local ou via un réseau
- ◆ Permettre à des utilisateurs distincts et souvent distants de travailler ensemble sur les mêmes fichiers
- ◆ Et encore beaucoup d'autres choses ...

Principaux clients svn

- ◆ Ligne de commande
 - rustique mais permet de tout faire
- ◆ TortoiseSVN
 - <http://tortoisesvn.tigris.org/>, Windows
- ◆ Subclipse
 - <http://subclipse.tigris.org/>, *plugin* pour Eclipse
- ◆ AnkhSVN
 - <http://ankhsvn.tigris.org/>, *plugin* pour Visual Studio .Net
- ◆ psvn.el
 - http://xsteve.nit.at/prg/vc_svn/, mode pour Emacs
- ◆ etc, etc.

Dépôt (*repository*)

- ◆ Emplacement central où sont stockées toutes les données
- ◆ Plusieurs méthodes d'accès
 - ❑ Local (`file:///path/to/repos`)
 - ❑ Protocole svn (`svn://serveur/repos`)
 - ❑ Protocole svn au dessus de SSH (`svn+ssh://serveur/repos`)
 - ❑ WebDAV (`http://serveur/repos`) ou https
- ◆ Contient
 - ❑ Historique des versions des fichiers stockés
 - ❑ Logs enregistrés lors des modifications
 - ❑ Dates et auteurs des modifications
 - ❑ Etc.

Copie de travail (*working directory*)

- ◆ Répertoire situé en local sur le poste de l'utilisateur
- ◆ Contient une copie d'une révision donnée des fichiers du dépôt
 - ❑ Sert de base de travail
 - ❑ Modifications faites en local
 - ❑ Sauvegarde vers le dépôt par l'utilisateur

- ◆ Révision
 - ❑ Modification faite au dépôt
 - ❑ Numéro unique pour chaque révision
 - » Commence à 1
 - » Augmente de 1 à chaque opération
 - » Sa valeur n'a aucune importance
 - » Permet de revenir à une version donnée d'un ou plusieurs fichiers

Administration

- ◆ Création d'un dépôt : `svnadmin create`
 - ❑ Crée un dépôt dans le répertoire courant
 - ❑ Indifférent de la méthode d'accès utilisée ultérieurement
 - ❑ Accès local et `svnserve+ssh`
 - » Contrôle d'accès via le système de fichier
 - ◆ Droits Unix classique
 - ◆ ACL sur les fichiers
- ◆ Exemple

```
matrix: svnadmin create repos
matrix: partage -r -e -a ferrari -a pallard repos
matrix: getfacl repos
user::rwx
user:pallard:rwx           #effective:rwx
user:ferrari:rwx          #effective:rwx
user:veille:rwx           #effective:rwx
group:---                  #effective:---
```

Importer un projet

◆ Objectif

- ❑ Placer dans le dépôt des fichiers locaux déjà existants pour y créer un nouveau projet

◆ Exemple

```
reunion: svn import -m "" hello
      svn+ssh://reveille@ssh.enseirb.fr/net/ens/reveille/tmp/repos
Adding      hello/hello.c
Adding      hello/main.c
Adding      hello/hello.h
Adding      hello/Makefile

Committed revision 1.
```

Principales opérations (1/3)

checkout

- ❑ Récupérer pour la première fois les fichiers déjà existant du dépôt
- ❑ Résultat → copie de travail

◆ Exemple

```
reunion: svn co
          svn+ssh://reveille@ssh.enseirb.fr/PATH/repos
          mylocaldir
A      mylocaldir/hello.c
A      mylocaldir/main.c
A      mylocaldir/hello.h
A      mylocaldir/Makefile
Checked out revision 1.
```

Principales opérations (1/3)

commit

- ❑ Mettre à jour le dépôt à partir de la copie de travail locale
- ❑ Création d'une nouvelle révision
- ❑ Remarques:
 - » La copie de travail correspond à la dernière version du dépôt
 - » Sinon : update → résolution des conflits → commit

◆ Exemple

```
reunion: echo '// Pour voir' >> mylocaldir/hello.c
reunion: svn commit -m 'Petite modif!' mylocaldir
Sending          mylocaldir/hello.c
Transmitting file data .
Committed revision 2.
```

Principales opérations (1/3)

update

- ❑ Opération inverse d'un `commit`
- ❑ Synchroniser la copie de travail locale avec le dépôt en récupérant la dernière version des fichiers du dépôt

◆ Exemple

```
reunion: svn update mylocaldir
```

```
At revision 2.
```

```
[ Modification de dépôt par un tiers ]
```

```
reunion: svn update mylocaldir
```

```
U    mylocaldir/hello.h
```

```
Updated to revision 3.
```

```
reunion: svn update mylocaldir
```

```
At revision 3.
```

Récupération d'une version antérieure

- ◆ Récupération de la dernière version d'un fichier
- ◆ Exemple

```
reunion: echo >> hello.c
reunion: svn revert hello.c
Reverted 'hello.c'
```

- ◆ Revenir à une version située dans le dépôt
- ◆ Exemple

```
reunion: svn update -r 2 hello.h
U    hello.h
Updated to revision 2.
```

Gestion des fichiers

◆ Ajout d'un fichier

- Effectif au prochain `commit`

```
reunion: svn add README.txt
```

```
A README.txt
```

```
reunion: svn commit -m "Ajout du fichier readme"
```

```
Adding README.txt
```

```
Transmitting file data .
```

```
Committed revision 4.
```

◆ Suppression d'un fichier

- Effectif au prochain `commit`

```
reunion: svn delete Makefile
```

```
D Makefile
```

Gestion des fichiers (suite)

◆ Renommer un fichier

- Effectif au prochain `commit`

```
reunion: svn move README.txt lisezmoi.txt
```

```
A lisezmoi.txt
```

```
D README.txt
```

```
reunion: svn commit -m "Renommage d'un fichier"
```

```
Deleting          README.txt
```

```
Adding           lisezmoi.tx
```

```
Committed revision 5.
```

Modifications concurrentes

◆ Quand?

- ❑ Modifications faites à la fois dans la copie de travail et dans le dépôt

◆ Exemple

- ❑ Vous modifiez en local un fichier
- ❑ Un autre utilisateur `commit` ses propre modifications

```
reunion: svn commit -m " " hello.h
```

```
Sending          hello.h
```

```
Transmitting file data .svn: Commit failed (details follow):
```

```
svn: Out of date: 'hello.h' in transaction '7-1'
```

Gestion des conflits

- ◆ Mise en concurrence des deux versions
- ◆ Deux cas de figure
 - Modifications concernant des parties différentes d'un fichier
 - Résolution automatique par `svn`

```
reunion: svn update
```

```
G    hello.h
```

```
Updated to revision 7.
```

- Fusion automatique impossible

```
reunion: svn update
```

```
C    hello.h
```

```
Updated to revision 7.
```

Informations sur les conflits

◆ Nouveaux fichiers dans la copie locale

`hello.h.mine`

- ❑ Copie du fichier tel qu'il se trouvait dans votre copie de travail, en local, avant de faire l'*update*.

`hello.h.r6`

- ❑ Version du fichier pour la révision 6
 - » Correspond à votre dernier *update*
 - » C'est la version qui a servi de base pour les deux utilisateurs du dépôt qui ont travaillé en parallèle

`hello.h.r7`

- ❑ Version du fichier pour la revision 7
 - » Version actuellement dans le dépôt
 - » Version modifiée par un autre utilisateur

`hello.h`

- ❑ Version qui fait apparaître les différences entre versions au sein d'un seul fichier

Résoudre les conflits

- ◆ Édition du fichier `hello.h` pour résoudre les conflits
- ◆ Signaler que le conflit est résolu
 - Effectif au prochain `commit`

```
reunion: svn resolved hello.h
```

```
Resolved conflicted state of 'hello.h'
```

Autre fonctionnalités

- ◆ Notions de branches

- ◆ Informations sur la copie de travail ou sur un fichier

 - `svn info`

- ◆ Historique des modifications

 - `svn log`

- ◆ Statut de la copie de travail

 - `svn status -v`

- ◆ Afficher le contenu du dépôt à distance

 - `svn list svn+ssh://veille@ssh.enseirb.fr/PATH/repos`

- ◆ Propriétés

 - `svn propset svn:ignore *~`

 - `svn propset svn:keyword "Id" *.[ch]`

Interface utilisateur (TortoiseSVN)

