

# Utilisation de subversion (gestion de versions)

Xavier Crégut `Prenom.Nom@enseeiht.fr`

6 janvier 2017

## Installation de subversion (SVN)

```
sudo apt-get install subversion
```

## Création d'une copie locale du dépôt (checkout)

```
svn co http://cregut.svn.enseeiht.fr/2017/dafs/$USER
```

**Attention :** \$USER est à remplacer par votre identifiant ENSEEIHT, celui qui sert à se connecter sur Moodle, sur le portail captif pour accéder au réseau, etc.

Cette commande crée une copie locale du dépôt SVN en créant le dossier \$USER. Faire ls pour le constater.

**Remarque :** On peut ajouter à la fin de la commande un nom de dossier qui deviendra la copie local du dépôt. Le fichier peut exister.

## Ajouter des fichiers

Normalement, on ne met sur SVN que les fichiers sources et pas les fichiers qui peut être obtenus à partir d'autres.

Quand on veut ajouter un nouveau fichier, il faut le dire explicitement :

```
svn add fichier...
```

fichier... sera alors marqué comme à ajouter. Il n'est cependant encore que dans la copie locale du dépôt.

## Supprimer un fichier (remove ou rm)

```
svn remove fichier...
```

Les fichiers seront supprimés localement et marqués comme devant être supprimés du dépôt.

## Renommer un fichier (rename, ren, move, mv)

```
svn mv f1 f2
```

Renomme le fichier `f1` en `f2`.

## État des fichiers

Pour obtenir l'état de tous les fichiers du dépôt, faire :

```
svn status
```

## Pousser les modifications locales vers le dépôt centralisé

```
svn commit -m 'Un message significatif'
```

Il faut bien sûr fournir un message qui explique les changements ont été faits.

Si on omet l'option `-m`, l'éditeur de texte (vi par défaut) sera appelé.

**Remarque** : Si on ne veut pousser que quelques fichiers, il suffit de les indiquer explicitement sur la ligne de commande.

**Attention** : Un *commit* ne peut être fait que si la version locale est à jours vis à vis du serveur. Si d'autres ont poussé leurs modifications, il faut d'abord les intégrer dans la copie locale du dépôt, résoudre les éventuels conflits et ensuite les pousser (voir points suivants).



## Mettre à jour son dépôt local

Pour intégrer les modifications faites sur le serveurs par d'autres dans la copie locale du dépôt, il faut faire :

```
svn update
```

**Remarque :** SVN essaiera de fusionner les modifications s'il y a des modifications faites en local et sur le serveur. S'il n'y arrive pas, il signalera un conflit. La commande propose alors plusieurs choix. Dans ce cas, le plus simple est de choisir 'p' (postpone). Le fichier sera alors marqué en conflit. . .

## Résoudre un conflit

Un fichier en conflit ne peut pas être poussé. Les `svn commit` seront refusés. Il faut d'abord résoudre le conflit.

Il suffit d'éditer le fichier en conflit. SVN a ajouté des marqueurs pour montrer les différences entre la version locale du fichier (mine) et la version distante (identifiée par son numéro de version).

```
<<<<<<< .mine
    message = "bonjour";
=====
    message = "salut";
>>>>>>> .r32931
```

Il faut alors faire la fusion manuellement sous l'éditeur de texte. Par exemple, on garde la ligne avec "bonjour" et on supprime les autres :

```
    message = "bonjour";
```

Ceci étant fait, il faut dire à SVN que le conflit est résolu :

```
svn resolved fichier
```

On peut alors le pousser.

## Différences entre la version locale et celle récupérée du dépôt

On peut savoir quelles sont les modifications faites par rapport à la dernière version récupérée du dépôt :

```
svn diff
```

**Attention** : Ceci ne donne pas les modifications par rapport à la dernière version du dépôt !

## Accéder aux différentes versions

Les commandes

```
svn log
```

ou

```
svn log fichier
```

permettent de voir les dernières modifications du dossier courant ou du fichier donné en argument. Pour chaque modifications, le numéro de version, l'auteur, la date, etc. sont données. Le message fourni lors du 'commit' est aussi affiché.

Le numéro de version peut être utilisé comme option (`-rxxx`) de `update`, `checkout`, etc. pour récupérer une version précise, de `diff` pour comparer un le fichier locale avec la version correspondante, etc.

## Conseils

Toujours penser à faire un 'update' avant de commencer à travailler sur ses fichiers pour être sûr de récupérer les dernières modifications qui auraient pu être faites et poussées par d'autres personnes. Dans un gestionnaire de version, on ne met que les fichiers source et pas les fichiers qui peuvent être engendrés automatiquement (.o, exécutables, documentation engendrée, etc.).