

Svn Client

\$Id: svn.fodp 499 2012-05-14 16:37:15Z Damien \$

Damien Ferney

ANF Mathrice - Angers, mai 2012

Qu'est-ce svn (subversion)

➤ Gestion de version =

- Stocker une arborescence de fichiers et répertoires.
- Garder les changements intervenus (time machine) dans le temps
- Archiver des versions, les rescussiter, les examiner
- Travail collaboratif sur ces documents

=> Dépôt de données => SAUVEGARDE de ce dépôt.

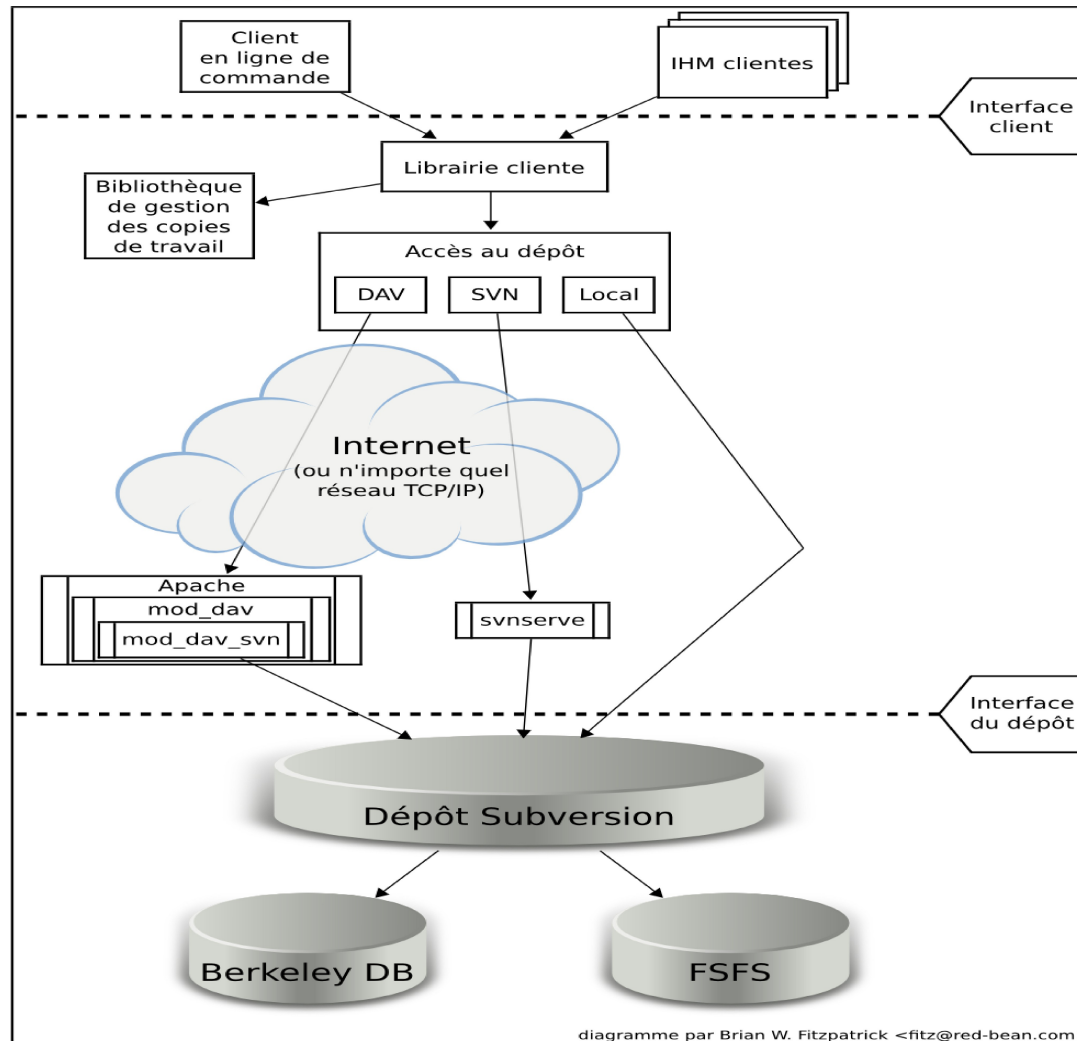
=> Nécessite de réaliser la gestion de l'arborescence avec SVN pour garder la trace des changements.

! si peu de changements => rsync ou unison

!! cumulatif en espace !! **attention** aux fichiers binaires qui changent souvent! (images.....)

Architecture

Date de 2001, depuis 2011 licence Apache et BSD
“apache subversion”



Clients

- ligne de commande : svn
 - (! man svn ne donne pas gd chose)
 - *svn help cmd*
- MacOSX : xsvn , rapidsvn (abandon?)
- Windows : tortoise (intégré)
- Linux : rapidsvn, esvn, smartsvn, kdesvn, nautilussvn, rabbitsvn.

Coté Serveur

- Un peu de config
 - 1 ou plusieurs repo
 - repo =
 - `file:///var/depot/monprojet`, local
 - `http://mon.serveur.fr/monrepo`, apache (mod_dav)
 - `https://mon.serveur.fr/monrepo`, apache+ssl (mod_dav)
 - `svn://mon.serveur.fr/monrepo` svnservice
 - `svn+ssh:// mon.serveur.fr/monrepo` svnservice+tunnel
 - Parametres, authentication, mails commit, postcommit, precommit....
 - Creation d'un projet (repository)

Début

- les commandes sont exécutées sur copie locale par défaut ou sur repo pour certaines.
- Repo vide :
 - une arborescence à mettre dedans : (même vide)
 - `svn import monrep URL/monrepo`
- Version locale d'un repo :
 - `svn (checkout,co) [- r x] URL/monrepo monrep`
- (+ `--depth empty, files, immediates, infinity`)
- `svn help cmd`
- `svn --version` : type d'URL et repo utilisables

Début (suite)

- L'organisation d'un dépôt est libre mais souvent /trunk, /branches, /tags
- .svn dans chaque répertoire de la copie locale (informations/config)
 - Si détruit => plus sous contrôle svn
 - Le recréer = svn update depuis répertoire parent
- Mise en cache du mdp ; -u username
 - Cf fichier de configuration : store-password=no ou enlever dans section auth.
- C'est le serveur qui demande user et password. si c'est propagé de manière anonyme cf la conf du serveur

Utilisation basique

- Edit file `svn add [--depth empty]`
- `svn mkdir dir` (= `mkdir dir; svn add`)
- `svn copy` `svn move`
- `svn (remove,rm,del,delete)`
- `svn status`
 - A append ; D :deleted
 - C conflit ; M : modify
- `svn info`
- `svn (commit,ci) -m "mes"`
- `svn (update,up)`

Consulter les modifications

- **A chaque commit** : incrémentation de révision (cf info)
en fait creation d'une arborescence nouvelle (liens)
- **svn diff** : entre local et repo
- **svn log [-r z] [-v]** : modif entre local/rev z, -verbose
- **svn log [-r x:y] [-v]** : modif entre rev x et y
- **svn diff -r z elt** : diff de elt copie / rev z
- **svn diff -r x:y elt** : diff de elt rev x / rev y
- **svn diff -c x** : diff entre rev x et rev x-1
- **svn cat**
- **svn list [-v] [repo]**
- **svn export**

Changements, révisions et versions(*révisions*)

- Dans la copie de travail :
 - HEAD : dernière; BASE : version locale non modifiée; COMMITTED : rev <= Base dans laquelle un élt a changé ; PREV : Committed - 1
- Rev = {date} : {2012-05-15} => heure = 00:00, {15:30}, {"2012-05-15 15:30"}
- svn update -r *rev*
- svn resolv
- svn revert *file*

Exemples

```
$ svn diff -r PREV:COMMITTED machin.c
```

```
# affiche le dernier changement propagé concernant machin.c
```

```
$ svn log -r HEAD
```

```
# affiche le message associé à la dernière propagation dans le dépôt.
```

```
$ svn diff -r HEAD
```

```
# compare votre copie de travail (avec tous ses changements locaux)
```

```
# à la dernière version de l'arborescence correspondante du dépôt.
```

```
$ svn diff -r BASE:HEAD machin.c
```

```
# compare la version non modifiée localement de machin.c avec la dernière version de machin.c dans le dépôt.
```

Exemples (suite)

```
$ svn log -r BASE:HEAD
```

```
# affiche, pour le répertoire suivi en versions courant, les messages  
# de propagation depuis la dernière mise à jour (svn update).
```

```
$ svn update -r PREV machin.c
```

```
# revient une version en arrière pour le fichier machin.c. Ceci diminue  
# de un la révision de la version de travail du fichier machin.c.
```

```
$ svn diff -r BASE:14 machin.c
```

```
# compare la version non modifiée localement de machin.c avec  
# la version de ce fichier à la révision 14.
```

Résoudre les conflits

- `svn status -u`
- `svn update` :
 - U : updated
 - G : fusionné
 - B : broken (verrou cassé)
 - O : other verrou posé
 - L : locked (fichier de trace peut-etre rejoué avec cleanup)
 - T : verrou volé
 - K : verrouillé par soi
- Il faut avoir configuré un éditeur(vi) : `--- repo; +++ local`
 - p : report (=> fic +.mine,.r1,.r2) mf : my (ma version)
 - df : diff complet
 - e : edit
 - r : résolu (versfusionnée)
 - tf : they (version du depot)
 - l : lance (outil externe)
 - h : help

Résurrection

svn log -v (peut etre dans le rep parent si destruction!)

```
svn copy ../depot/calc/trunk/reel.c@807 ./reel.c
```

```
$ svn status
```

```
A + reel.c
```

```
$ svn commit -m "Ressuscité reel.c à partir de la révision 807,  
/calc/trunk/reel.c."
```

Garde alors l'historique du fichier

Sinon

```
svn cat ../depot/calc/trunk/reel.c@807 > ./reel.c
```

```
$ svn status
```

```
A reel.c
```

Exemples

~/MATHRICE/ANF/anf2012\$ svn info

Chemin : .

URL : <https://svn.math.cnrs.fr/anf2012>

Racine du dépôt : <https://svn.math.cnrs.fr/anf2012>

UUID du dépôt : 0ebb6997-1b51-49bb-9ccb-0d53121da5e9

Révision : 537

Type de nœud : répertoire

Tâche programmée : normale

Auteur de la dernière modification : contensin

Révision de la dernière modification : 537

Date de la dernière modification: 2012-05-18 22:00:00 +0200
(ven. 18 mai 2012)

Exemples

\$svn list

CR-visio-2011-12-14.txt

accueil.fodp

adm/

langages/

organisation-TP.txt

programme.txt

\$svn status -v -u

? langages/SVN

Listes de modifications

- Permet de grouper différents fichiers afin de travailler et associer les commandes uniquement à cette liste. (un fichier ne peut appartenir qu'à une liste à la fois)
- `svn changelist maliste fic1 fic2 fic3 fic4`
- `svn changelist --remove fic3; svn changelist maliste2 fic4`

svn: avertissement : Retrait de 'fic4' de la liste de changements (changelist) 'maliste'.

Le chemin 'fic4' fait maintenant partie de la liste de changement 'maliste2'.

\$ svn status

--- Liste de changements 'maliste' :

fic1

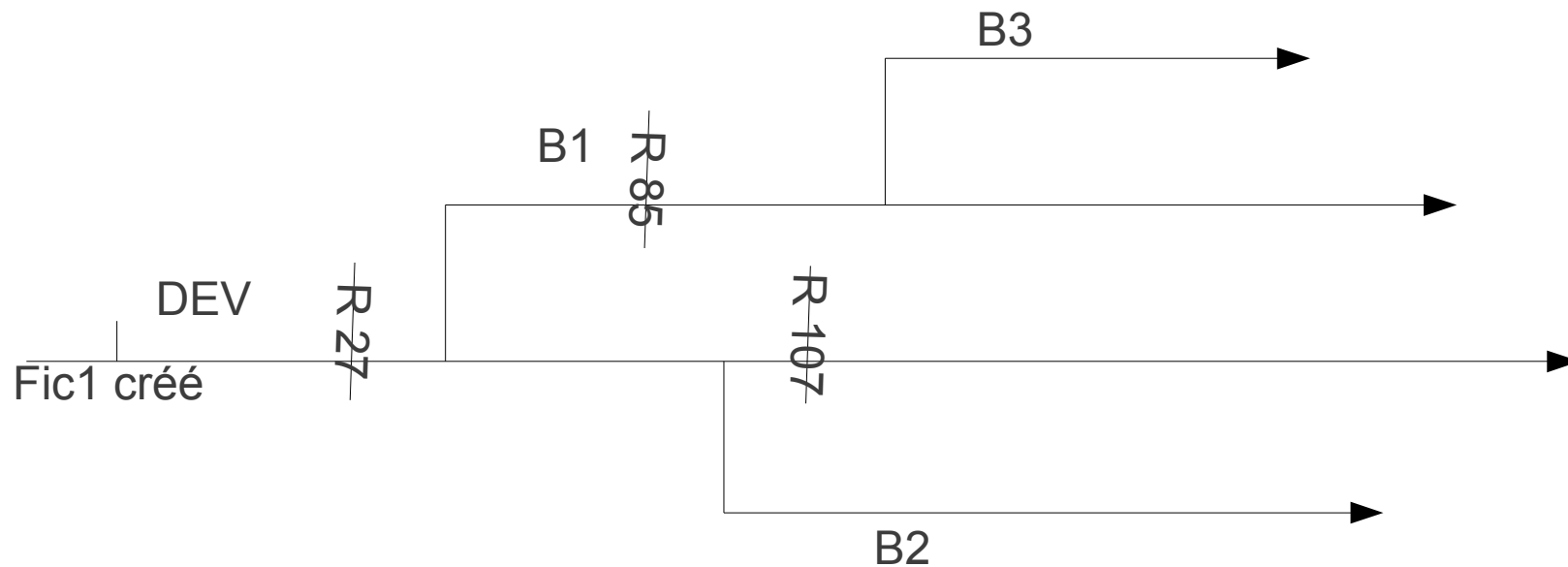
M fic2

--- Liste de changements 'maliste2' :

M fic4

- `svn diff --changelist maliste` `svn ci --keep-changelists`
`svn changelist maliste3 --changeliste maliste`

Les branches



- `svn copy`
`http://svn.exemple.com/depot/proj1/trunk`
`http://svn.exemple.com/depot/proj1/branches/`
`ma-branche-calc -m "Création d'une branche`
`privée à partir de /calc/trunk."`

Exemple de branche

```
$ svn log -v fic1
```

```
r85 | user2 | 2011-11-07 15:27:56 -0600 (jeu. 07 nov. 2011) | 2 lignes
```

```
Chemins modifiés :
```

```
  M /calc/branches/ma-branche-calc/fic1
```

```
* fic1: machiné le bidule.
```

```
r53 | user2 | 2011-11-03 15:27:56 -0600 (jeu. 07 nov. 2011) | 2 lignes
```

```
Chemins modifiés :
```

```
  A /calc/branches/ma-branche-calc (from /calc/trunk:340)
```

```
Création d'une branche privée à partir de /calc/trunk.
```

```
r27 | user1 | 2011-10-29 21:14:35 -0600 (mar. 29 oct. 2011) | 2 lignes
```

```
Chemins modifiés :
```

```
  M /calc/trunk/fic1
```

```
* fic1: modifié une docstring.
```

```
r18 | user1 | 2011-02-22 15:35:29 -0600 (ven. 22 fev. 2011) | 2 lignes
```

```
Chemins modifiés :
```

```
  A /calc/trunk/fic1
```

```
* fic1: ajout du fichier dans ce projet.
```

Gestion de branches

- On a donc 2 vues différentes de l'arborescence avec une partie commune avant copie.
- `cd mabranchelocale; svn merge ...depot/trunk` permet de placer dans ma branche les modifications faites dans trunk. Ainsi `svn status` donnera les fichiers modifiés (à regarder avant de faire `commit!`)
- `Svn ci -m "integre tronc dans ma branche"`

Integration branche → tronc

- Cd ...mon_rep_local (projet ≠ branche)
- svn up (etre sur que l'on est propre!)
- **svn merge --reintegrate ...depot/calc/branches/ma-branche**

=> modif dans la copie locale on a donc des fichiers modifiés -

- **svn ci -m “integre ma branche au projet”**

la branche ne pourra plus servir (reintegrate) il faut en recreer une si necessaire.

- **détruire : svn delete http://...
depot/branche/mabranche -m “fin branche b1”**

- Si on veut repartir dessus : **svn switch URL**
permet d'attacher un depot local existant à une URL

- `svn merge -c -303/depot/calc/trunk/`
Permet de revenir sur la modif effectuée par la rev 303

-

Résumé de branches

Action	Commande
Créer une branche ou une étiquette	svn copy <i>URL1</i> <i>URL2</i>
Faire pointer une copie de travail vers une branche ou une étiquette	svn switch <i>URL</i>
Synchroniser une branche avec le tronc	svn merge <i>trunkURL</i> ; svn commit
Voir l'historique des fusions ou les ensembles de modifications susceptibles d'être fusionnés	svn mergeinfo target --from-source= <i>URL</i>
Fusionner une branche vers le tronc	svn merge --reintegrate <i>branchURL</i> ; svn commit
Fusionner une modification précise	svn merge -c <i>REV URL</i> ; svn commit
Fusionner un ensemble de modifications	svn merge -r <i>REV1:REV2 URL</i> ; svn commit
Empêcher qu'une modification ne soit fusionnée automatiquement	svn merge -c <i>REV --record-only URL</i> ; svn commit
Prévisualiser une fusion	svn merge <i>URL --dry-run</i>
Abandonner une fusion	svn revert -R .
Ressusciter un élément de l'historique	svn copy <i>URL@REV localPATH</i>
Revenir en arrière sur une modification déjà propagée	svn merge -c - <i>REV URL</i> ; svn commit
Examiner l'historique en tenant compte des informations de fusion	svn log -g; svn blame -g
Créer une étiquette à partir d'une copie de travail	svn copy . <i>tagURL</i>
Réorganiser une branche ou une étiquette	svn mv <i>URL1 URL2</i>
Supprimer une branche ou une étiquette	svn rm <i>URL</i>

Propriétés

- `svn propset copyright monfic`
- `svn propset copyright -F ~/malice mon fic`
- `svn propset svn:log '* bouton.c: nouveau message' -rX --revprop ../depot/projet`
- `svn propedit`
- `svn proplist`
- `svn proget`
- `svn propdel`

Les propriétés utiles

- Ignorer certains fichiers (executables , *~ , ...)
svn propedit svn:ignore monrep

- Peux utiliser ?, *, [],

svn propset svn:executable ON un-script

- Substitutions de variables

svn propset Svn:keywords mon fic

- \$Date\$
 - \$Author\$
 - \$Id\$
 - \$Rev\$
 - \$HeadURL\$

\$Id: svn.txt 137 2011-12-16 13:56:42Z ferney \$

Références externes

- Construire une copie à partir de différentes extractions et propager cette architecture (répertoires)
- `svn:externals`

Mais ! Avec les références absolues si le répertoire change de nom ?

– Utiliser

`../ r/r URL rep parent ; ^/ racine du dépôt;`

`// r/r au type d'URL; / r/r à l'URL`

Exemple de référence externe

- `svn propset svn:externals calc`

`^/sons`

`tierce-partie/sons`

`/themes@148`

`tierce-partie/themes`

`//svn.exemple.com/outils-themes@21`

`tierce-partie/themes/outils`

Options valides :

- force : force l'exécution de l'opération
- q [--quiet] : n'affiche rien, ou seulement des informations résumées
- targets ARG : passe le contenu du fichier ARG comme des arguments
- m [--message] ARG : donne le message de propagation ARG
- F [--file] ARG : lit le message de propagation à partir du fichier ARG
- force-log : valide la source de l'entrée du journal
- editor-cmd ARG : utilise ARG comme éditeur externe
- encoding ARG : interprète les caractères comme encodés en ARG
- with-revprop ARG: Définit la propriété de révision ARG pour la nouvelle révision avec le format nom[=valeur]
- keep-local : garder le chemin dans la copie de travail

Options globales :

- `--username ARG` : précise le nom d'utilisateur ARG
- `--password ARG` : précise le mot de passe ARG
- `--no-auth-cache` : ne conserve pas les éléments d'authentification
- `--non-interactive` : pas de demande interactive
- `--trust-server-cert` : accepte les certificats de serveurs SSL inconnus sans confirmation (mais seulement avec '--non-interactive')
- `--config-dir ARG` : fichiers de configuration dans ce répertoire
- `--config-option ARG` : configuration des options utilisateurs avec le format `:FICHER:SECTION:OPTION=[VALEUR]`
Par exemple : `servers:global:http-library=serf`