

# Subversion (en abrégé svn)

## 1 Documentation sur Subversion

Une source d'information et de documentation sur Subversion est disponible en ligne via <http://svnbook.red-bean.com/>. Sous un système d'exploitation de type *Windows Microsoft*® il existe un client Subversion libre de droit implémenté comme une extension au shell *Windows* : TortoiseSVN<sup>1</sup>.

## 2 Créer un dépôt SVN

Examinez le contenu de votre variable d'environnement `EDITOR`. Si cette variable ne contient pas la valeur `emacs` alors rajoutez à votre fichier de configuration `~/.bashrc` la ligne `export EDITOR=emacs`. Sauvegardez votre fichier `.bashrc`, puis dans votre fenêtre `TERMINAL` recharger le fichier de configuration `.bashrc` en invoquant la commande `source ~/.bashrc`.

EXERCICE 1 – Initialisation du dépôt SVN : il s'agit de créer un répertoire où les projets mis sous SVN seront stockés. Le choix pour le dépôt sera `$HOME/svnroot`<sup>2</sup>. Créez votre dépôt. Une fois le dépôt créé, déplacez vous dans celui-ci et explorez les fichiers de cette arborescence. Lisez attentivement le fichier `README.txt`.

## 3 Initialiser et gérer un projet avec SVN

En partant d'un projet vierge, vous allez enchaîner les commandes permettant d'importer un projet dans votre dépôt SVN (option `import`), d'extraire un projet de la base (option `checkout`), d'ajouter un fichier (option `add`), de valider un ensemble de modifications (option `commit`), de consulter l'historique et d'autres informations (options `log` et `status`), de comparer (option `diff`) et finalement de supprimer (option `delete`).

Sortez du répertoire `$HOME/svnroot` et créez un nouveau répertoire (le nom est arbitraire, nous l'appellerons `projet-test`) en dehors de `$HOME/svnroot`. Dans ce répertoire, créez un fichier `essai.txt`. Vous allez initialiser un nouveau projet SVN dont le nom sera `mon-projet-1` et qui reflètera le contenu du répertoire que vous avez créé.

EXERCICE 2 – Utilisez l'option `import` de SVN pour créer le projet `mon-projet-1`. Une fois votre projet créé, vérifiez sa création en consultant l'arborescence de `$HOME/svnroot` et en utilisant l'utilitaire `svnlook`.

EXERCICE 3 – Effacez complètement votre répertoire `projet-test`. Placez vous en dehors du répertoire `$HOME/svnroot`. Vous allez maintenant extraire du dépôt une copie du projet (qui sera placée dans le répertoire courant) via la commande `SVN CHECKOUT` (ou `SVN CO`). Il est nécessaire de connaître l'adresse du dépôt et le nom du projet.

EXERCICE 4 – Consultez le contenu des fichiers de la copie locale de votre projet créée lors de cette exportation initiale.

EXERCICE 5 – Placez vous dans la copie locale de votre projet puis créez en utilisant `emacs` un fichier texte `essai1.txt`. Ajoutez ce fichier au projet `mon-projet-1`. Vérifiez que ce fichier est maintenant programmé pour être ajouté au dépôt.

EXERCICE 6 – Validez l'ajout du fichier `essai1.txt` dans le dépôt SVN (`commit` ou `ci`). L'option `-m` permet d'insérer message de description sans passer par l'éditeur de textes.

EXERCICE 7 – Examinez également le contenu du projet dans votre dépôt SVN afin de voir où et comment est enregistré le fichier `essai1.txt` dans le dépôt.

EXERCICE 8 – Toujours à l'intérieur de votre copie de travail, créez un répertoire `subdirtest`, ajoutez un fichier `essai2.txt` à ce répertoire. Insérez le répertoire `subdirtest` (et le fichier `essai2.txt`) dans le dépôt. Regardez les modifications induites par ces ajouts dans le dépôt.

EXERCICE 9 – Ajoutez un répertoire vide `subdirtest2` directement avec `svn` (sans passer par `mkdir`).

EXERCICE 10 – Consultez l'historique (i.e., les évolutions successives au cours du temps) de votre projet.

EXERCICE 11 – Effacez le fichier `essai1.txt`, consultez le status de votre copie locale afin de vérifier si celle-ci est synchronisée avec le dépôt. Restaurez le fichier manquant depuis le dépôt SVN dans votre copie locale.

1. Cf. <http://tortoisesvn.tigris.org/>

2. Le nom du dépôt est laissé à la discrétion de son créateur.

EXERCICE 12 – Modifiez le contenu du fichier `essai1.txt`. Insérez cette nouvelle version du fichier dans le projet SVN : il n'est pas utile de l'ajouter car il est déjà présent dans le dépôt.

EXERCICE 13 – Consultez l'historique (en mode verbeux) des modifications du fichier `essai1.txt`.

EXERCICE 14 – Affichez les différences entre les deux dernières version du fichier `essai1.txt`, puis entre la dernière version et la version 1.

EXERCICE 15 – Restaurez l'avant dernière version du fichier `essai1.txt`. Vérifiez que cette ancienne version est correctement installée dans votre copie locale.

EXERCICE 16 – Sortez de votre copie locale (mais **ne** vous placez **pas** dans votre dépôt). Créez un nouveau répertoire `test`. Insérez sous le répertoire `test` le contenu courant du projet `mon-projet-1`, vous obtenez ainsi une seconde copie locale du projet. Dans cette seconde copie locale ajoutez un fichier `essai1.1` sous la racine de `mon-projet-1` et ajoutez un fichier `essai2.1` sous le répertoire `subdirtest`. Insérez vos modifications dans le dépôt SVN.

EXERCICE 17 – Remplacez-vous dans le répertoire `$HOME/mon-projet-1`, effectuez la mise à jour de votre projet en fonction du dépôt SVN. Effacez le fichier `essai1.1` du dépôt SVN et refaites la mise à jour sous le répertoire dans la copie `$HOME/test/mon-projet-1` (vérifiez que la suppression est bien faite). Maintenant, effacez le fichier `essai2.txt` mais avant de valider la modification dans la base, faites-le réapparaître.

## 4 Gestion des conflits

Les conflits apparaissent quand les utilisateurs modifient le même fichier dans leurs copies respectives.

EXERCICE 18 – Dans deux copies différentes, modifiez indépendamment le fichier `essai.txt`. Faites `svn status` dans chacune des copies. Dans la première copie, faites un `commit` (`ci`). Dans la deuxième, essayez de faire un `commit`. Que se passe-t-il ? Effectuez un `update` et résoudre le conflit. Explorer les diverses possibilités de résoudre un conflit. Testez des cas où les modifications sont incompatibles et ne peuvent donc pas être fusionnées.

## 5 Travailler à distance avec SVN

À tester sur une machine extérieure au CREMI connectée à Internet.

L'accès à votre dépôt se fera grâce au protocole `svn+ssh` au lieu de `file`.

EXERCICE 19 – Faites une copie du projet, sur votre machine.

EXERCICE 20 – Modifiez le projet. Enregistrez les modifications. La prochaine fois que vous travaillez au CREMI n'oubliez pas de faire un `update`.

## 6 GIT

GIT est un logiciel libre de gestion de versions décentralisé et créé par Linus Torvalds, le créateur du noyau Linux. Une source d'information et de documentation sur GIT est disponible en ligne via <http://git-scm.com/book/fr/>.

EXERCICE 21 – Configurez GIT afin d'avoir la coloration syntaxique et établissez votre nom/email.

EXERCICE 22 – Initialisation du dépôt GIT : il s'agit de créer un répertoire. Le choix pour le dépôt sera `$HOME/git-exemple`<sup>3</sup>. Créez votre dépôt. Une fois le dépôt créé, déplacez-vous dans celui-ci et explorez les fichiers de cette arborescence (`.git/`).

EXERCICE 23 – Placez vous dans votre projet puis créez en utilisant `Emacs` un fichier texte `.gitignore` pour ignorer les fichiers avec extension `.log`. Créez un fichier texte `test.log` et vérifiez que GIT ignore ce fichier. Ajoutez `.gitignore` au projet.

EXERCICE 24 – Validez l'ajout du fichier `.gitignore` dans le dépôt GIT (`commit`). L'option `-m` permet d'insérer message de description sans passer par l'éditeur de textes.

EXERCICE 25 – Créez en un fichier texte `essai1.txt` contenant le résultat de la commande `date`. Ajoutez ce fichier au projet. Vérifiez que ce fichier est maintenant programmé pour être ajouté au dépôt et exécutez le `commit`.

EXERCICE 26 – Consultez l'historique (i.e., les évolutions successives au cours du temps) de votre projet.

EXERCICE 27 – Modifiez le message du dernier `commit` et vérifiez que le changement a bien été effectué en consultant l'historique.

EXERCICE 28 – Effacez le fichier `essai1.txt`, consultez le status de votre copie locale afin de vérifier si elle est synchronisée avec le dépôt. Restaurez le fichier manquant depuis le dépôt GIT dans votre copie de travail.

3. Le nom du dépôt est laissé à la discrétion de son créateur.

EXERCICE 29 – Ajoutez au fichier `essai1.txt` le résultat de la commande `date`. Insérez cette nouvelle version du fichier dans le projet GIT. Lors du `commit`, l'option `-a` permet de considérer uniquement les fichiers modifiés et de ne pas considérer les nouveaux fichiers.

EXERCICE 30 – Consultez l'historique (en mode verbeux) des modifications du fichier `essai1.txt`.

EXERCICE 31 – Affichez les différences entre les deux dernières version du fichier `essai1.txt`, puis entre la dernière version et la version 1.

EXERCICE 32 – Restaurez l'avant dernière version du fichier `essai1.txt`. Vérifiez que cette ancienne version est correctement installée dans votre copie locale.

EXERCICE 33 – Listez les branches existantes, créez et activez une branche avec le nom `devel`.

EXERCICE 34 – Modifiez encore une fois le contenu du fichier `essai1.txt` avec le temps à jour du système. Insérez cette nouvelle version du fichier dans le projet GIT (`commit`). Fusionner la branche `devel` avec la branche principale.

EXERCICE 35 – Créez un dépôt central `$HOME/central/` et insérez `git-example` (`push`) dans ce dépôt. L'option `init --bare` permet de créer un dépôt sans copie de travail, parfait pour être utilisé comme dépôt central.

EXERCICE 36 – Clonez le dépôt central `$HOME/central/` comme `$HOME/git-example2/`. Validez que il est mise a jour (`pull`). Modifiez le contenu du fichier `essai1.txt`. Insérez cette nouvelle version du fichier dans le projet GIT et finalement propagez les modifications vers le dépôt central (`push`).

## 7 Travailler à distance avec GIT

À tester sur une machine extérieure au CREMI connectée à Internet.

L'accès à votre dépôt se fera grâce au protocole `ssh`.

EXERCICE 37 – Clonez le projet depuis le dépôt central dans un répertoire `git-example`.

EXERCICE 38 – Allez dans la copie `git-example` et faites des modifications. Sauvez les modifications localement puis dans le dépôt central.

EXERCICE 39 – Mettez à jour votre copie de travail au CREMI.