
Utiliser la forge de l'UBP pour gérer un projet avec SVN

- Les bases -

Hélène Toussaint, 27 novembre 2012

But du document :

Le but de ce document est de découvrir la gestion de projets à l'aide des outils disponibles sur la forge de l'UBP, en particulier SVN, et de donner les concepts fondamentaux ainsi que quelques notions techniques pour apprendre à les utiliser.

Ce document s'adresse principalement aux développeurs novices en gestion de projet et qui souhaitent un aperçu rapide (et non exhaustif) des possibilités offertes par SVN.

Un grand nombre d'information et de documentations concernant la forge sont accessibles sur <http://forge.clermont-universite.fr/wiki/redmine-clruniv>. Entre autre vous trouverez un guide de démarrage rapide : <http://forge.clermont-universite.fr/documents/9> qui permet de prendre en main un projet à la fois pour les développeurs et les utilisateurs.

Sommaire

1	Création d'un projet	2
1.1	Pourquoi créer un projet sur la forge ?	2
1.2	Que mettre dans un projet ?	2
1.3	Comment créer un projet ?	2
2	Les différentes parties d'un projet	2
3	Le gestionnaire de versions : Subversion	3
3.1	Au cœur du système : le dépôt	3
3.2	Eviter les fichiers concurrents	3
3.3	Concrètement : comment gérer son dépôt ?	4
4	Références	7

1 Création d'un projet

1.1 Pourquoi créer un projet sur la forge ?

Extrait du site <http://forge.clermont-universite.fr/wiki/redmine-clruniv> :

"Une forge est un outil de développement collaboratif de logiciels informatiques. Une forge permet d'intégrer au sein d'un projet une gestion du code source, la gestion des bugs, la documentation et la distribution des releases."

La forge a donc pour but de faciliter le travail de développement collaboratif. Cependant elle peut aussi être très utile pour un travail individuel car elle fournit différents outils de gestion de projet comme le gestionnaire de version SVN et des outils qui permettent d'interagir avec les utilisateurs (demande d'assistance, rapport de bugs...).

1.2 Que mettre dans un projet ?

On peut mettre tous les documents que l'on veut : code source, documentations, images...

1.3 Comment créer un projet ?

1. Connectez vous sur la forge à l'aide de vos identifiant UBP : <http://forge.clermont-universite.fr/> (voir [1] pour plus de détails).
2. Une fois connecté ouvrir la page des projets en cliquant sur "projet" en haut à gauche de l'écran
3. Sur la page des projets cliquer sur "Nouveau projet". Remplir les différents champs demandés et sauvegarder.

2 Les différentes parties d'un projet

Sur la page de votre projet vous avez différents onglets (cf Figure 1).



Figure 1 : la page projet

Signification des onglets :

Aperçu : donne une vue synthétique du projet

Activité : donne une vue synthétique des dernières activités sur le projet

Demandes : donne une vue synthétique des demandes (émanent des utilisateurs) faites sur le projet

Nouvelle demande : permet d'éditer une nouvelle demande

Annonces : donne une vue synthétique des annonces liées au projet

Documents : permet de déposer / télécharger de la documentation liée au projet (par exemple Rapports, articles)

Wiki : permet d'éditer un wiki

Fichiers : permet de déposer / télécharger les fichiers liés à une version du logiciel (en général une version stable au format tar.gz du projet)

Dépôt : donne une vue synthétique du dépôt SVN lié au projet (cf section 3)

Configuration : permet de modifier la configuration du projet : membres, accès...(pour les personnes autorisées)

3 Le gestionnaire de versions : Subversion

Subversion est le gestionnaire de versions disponible sur la forge. Il permet de centraliser, partager et tracer l'historique des différents fichiers d'un projet. Pour une présentation exhaustive de SVN voir la documentation ([2]).

3.1 Au cœur du système : le dépôt

Le dépôt est un serveur particulier : il permet de stocker les données (fichiers, images, documents...) et il est accessible par tous les membres du projet (appelés clients du dépôt, cf Figure 2). Sa particularité vient du fait qu'il conserve un historique de toutes les modifications qui ont eu lieu sur les fichiers depuis sa création.

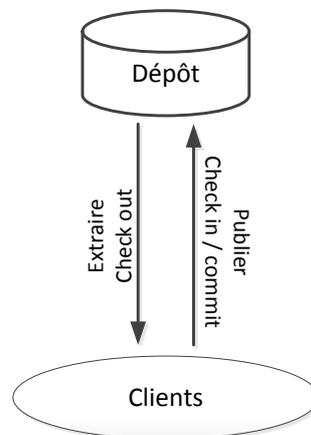


Figure 2 : Interactions avec le dépôt

Les clients peuvent :

- déposer des données sur le dépôt (on utilise généralement le verbe **publier** ou en anglais **commit** ou **check in**)
- récupérer des données sur le dépôt (on utilise généralement le verbe **extraire** ou en anglais **check out**)

3.2 Eviter les fichiers concurrents

Nous appelons "fichiers concurrents" deux (ou plusieurs) versions différentes d'un même fichier. Les fichiers concurrents arrivent lorsque l'on travaille à plusieurs. Par exemple deux développeurs A et B modifient le même fichier source. Le développeur A publie le fichier source modifié sur le dépôt puis le développeur B publie à son tour. On a alors deux versions différentes du même fichier source. Comment SVN gère ce conflit ?

SVN utilise le paradigme "*copier - modifier - fusionner*" : chaque client copie (extrait) les fichiers du dépôt sur sa machine : on appelle cette copie *une copie de travail*. Il les modifie comme il le souhaite en local sur sa machine et les renvoie (publie) sur le dépôt. Si SVN constate qu'un des fichiers avaient déjà été modifié par quelqu'un d'autre il stoppe momentanément la publication et en informe le client. Le client doit alors fusionner les deux fichiers concurrents. Pour cela SVN offre différents outils qui permettent, entre autre, d'examiner les différences entre les fichiers. Une fois la fusion effectuée le client peut déposer (publier) le fichier fusionné sur le dépôt qui sera alors la version courante du fichier. Tout repose donc sur la personne qui a en charge la fusion des fichiers et sur la communication entre clients !

3.3 Concrètement : comment gérer son dépôt ?

*Cette partie s'adresse plus particulièrement aux utilisateurs de Windows puisqu'il y est question de **TortoiseSVN** un client SVN pour Windows. Cependant tous les concepts sont les mêmes pour linux, seul les commandes liées au client sont différentes.*

3.3.1 Initialiser le dépôt

A la création d'un projet sur la forge un dépôt SVN est automatiquement créé. Il est accessible au plus tard une heure après la création du projet.

3.3.2 Télécharger et installer un client SVN : TortoiseSVN

Pour publier et extraire les données du dépôt on a besoin d'un client SVN. Nous allons utiliser dans ce document **TortoiseSVN** qui est un client pour Windows. Nous donnons dans les paragraphes suivants les commandes de base. Pour plus de détails consultez la documentation de référence [3].

Tortoise est une extension de shell : les fonctionnalités de Tortoise apparaissent lorsque vous faites un clic droit sur un dossier ou fichier (cf Figure 3).

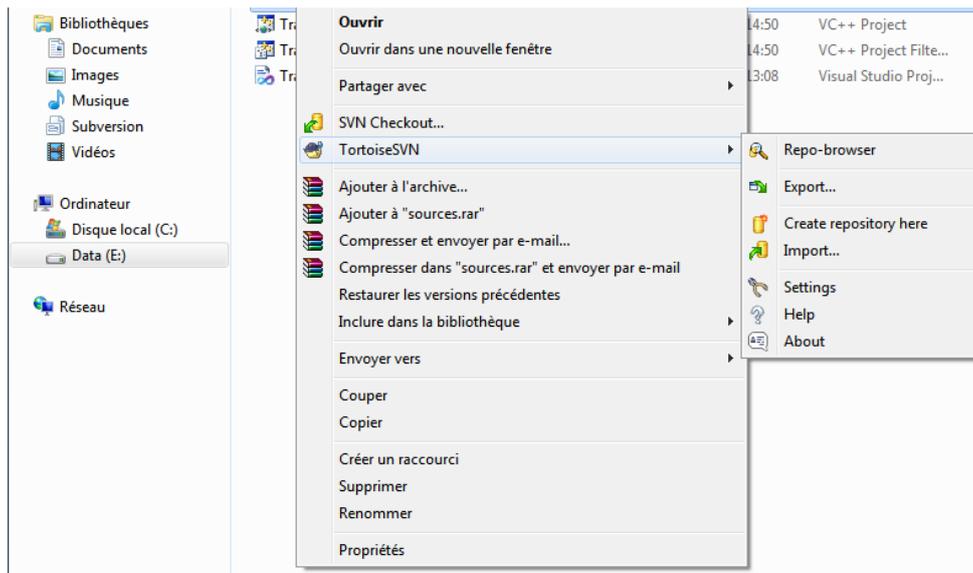


Figure 3 : TortoiseSVN

A partir de maintenant Tortoise a été installé sur votre machine et nous allons utiliser ses fonctionnalités pour "dialoguer" avec le dépôt

3.3.3 Le premier *import*

Il est recommandé de diviser son dépôt en 3 répertoires : trunk, branches et tags (pour une explication détaillée voir le chapitre 4 du manuel de référence [2]). Dans le but de mettre l'accent uniquement sur les commandes principales je ne le fais pas dans cet exemple. Cependant il est indispensable de structurer un minimum son répertoire de travail afin que le dépôt soit également structuré correctement et qu'on puisse s'y retrouver facilement.

Par exemple, je nomme mon répertoire de travail `TransvCover` (nom de mon projet) et je crée des sous répertoires `source`, `doc` et `resultat`. Pour importer le contenu du répertoire local `TransvCover` dans le dépôt faire un clique droit dessus et choisir TortoiseSVN/Import.

Une boîte de dialogue s'ouvre, remplissez le champ "URL" avec l'URL de votre dépôt SVN (le lien est disponible sur la page de votre projet sur la forge, onglet dépôt : il commence par `https://forge.clermont-universite.fr/svn/` suivi de l'identifiant de votre projet) et le champ "message" avec un commentaire décrivant en quoi consiste votre import. Cliquer sur OK. Dans votre projet sur la forge, onglet dépôt, vous voyez maintenant apparaître les sous-répertoires (cf Figure 4).

Attention c'est bien le contenu du dossier qui est importé et non le dossier : si vous faites cela avec un dossier qui contient un grand nombre de fichiers, ces fichiers vont se retrouver directement à la racine de votre dépôt (en vrac). D'où l'importance de bien structurer votre répertoire de travail dès le départ.

The screenshot shows the 'Transversal Cover' repository interface for CLERMONT UNIVERSITÉ. The user is logged in as 'hetoussa'. The main menu includes 'Aperçu', 'Activité', 'Demandes', 'Nouvelle demande', 'Annonces', 'Documents', 'Wiki', 'Fichiers', 'Dépôt', and 'Configuration'. The current view is 'root', showing a table of files with columns for 'Nom', 'Taille', 'Révision', 'Âge', 'Auteur', and 'Commentaire'. Three folders are listed: 'doc', 'resultat', and 'source', all with revision 3 and age 'moins d'une minute'.

Nom	Taille	Révision	Âge	Auteur	Commentaire
doc		3	moins d'une minute	Helene TOUSSAINT	projet avec toutes les fonctions testées et non...
resultat		3	moins d'une minute	Helene TOUSSAINT	projet avec toutes les fonctions testées et non...
source		3	moins d'une minute	Helene TOUSSAINT	projet avec toutes les fonctions testées et non...

Figure 4 contenu du dépôt

3.3.4 Extraire une copie de travail

Le répertoire importé ne devient pas une copie de travail. Pour commencer à travailler avec SVN (c'est-à-dire pour que SVN puisse tracer les changements) il est indispensable de créer une copie de travail sur votre machine. Pour cela créer un répertoire vide, il sera votre répertoire de travail. Faites un clic droit sur ce répertoire puis choisissez *SVN Checkout*. Une fenêtre s'ouvre, elle contient déjà les chemins vers le dépôt et vers le répertoire de travail (sinon remplissez les !). Cliquer sur OK, vous venez de créer une copie de travail. Les fichiers et dossiers sont tous marqués d'un petit icône "check" vert ce qui signifie qu'ils sont à jour par rapport à la version du dépôt.

3.3.5 Travailler sur sa copie de travail

A partir de maintenant vous devez travailler uniquement sur votre copie de travail pour pouvoir répercuter les changements que vous apportez dans le dépôt. Vous pouvez modifier les fichiers textes avec votre éditeur habituel, SVN peut tracer ce genre de modifications sans problème. Par contre vous devez utiliser les outils fournis par Tortoise pour tout ce qui concerne les changements dans l'arborescence : renommage de fichiers, suppression, déplacements, création de nouveaux fichiers / dossiers.

3.3.6 Publier sa copie de travail

Tant que vous ne publiez pas, vos modifications sont locales : elles n'apparaissent pas sur le dépôt et ne sont donc pas "téléchargeables" par les autres utilisateurs. Pour publier vos modifications faites un clic droit sur votre répertoire de travail et choisir *SVN Commit*. Une fenêtre s'ouvre et vous invite à saisir un message qui décrit les changements que vous avez effectué. Cliquez sur OK, vos changements sont alors sauvegardés sur le dépôt. Chaque *SVN Commit* crée une nouvelle *révision* associée à un numéro unique (celui de la révision précédente + 1). Avant de publier vos changements il peut être intéressant de faire le point sur ce que vous avez changé, cela vous aidera à écrire un message de publication adéquat. Pour cela Tortoise dispose d'un outil très pratique : "Tortoise / check for modifications" qui vous permet de voir les modifications entre votre copie de travail

actuelle et la dernière révision en date. Si vous double cliquer sur un fichier qui a été modifié un éditeur de texte s'ouvre et montre les modifications dans le texte.

3.3.7 Ne pas publier certains fichiers présents dans la copie de travail

Vous pouvez avoir envie de ne pas publier certains fichiers car ils ne sont pas utiles dans la vie du projet mais qui sont pourtant dans votre copie de travail. C'est le cas par exemple des fichiers .o / .exe générés par un makefile ou, encore pire si vous travaillez avec une plateforme de développement comme Visual Studio, de tous les fichiers générés automatiquement. Pour cela : clique droit sur le répertoire ou fichier à ignorer et "Tortoise / Unversion and add to ignore list".

3.3.8 Synthèse : une journée de travail type

1. Créer une copie de travail (*SVN Checkout*) ou mettre à jour sa copie de travail (*SVN Update*).
2. Modifier sa copie de travail en prenant soin d'utiliser les outils TortoiseSVN si on modifie l'arborescence
3. Propager (publier) les changements (*SVN commit*).

4 Références

- [1] Antoine Mahul, Quickstart Forge Clermont Université, 2010, <http://forge.clermont-universite.fr/documents/9>.
- [2] Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato, Version Control with Subversion: For Subversion 1.5: (Compiled from Revision 4309), 2008.
- [3] Stefan Küng, Lübbe Onken, Simon Large, TortoiseSVN: Un client Subversion pour Windows: Version 1.7, 2012.