

Mise en oeuvre de la solution SVN

1 OBJECTIFS

Ce TP sert à la compréhension d'un outil de gestion des révisions des fichiers d'une application gérée par plusieurs développeurs. L'outil retenu est SVN, connu aussi sous le nom de "SubVersioN". Ce système propose la gestion de projets d'applications sur un serveur centralisé.

C'est la différence majeure entre SVN et GIT : GIT propose une gestion décentralisée, un peu comme un système de gestion Torrent.

Pour cela, il sera nécessaire d'utiliser un serveur et un client SVN.

Le serveur utilisera WANDisco Subversion, le client sera TortoiseSVN, deux outils fonctionnant sous Microsoft Windows.

2 INSTALLATION SERVEUR

Le choix d'un serveur sous Windows n'est pas le plus judicieux, cependant, pour gagner en temps sur le projet de messagerie instantanée, le choix pédagogique est l'installation du serveur VisualSVN.

2.1 TÉLÉCHARGEMENT

Le site <http://subversiondownload.com/> permet de choisir parmi plusieurs versions de subversion, disponibles gratuitement ou non, compatible sur plusieurs OS mais VisualSVN n'y figure pas : conservez tout de même ce lien pour une éventuelle installation plus réaliste (sous Linux ou Apache en mode avancé).

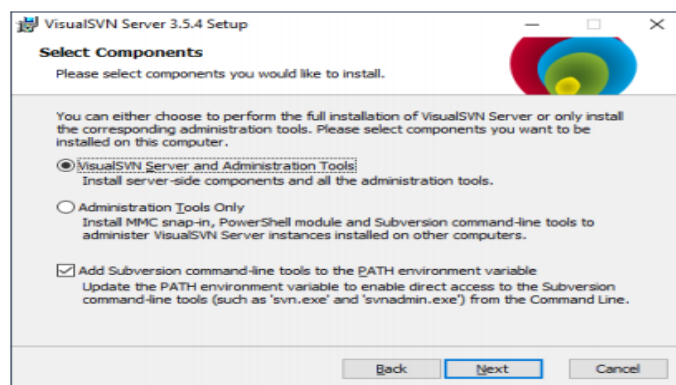
VisualSVN se trouve sur <https://www.visualsvn.com/server/> Cependant, pour les besoins du TP et par égard aux autres utilisateurs, vous utiliserez le serveur local [\\srv1-lab\](https://srv1-lab/)

2.2 INSTALLATION

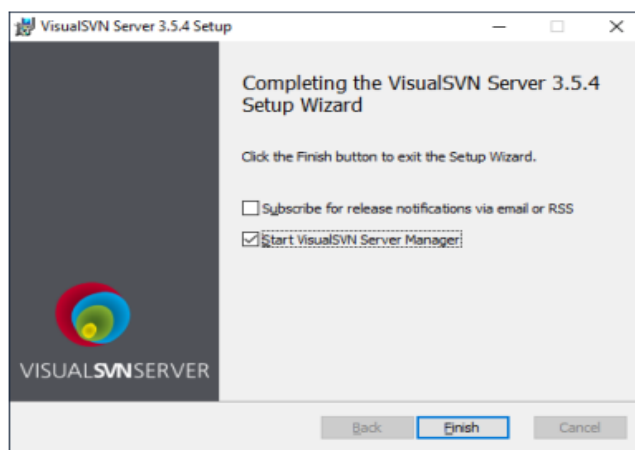
L'installation est relativement simple.

Notez cependant au lancement de l'installateur, la version d'Apache et de SVN : en terme de sécurité, il peut être utile de vérifier ce genre d'informations.

La fenêtre de dialogue indiquant les outils à installer ne sera pas modifiée, mais il faut impérativement laisser coché la case "Add Subversion command-line tools to the PATH environment".

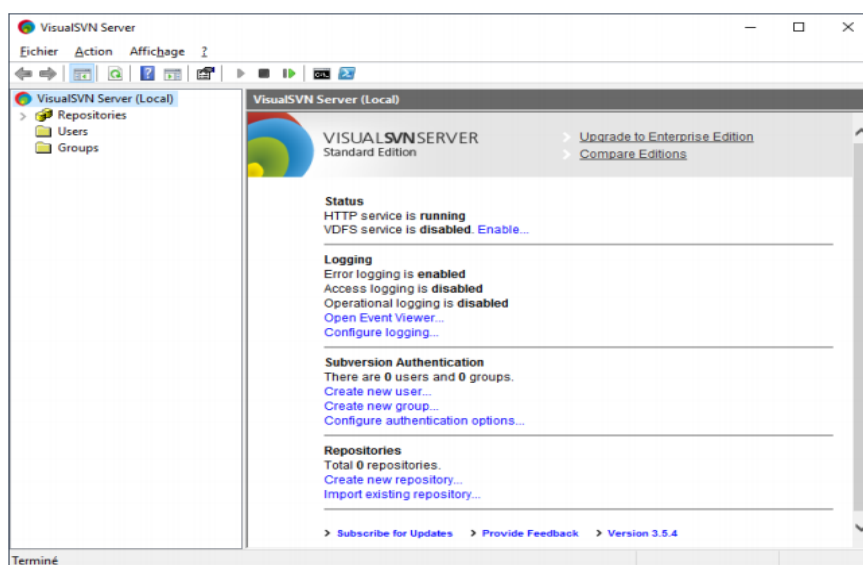


La fenêtre suivante propose deux versions mais la version gratuite est "Standard Edition". Il n'est pas nécessaire de changer la configuration initiale, le répertoire C:\Repositories sera créé et il sera possible de sécuriser les accès au serveur. Enfin, vous pourrez cocher la case "Start VisualSVN Server Manager" pour terminer l'installation.



2.3 CONFIGURATION

L'interface de gestion de VisualSVN s'affiche comme ceci :

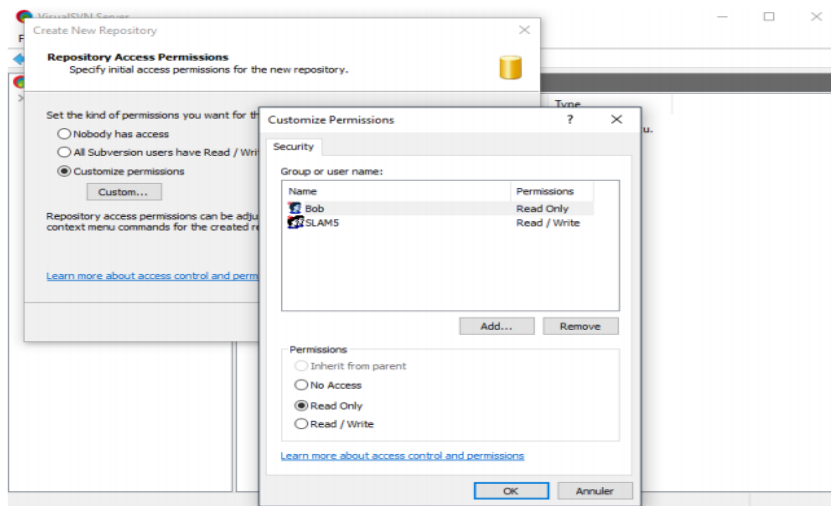


La première action concerne la création de comptes utilisateurs, en effectuant un clic droit sur Users ► Create users...

Créez autant d'utilisateurs qu'il y a dans votre projet et ajoutez un utilisateur Bob ayant pour mot de passe 'bob'. Ensuite, il est intéressant de créer un groupe d'utilisateurs : c'est la même démarche mais sur l'icône "Groups". Appelez-le "SLAM5" par exemple. Ajoutez-y vos utilisateurs, sauf Bob. Enfin, il faut créer un dépôt (repository) : clic droit sur Repositories ► Create new repository...

Sélectionnez Regular FSFS qui est un format standard de suivi des versions puis donnez-lui le nom de votre projet (pour cet exemple, "UnProjet"). Sélectionnez le bouton radio "Single-project repository".

Pour les permissions, choisissez de modifier les permissions comme suit (votre groupe peut lire et écrire, Bob ne peut que lire) :



La dernière fenêtre récapitule la configuration du dépôt : veuillez noter l'URL fournie (Copier).

3 INSTALLATION CLIENT (TORTOISESVN)

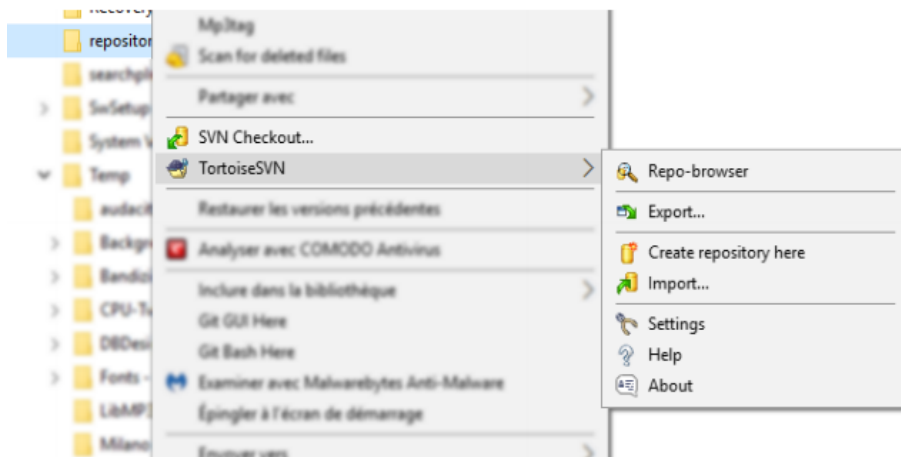
TortoiseSVN est un client intéressant, car il limite l'usage des commandes à écrire. Là encore, l'objectif de ce TP est d'être rapide et concis.

3.1 TÉLÉCHARGEMENT

Vous trouverez TortoiseSVN sur le site sourceforge : <https://sourceforge.net/projects/tortoisetsvn/> Vous pouvez également utiliser la version disponible sur le réseau local pour gagner du temps.

3.2 INSTALLATION

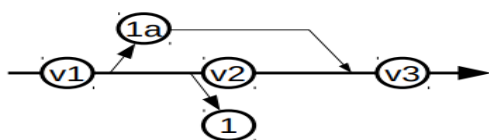
L'installation de TortoiseSVN est très succincte : le programme existe en version x86 ou x64, en fonction de l'OS sur lequel vous l'installez. Une fois l'installation terminée, vous disposez d'un menu contextuel amélioré, dans l'Explorateur Windows.



4 FONCTIONNEMENT SVN

La structure d'un SVN contient un ensemble de révisions de fichiers. Il n'y a pas (en cela) de différences avec un gestionnaire de fichiers classique qui archiverait chaque version de fichiers. Cependant, SVN apporte également une méthodologie de développement qu'il faut intégrer : chaque projet commence avec une arborescence devenue un standard de fait (par opposition à un standard normalisé par un organisme international). Le contenu du répertoire projet contient les sous-répertoires suivants :

- trunk contiendra le code stable représentant la version destinée aux utilisateurs
- branches contiendra les codes pour les nouvelles versions pendant les tests et développements de nouvelles fonctions. Lorsqu'une branche est créée, aucun fichier n'est déplacé et seuls les liens symboliques permettent d'accéder au code de la branche. Dès l'édition d'un code en revanche, le fichier est copié puis modifié dans la branche correspondante.
- Tags contiendra une version donnée des fichiers du trunk mais uniquement une version figée. Même si le trunk avance, la version contenue dans un tag ne sera pas modifiée. Cela permet de conserver une version particulière d'un ensemble de fichiers et par conséquent, une version stable de l'application. Il ne faut donc effectuer aucune modification dans les tags (tag est une sorte d'étiquette "cuvée 2016"). Dans l'exemple ci-après, le trunk est représenté par les chiffres de la version stable v1, v2, v3. Une branche peut-être créée pour résoudre un bug sur le module d'impression de l'application (v1a). Lorsque le bug est résolu, on réintègre les développements de cette branche dans le trunk. Tag permet de conserver une trace des fichiers utilisés dans la version v1 en cas de régression sur des fonctions qui étaient opérationnelles et qui dans la version v2 ne marchent plus.



Dessin 1: Exemple de SVN

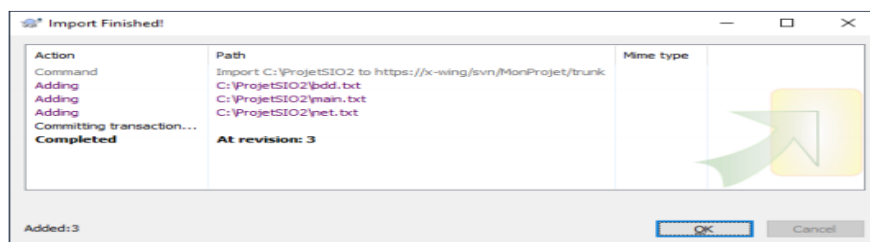
5 UTILISATION SVN

Nous allons maintenant effectuer quelques manipulations de SVN pour comprendre son fonctionnement et son intérêt.

5.1 CRÉATION D'UN PROJET

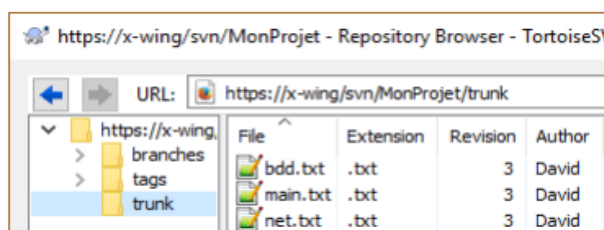
Lors de la configuration de VisualSVN, nous avons créé la structure sur le serveur, il faut maintenant y intégrer votre dossier de travail. Pour le TP, créez le répertoire 'C:\ProjetSIO2'. Nous allons créer y créer 3 fichiers : main.txt, net.txt et bdd.txt. Faites un clic droit sur le répertoire et choisissez TortoiseSVN ► Import...

Vérifiez que l'URL correspond à votre lien SVN : <https://localhost/svn/MonProjet> puis cliquez sur le bouton [...] pour choisir le répertoire d'arrivée de votre code à importer : choisissez 'Trunk' Vous obtenez alors le lien SVN suivant : <https://localhost/svn/MonProjet/trunk>, Validez



Si votre projet réel comporte des sous-dossiers, ils seront également importés. Vous pouvez également voir ce qu'un autre utilisateur verra en effectuant un clic droit et choisissant TortoiseSVN ► Repo-browser

Désormais votre projet est accessible sur votre serveur SVN. Il sera impératif de synchroniser celui-ci avec votre disque local avant de pouvoir travailler sur vos fichiers locaux. En effet, un ou plusieurs développeurs peuvent désormais modifier le code sur le SVN.

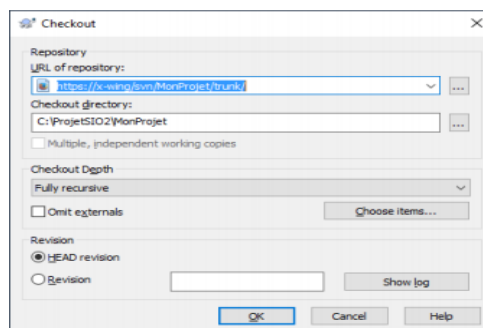


5.2 MODIFICATION D'UN PROJET

Après plusieurs jours (quelques minutes dans notre TP), vous décidez de modifier votre code.

5.2.1 Mettre à jour le répertoire local

Faites un clic droit sur le répertoire local (C:\ProjetSIO2) et choisissez SVN Checkout... Si nous laissons les paramètres par défaut, TortoiseSVN va créer un répertoire "MonProjet" dans notre répertoire de travail. Il faudrait enlever cet ajout, cependant nous allons laisser créer le répertoire, pour constater la différence avec notre code source initial. Validez en cliquant sur [OK], les fichiers sont transférés depuis le serveur vers C:\ProjetSIO2



Désormais, nous disposons de la version la plus récente du serveur. TortoiseSVN a modifié les icônes pour faciliter le repérage des modifications. Il faut donc désormais travailler sur les fichiers présents dans le répertoire MonProjet : c'est ainsi qu'il est nommé sur le serveur SVN !



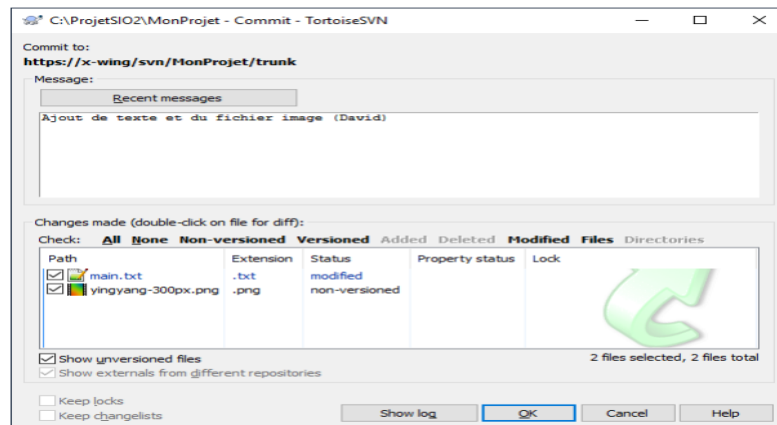
Ajoutez un autre fichier dans le répertoire MonProjet (une image, un fichier texte, etc.) Modifiez le fichier main.txt en ajoutant du texte dedans, par exemple : Voici la première modification du fichier texte

ligne 1
ligne 2
ligne 3
ligne 4

5.2.2 Mettre à jour le projet sur le serveur

Les icônes des éléments modifiés ont changé. Il faut maintenant faire un clic droit sur le répertoire MonProjet. Choisir ► SVN Commit...

Par défaut, TortoiseSVN n'ajoutera pas le nouveau fichier, veillez à le sélectionner. De même, vous pouvez ajouter un commentaire sur les modifications effectuées.



Validez en cliquant sur [OK].

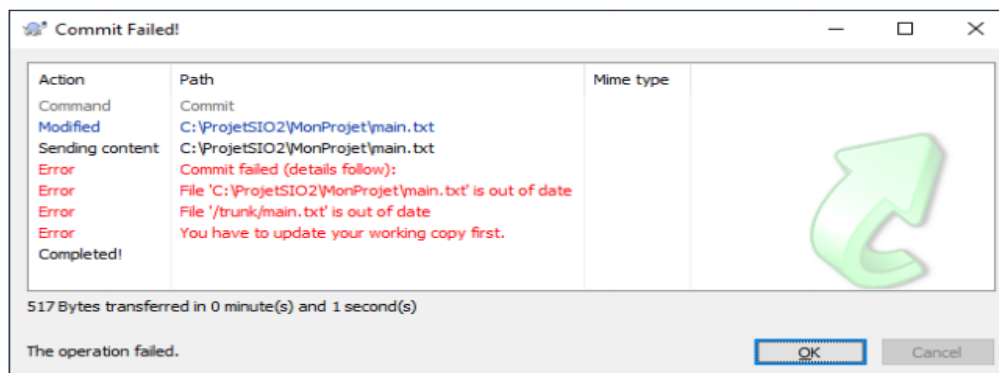
5.3 GESTION DES CONFLITS

5.3.1 Gestion automatique

Depuis un autre poste (votre binôme), essayez de récupérer le même projet et modifiez le fichier main.txt, en dessous de la ligne 4.

Plusieurs éléments peuvent empêcher la connexion depuis un autre poste : le parefeu, l'adresse IP (utilisez la commande ipconfig pour connaître l'adresse IP du serveur à communiquer à votre voisin), etc.

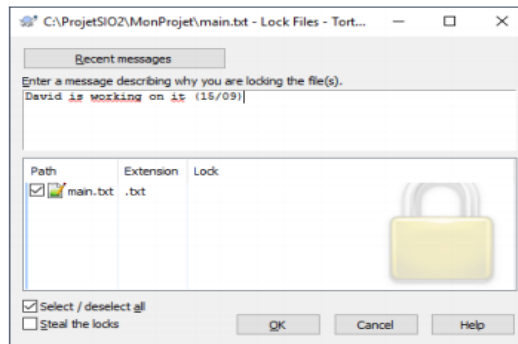
De votre côté, modifiez la ligne 2 du même fichier. Demandez à votre voisin de faire un "SVN Commit..." en premier, puis essayez à votre tour. Vous devez obtenir un message d'erreur :



Cependant, SVN dispose de la capacité à "mélanger" les fichiers. Faites un clic droit sur le fichier en défaut et sélectionnez SVN Update... Si les lignes modifiées ne sont pas les mêmes, la mise à jour devrait intégrer les modifications de votre voisin et les vôtres.

5.3.2 Gestion par verrouillage

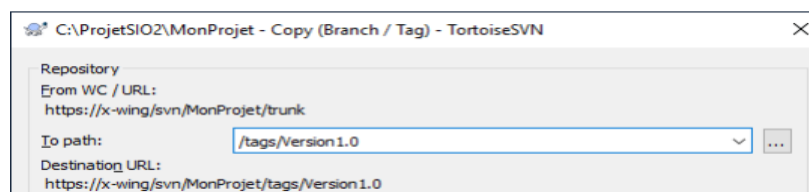
Si vous devez modifier un fichier, il est possible de le verrouiller en écriture (une sorte de réservation). Après un SVN Update... faire un clic droit sur le fichier à verrouiller et choisir TortoiseSVN ► Get Lock



La fenêtre suivante s'affiche : Soyez respectueux des autres utilisateurs en laissant un petit commentaire : date à laquelle vous y travaillez mais aussi durée, explication, etc. Attention, ce n'est qu'une tentative pour obtenir le verrouillage, pour le moment, le fichier n'est pas verrouillé. La fenêtre suivante indique si l'opération est acceptée ou non. Grâce à ce mécanisme, aucun autre utilisateur ne peut effectuer de modification sur le fichier verrouillé. N'oubliez pas de déverrouiller le fichier une fois les modifications faites : TortoiseSVN Release ► Lock...

5.4 CRÉER UNE VERSION (TAGS)

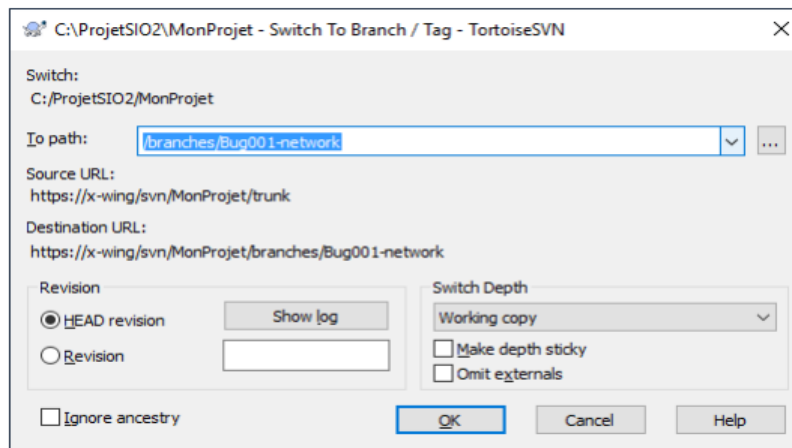
Cela permet de conserver l'ensemble des fichiers dans la version actuelle (numéro de révision). Clic droit TortoiseSVN Branch/Tags... ► Dans la fenêtre de dialogue, il est demandé le chemin pour cette nouvelle branche ou "version" (Tags). Tapez /tags/Version1.0 pour ajouter l'ensemble des fichiers dans ce répertoire.



Notez cependant, que les fichiers n'y sont pas physiquement copiés, tant qu'ils ne seront pas modifiés dans /trunk : il s'agit d'un lien symbolique (équivalent d'un raccourci pour les utilisateurs Windows). On peut le vérifier en mesurant la taille du répertoire C:\Repositories avant puis après plusieurs créations de tags : elle augmente mais de manière non-proportionnelle à la taille des fichiers.

5.5 TRAVAILLER SUR UNE BRANCHE

Commencez par créer une nouvelle branche (TortoiseSVN ► Branch/Tags...) nommé "Bug001- network" Une fois la nouvelle branche créée, vous devez basculer vos fichiers et votre synchronisation dessus. Pour cela, faites un clic droit sur le répertoire MonProjet ► TortoiseSVN ► Switch... Modifiez le chemin pour basculer sur /branches/bug001-network

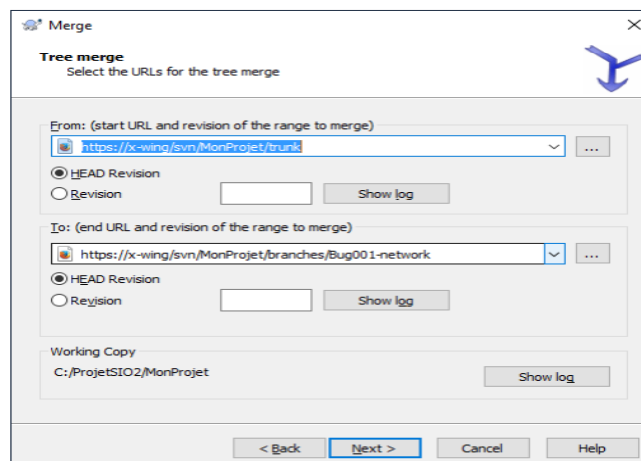


Désormais, vous travaillez sur cette branche. Modifiez le fichier net.txt en y ajoutant du texte, sauvegardez puis clic droit dessus ► SVN Commit.



Notez la modification du chemin dans la boîte de dialogue qui apparaît. 5.6 INTÉGRER LES MODIFICATIONS D'UNE BRANCHE DANS LE "TRUNK" Pour cela, il suffit de faire un clic droit sur le répertoire MonProjet ► TortoiseSVN ► Merge Dans la boîte de dialogue, choisir "Merge two different trees"

Puis d'indiquer le chemin de destination (To :) : /branches/Bug001-network



Même si cela paraît inversé, souvenez-vous que trunk est le répertoire principal : c'est de lui que débute toutes les branches (from:).

La fenêtre suivante propose un bouton [Test Merge], sinon validez. Le trunk est désormais corrigé. N'oubliez pas de basculer à nouveau sur cette version (trunk). Vous savez désormais utiliser un système de versionning. Vous pouvez continuer à expérimenter avec les fichiers courants, ou bien supprimer le lien SVN pour pouvoir supprimer le répertoire : TortoiseSVN ► Properties et choisir le bouton [Remove]