

Utiliser tortoiseSVN

Avant de commencer il faut que subversion (SVN) soit installé sur un serveur, et qu'un projet soit créé et disponible. Des tutoriels existent pour mettre en place un serveur SVN sous linux (<http://www.betaphile.net/index.php/2008/03/22/8-installer-un-serveur-subversion-avec-authentification-http-https-et-websvn-sous-linux>).

Ceci constitue un tutoriel sur la partie client sous Windows du SVN, et nous utiliserons pour cela TortoiseSVN (<http://tortoisesvn.net/downloads>).

Commencez par l'installer (il faudra rebooter). Dans notre exemple, votre copie locale sera un site web PHP, nous travaillerons donc dans le répertoire « c:/wamp/www », mais vous pourrez enregistrer votre copie locale où bon vous semble.

1) Après l'installation

TortoiseSVN s'intègre à votre environnement Windows. Lorsque vous faites un clic droit, de nouvelles options sont disponibles et nous allons apprendre à nous servir des principales.

Présentation rapide des commandes

-  Checkout : permet de créer votre copie locale du dépôt complet la première fois
-  Check for modifications : permet de vérifier que votre copie locale est bien la version la plus récente disponible, et vous indique quelles sont les différences avec votre version le cas échéant
-  Update : permet de mettre à jour votre copie locale (n'écrase pas vos modifications s'il y a des différences, vous devrez « merger » manuellement)
-  Update to revision : permet de revenir à la version de votre choix
-  Diff : permet de voir les différences entre 2 fichiers sélectionnés, ou entre le fichier sélectionné et la version du dépôt SVN. Cela vous permettra de fusionner proprement vos modifications avec celles du serveur SVN (si vous avez travaillé à plusieurs sur un même fichier simultanément, vous pourrez comparer les deux versions et fusionner à la main le contenu)
-  Commit : permet d'envoyer vos modifications au serveur SVN (n'oubliez pas de faire un update avant pour éviter les conflits)
-  Add : permet d'ajouter un fichier ou un répertoire qui n'était pas répertorié sur le dépôt, lors du prochain commit
-  Delete : permet de supprimer un fichier ou un répertoire lors du prochain commit
-  Revert : permet d'annuler un changement de statut (add, delete,...) avant le prochain commit (une fois commité il est trop tard)

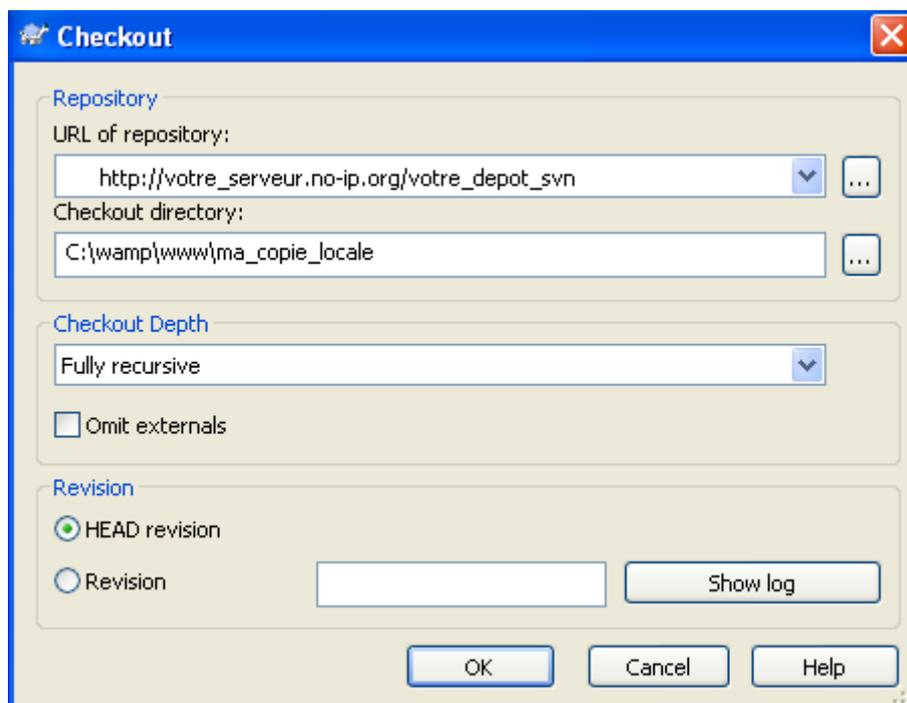
-  Rename : permet de renommer un fichier proprement, mais aussi de le déplacer (par exemple en le nommant « dossier/fichier.ext » il descendra dans « dossier »)
-  Clean up : permet de régler les conflits lorsqu'il y en a
-  Add to ignore list : permet de spécifier qu'un fichier de la copie locale ne doit pas être ajouté au dépôt.
-  Export : permet d'obtenir une copie locale propre (sans répertoires .svn) afin de la publier sur un FTP par exemple.

Toutes ces commandes sont disponibles à partir du clic droit sur un fichier ou un dossier. (Vous pouvez ajouter ou supprimer ces commandes dans « TortoiseSVN > Settings > General > Context Menu » et je vous conseille d'activer toutes les commandes citées précédemment et de désactiver les autres).

2) Checkout

La commande Checkout permet d'obtenir une « copie locale de travail » du « dépôt SVN ». Pour obtenir votre copie locale, rendez-vous dans c:/wamp/www et créez un nouveau dossier « ma_copie_locale ». Faites un clic droit sur le dossier puis « SVN Checkout ».

Et complétez les informations comme sur l'image avant de cliquer sur OK :



Après avoir annoncé quelques lignes «Added C:/wamp/www/ma_copie_locale/fichier » (selon la taille de votre dépôt cette opération peut prendre du temps), TortoiseSVN annoncera « Completed at Revision : X » (X étant le numéro de dernière révision).

Vous avez donc une copie locale sur laquelle vous pouvez travailler. Vous pouvez modifier vos fichiers à loisir sans entrainer de modifications sur le dépôt tant que vous n'effectuez pas de commit.

3) Check for modifications

Cette commande existe afin d'éviter les conflits de version, il faut vérifier de temps à autres (et TOUJOURS avant d'effectuer un commit) que votre version locale profite bien des dernières versions des fichiers contenus sur le dépôt SVN.

4) Update

La commande Update permet d'obtenir la version la plus récente des fichiers du dépôt.

Pour effectuer un update, faites un clic droit sur le fichier ou répertoire à mettre à jour, puis « SVN Update ».

Avant de travailler sur un fichier pensez à faire une update de votre copie locale de travail, vous profiterez ainsi des modifications qui ont eut lieu sur le serveur SVN. Pensez également à effectuer un Update avant vos Commit, vous pourrez ainsi déceler les fichiers sur lesquels vous devrez effectuer un Diff.

5) Update to revision

La commande Update permet d'obtenir la version de votre choix des fichiers du dépôt.

Le principe est le même que pour un update, mais vous pourrez revenir à une version précise d'un fichier ou d'un répertoire.

6) Diff

Lorsque plusieurs personnes ont accès au dépôt SVN, il arrive qu'un fichier soit modifié dans les copies locales de travail de différentes personnes en même temps. Le premier à Commiter son fichier ne verra pas la différence lorsqu'il fera un Update pour vérifier qu'il n'écrase pas les modifications d'une autre personne, mais les suivants verront leur fichier marqué d'un gros « ! » rouge 🚫, qui signifie que leur fichier est différent de celui de la version du dépôt.

Pas de panique, il suffit de faire un clic droit, puis « TortoiseSVN » puis « Diff » afin de visionner les différences entre votre copie locale et le fichier du dépôt. A vous de fusionner le fichier ou de choisir les modifications qui seront conservées.

7) Commit

La commande Commit permet de répercuter les modifications que vous avez effectué en local, sur le dépôt, afin que les autres puissent profiter de vos apports.

Attention il est fortement recommandé d'effectuer avant tout un « check for modifications » suivi d'un Update (lui-même suivi d'un Diff si besoin) avant chaque Commit, vous éviterez ainsi des conflits, ou encore d'écraser les modifications de vos partenaires !

N'oubliez pas non plus, de commenter chacun de vos commits en expliquant ce que vous avez modifié afin d'en conserver une trace, mais aussi d'expliquer à vos partenaires pourquoi faire un update sur cette version.

8) Add

La commande Add permet d'ajouter un nouveau fichier ou répertoire au dépôt. Cette action n'entraîne pas immédiatement l'ajout du fichier dans le dépôt, il ne sera envoyé que lors du prochain Commit.

En effet lorsque vous ajoutez un nouveau fichier à votre copie locale, il est tout d'abord marqué d'un « ? »  signifiant que le fichier est « unversioned » (il ne fait pas partie du dépôt). Après avoir lancée cette commande, il sera marqué d'un « + »  signifiant qu'il sera ajouté lors du prochain commit. En cas d'erreur de votre part, ne vous inquiétez pas, il suffit de faire un « Revert » avant le prochain commit.

9) Delete

La commande Delete permet de supprimer un fichier ou un répertoire du dépôt. Cette action n'est pas effective immédiatement et n'aura lieu que lors du prochain Commit, vous pourrez donc l'annuler grâce à un « Revert » en cas d'erreur. Le fichier sera marqué d'une croix rouge .

10) Revert

La commande Revert permet d'annuler une commande « Add », « Delete »,... en cas d'erreur. Cela ne sera possible qu'AVANT d'effectuer un commit qui validera « définitivement » votre modification.

11) Rename

La commande Rename permet de renommer un fichier proprement. Si vous le renommer sans passer par cette commande, Tortoise pensera que vous avez perdu le fichier « ancien_nom » et verra un fichier « nouveau_nom » unversioned.

Cette commande permet également de déplacer un fichier proprement dans l'arborescence., il suffit pour cela de spécifier le répertoire de destination dans le nom du fichier. Par exemple le fichier « ancien_nom » devra être renommé « repertoire/sous_repertoire/nouveau_nom ».

12) Clean up

Il arrive que des conflits surviennent dans une copie locale, pour les corriger il suffit de lancer la commande Clean Up, à la racine de votre copie locale.

13) Add to ignore list

Cette commande permet de spécifier qu'un fichier unversioned, ne devra pas être ajouté au dépôt. Cela est valable pour les fichiers qui ne servent qu'à vous et que vous ne souhaitez pas voir diffusé sur le dépôt, ou encore pour les fichiers temporaires créé par votre IDE (et qui risqueraient de corrompre le répertoire de travail de vos partenaires).

14) Export

La commande Export permet d'obtenir une version « propre » de votre copie locale. Une version propre est une version épurée des répertoires « .svn ». Si votre projet doit être mis sur un ftp pour être accessible à des visiteurs par exemple, les répertoires « .svn » n'ont rien à faire sur le ftp. Grâce à l'Export, vous n'aurez pas à les supprimer à la main.

15) Conseils aux débutants

Lorsqu'on effectue beaucoup de modifications d'un coup sur la copie locale de travail, il arrive qu'on supprime des fichiers ou qu'on en modifie par erreur. Lorsque cela arrive il ne faut surtout pas effectuer de Commit avant d'être certain d'avoir tout remis en ordre. Le plus simple est en général de supprimer carrément votre copie locale de travail, de récréer le dossier, et d'effectuer un nouveau Checkout.

Si vous souhaitez conserver certaines de vos modifications, n'hésitez pas à mettre de côté les fichiers intéressants, puis à utiliser la technique ci-dessus.

Vous remettre ensuite un à un les fichiers dont vous souhaitez conserver les modifications en effectuant des « diff ».

ATTENTION : N'effectuez des commits que lorsque vous êtes absolument certain que votre copie locale ne comporte pas d'erreurs, ou spécifiez le clairement dans le message de votre commit.

N'oubliez pas de commenter chacun de vos commits en expliquant ce que vous avez modifié afin d'en conserver une trace, mais aussi d'expliquer à vos partenaires pourquoi faire un update sur cette version.

pisto