

Subversion

par hugo123 (hugo.developpez.com)

Date de publication : 31/05/2006

Dernière mise à jour :

Cet article traite de l'installation et de l'utilisation de Subversion sous Windows. Il abordera son utilisation en protocole HTTP et montrera une façon simple et rapide de créer son serveur de version.

I - Introduction.....	3
Remerciements.....	3
I-1 - Prérequis.....	3
I-2 - Pourquoi cet article ?.....	3
I-3 - Qu'est-ce que Subversion ?.....	3
I-3-a - Généralités.....	3
I-3-b - Spécificités.....	3
II - Installation.....	5
II-1 - Outils nécessaires.....	5
II-2 - Mode opératoire et configuration.....	5
III - Utilisation.....	9
III-1 - Créer son espace de travail.....	9
III-2 - Icône de la copie de travail.....	13
III-3 - Mettre à jour sa copie de travail.....	13
III-4 - Propager ses modifications.....	16
Conclusion.....	19

Remerciements

Je tiens à remercier sjrd pour la correction de cet article, Ricky81, elitost et GnuX pour leurs conseils. Mes remerciements vont également droit vers les membres de Developpez.com qui oeuvrent pour un processus de publication d'une qualité remarquable.

I - Introduction

I-1 - Prérequis

Le lecteur, dans l'idéal, a déjà installé les outils nécessaires à ce tutoriel. Cependant, cet article reste à la portée de tout le monde si on ne s'attache pas à comprendre certaines parties (comme le paramétrage Apache).

I-2 - Pourquoi cet article ?

Subversion est de plus en plus utilisé mais les documentations qui s'y rapportent sont hétéroclites ou ne contiennent pas toujours toutes les informations nécessaires à son bon fonctionnement. Ici nous aborderons son utilisation dans un environnement Windows. Le lecteur pourra ainsi installer et utiliser une interface graphique et un serveur web en quelques clics.

I-3 - Qu'est-ce que Subversion ?

I-3-a - Généralités

Subversion est un outil de gestion de documents. Il se veut comme successeur logique de CVS et nous provient du monde de l'open source. Vous pourrez consulter à cette adresse le site du [projet Subversion](#).

I-3-b - Spécificités

Ce outils reprend toutes les fonctionnalités de CVS, à savoir :

- La consultation et la possibilité de restauration des anciennes versions d'un fichier.
- Les raisons des modifications apportées et les auteurs de celles-ci.
- Les modifications de versions sont stockées sous forme de delta (différence entre deux versions) pour les fichiers textes. Ce qui implique qu'une propagation est proportionnelle aux modifications apportées et non à la tailles des données initiales.

Les avantages rajoutés par Subversion sont entre autres (cf. page d'accueil du site ci-dessus) ceux-ci :

- Subversion permet de tracer les versions de répertoires, de fichiers et de droits sur les fichiers.
- Subversion permet de renommer un fichier ou un répertoire tout en conservant son historique.
- Les propagations de version (commit) sont atomiques. Une propagation réussit uniquement si tout les fichiers de la version sont correctement propagés.
- Les numéros de versions concernent une propagation et non les fichiers eux-mêmes.
- Subversion peut se coupler avec le protocole HTTP Webdav/deltaV.
- Les sauvegardes par delta sont aussi possibles sur les fichiers binaires.

Voici quelques points de vocabulaire pour la suite de cet article :

- repository/répertoire de dépôt : Ce répertoire contiendra tout votre référentiel de versionnage. C'est une base de données et un ensemble de fichiers de configuration.


- working area/copie de travail : Une copie de travail est une image complète ou partielle du répertoire de dépôt. Une copie de travail peut se synchroniser avec le répertoire de dépôt suivant des mécanismes que nous verrons plus bas.

II - Installation

II-1 - Outils nécessaires

Dans cet article, nous traiterons d'une installation sur un système Windows. Nous allons donc utiliser un ensemble de logiciels, tous issus du monde libre, afin de nous permettre de créer un serveur de configuration très rapidement. Téléchargez les logiciels suivants :

- **Apache 2** : serveur web. Ce logiciel ne sera pas nécessaires sur les postes clients.
- **Subversion** : gestionnaire de version. Ce logiciel ne sera pas nécessaires sur les postes clients.
- **TortoiseSvn** : interface graphique pour subversion.

 *Si vous travaillez sur des sites Internet ou Intranet en PHP, n'hésitez pas à utiliser **Wamp** contenant apache2, mysql5, php5.*

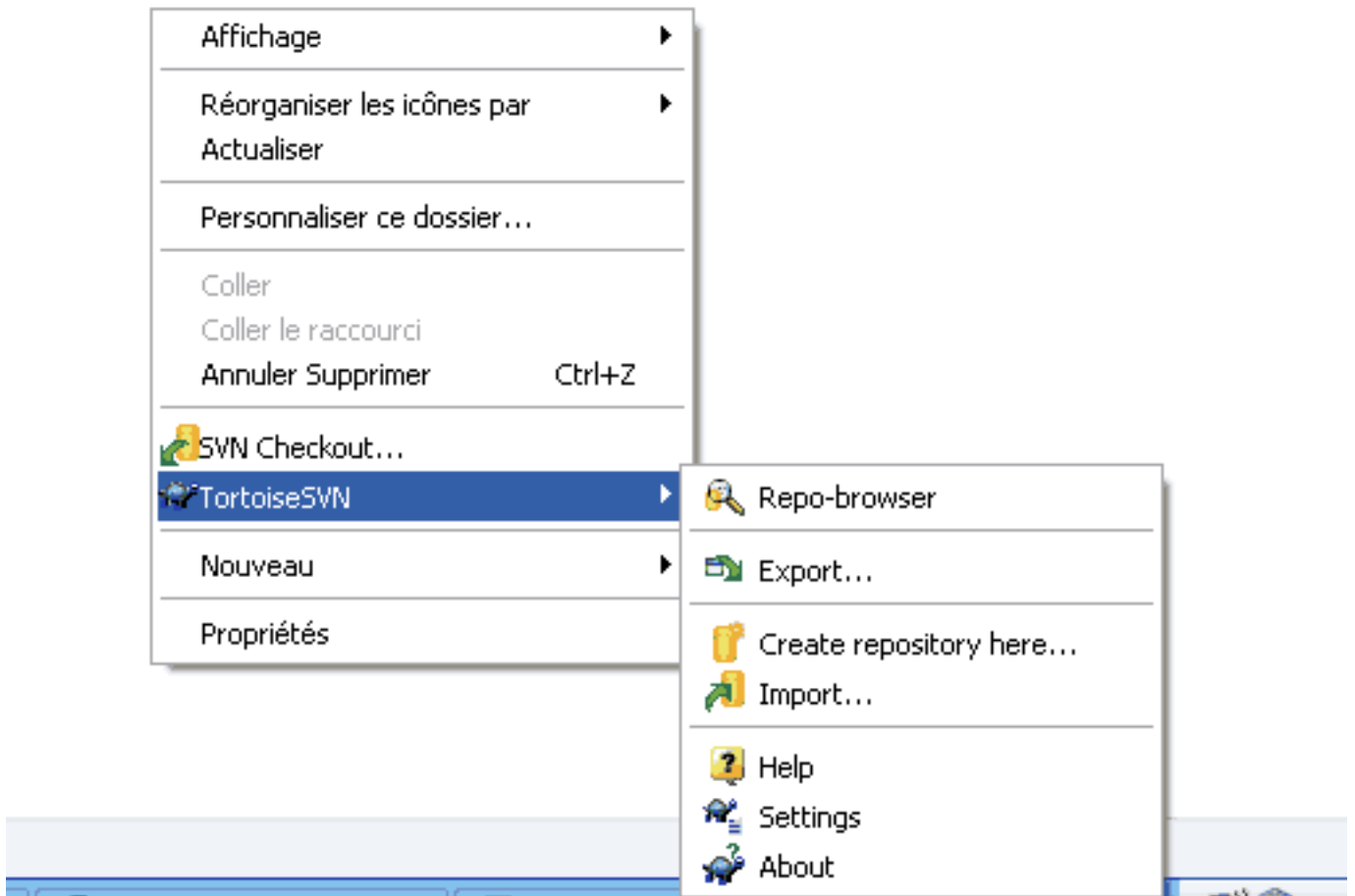
II-2 - Mode opératoire et configuration

Pour tous les logiciels cités plus haut, contentez-vous de suivre les écrans d'installations de leurs installateurs réciproques.


Une fois vos logiciels installés, nous allons maintenant les configurer pour les faire fonctionner ensemble.

La première étape consiste à créer son répertoire de dépôt. Nous allons pour cela utiliser l'interface graphique de TortoiseSvn.

Si votre installation s'est bien passée, vous devriez avoir des options supplémentaires dans le menu contextuel de l'explorateur Windows.



menu contextuel

 **Attention**, si vous utilisez une plateforme 64 bits, le menu contextuel ne sera pas modifié dans l'explorateur Windows. Cependant, vous pourrez utiliser l'explorateur de Internet Explorer qui est resté compatible 32 bits.

Choisissez donc un répertoire, par exemple d:\svn et utilisez l'option : "Create repository here". Ici vous aurez la possibilité de choisir deux types de base de données pour stocker vos versions :

- Berkeley DB
- FSFS (native file system)

Traditionnellement on conseille d'utiliser FSFS qui est développé par Subversion. Cependant les deux possèdent des avantages et inconvénients. Vous pourrez trouver des informations sur les deux bases de données [ici](#).

Vous avez donc maintenant un répertoire de dépôt pleinement opérationnel. Nous allons le rendre accessible par HTTP. Vous pouvez déjà l'explorer à partir du repo-browser en mode file.

Dans le répertoire bin/ d'installation de subversion, vous devez trouver les fichiers suivants : mod_authz_svn.so et mod_dav_svn.so. Copiez les dans le répertoire Apache2\modules de Apache.


Ouvrez maintenant le fichier Apache2\conf\httpd.conf. Vous allez activer les modules qui permettront à Subversion d'utiliser le protocole http et l'authentification.

```
# Décommentez la ligne : LoadModule dav_module modules/mod_dav.so
LoadModule dav_module modules/mod_dav.so
# et ajoutez les lignes suivantes en dessous :
LoadModule dav_svn_module modules/mod_dav_svn.so
LoadModule authz_svn_module modules/mod_authz_svn.so
```

Les modules sont activés mais nous allons maintenant rajouter un emplacement pour accéder à votre dépôt :

```
# nom du répertoire. L'accès se fera par http://localhost/svnrepository
<Location /svnrepository>
  DAV svn
  # répertoire de dépôt
  SVNPath d:/svn
  SVNIndexXSLT "/svnindex.xsl"
  SVNAutoversioning on

  # Indique comment authentifier les utilisateurs
  AuthType Basic
  AuthName "Référentiel Subversion"
  # fichiers des utilisateurs
  AuthUserFile d:\svn\conf\utilisateurs.txt
  Require valid-user
  # fichiers des permissions
  AuthzSVNAccessFile e:\GDASvn\conf\authz
  ErrorDocument 404 default
</Location>
```

 Afin d'explorer le répertoire de dépôt à partir d'un navigateur web, nous avons ajouté la directive **SVNIndexXSLT "/svnindex.xsl"** qui permet de définir une feuille de style pour l'affichage. La feuille de style de base se trouve **sur le site de subversion**. Les fichiers **svnindex.xsl** et **svnindex.css** doivent être déposés dans le répertoire racine des documents Apache. (Le répertoire **www** si vous préférez).

Désormais il ne reste plus qu'à gérer les permissions et les utilisateurs. Les fichiers qui les définissent sont dans votre répertoire de dépôt sous le répertoire conf. Nous allons d'abord nous intéresser au fichier des permissions : **authz**.

```
# Ce fichier a le gros avantage d'être documenté et surtout, bien documenté.
# Voici un exemple de fichier qui définit deux groupes, des développeurs, des traducteurs.
# Le répertoire svn contient
# - un répertoire html qui est accessible aux traducteurs
# - des répertoire src/ accessible aux développeurs
# - un répertoire changelog accessible à tous

### This file is an example authorization file for svnserve.
### Its format is identical to that of mod_authz_svn authorization
### files.
### As shown below each section defines authorizations for the path and
### (optional) repository specified by the section name.
### The authorizations follow. An authorization line can refer to a
### single user, to a group of users defined in a special [groups]
### section, or to anyone using the '*' wildcard. Each definition can
### grant read ('r') access, read-write ('rw') access, or no access
### ('').

[groups]
dev = user1,user2,user3
trad = user4,user5

[svnrepository:/]
@dev = rw
* =

[svnrepository:/html]
@trad = rw
@dev = rw
* =
```

```
[svnrepository:/src]
@trad =
@dev = rw
* =

[svnrepository:/changelog]
@dev = rw
* = r
```

Vos permissions sont définies mais votre application ne connaît pour l'instant aucun utilisateur. Ceux-ci doivent être définis dans le fichier utilisateurs.txt. Ce fichier est géré par htpasswd, utilitaire fourni avec Apache dans son répertoire bin.



N'hésitez pas à rajouter le répertoire Apache2/bin dans le PATH (variable d'environnement Windows), cela vous évitera de taper le chemin complet de l'utilitaire à chaque fois

Pour créer votre fichier utilisateurs.txt avec votre premier utilisateur :

```
htpasswd -bc utilisateurs.txt user1 passwordUser1
```

Pour rajouter un utilisateur :

```
htpasswd -bm utilisateurs.txt user2 passwordUser2
```

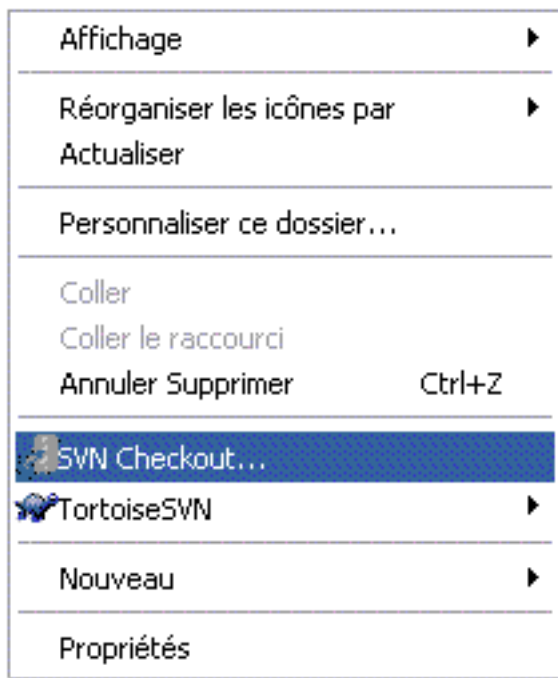
Voilà, votre serveur est désormais configuré et vous pouvez y accéder par votre repo-browser à partir de l'adresse : <http://votreadresse/svnrepository>

III - Utilisation

Les personnes désirant utiliser le serveur que vous venez de créer devront tout d'abord installer TortoiseSvn. Subversion et Apache2 ne sont pas nécessaires sur le poste client.

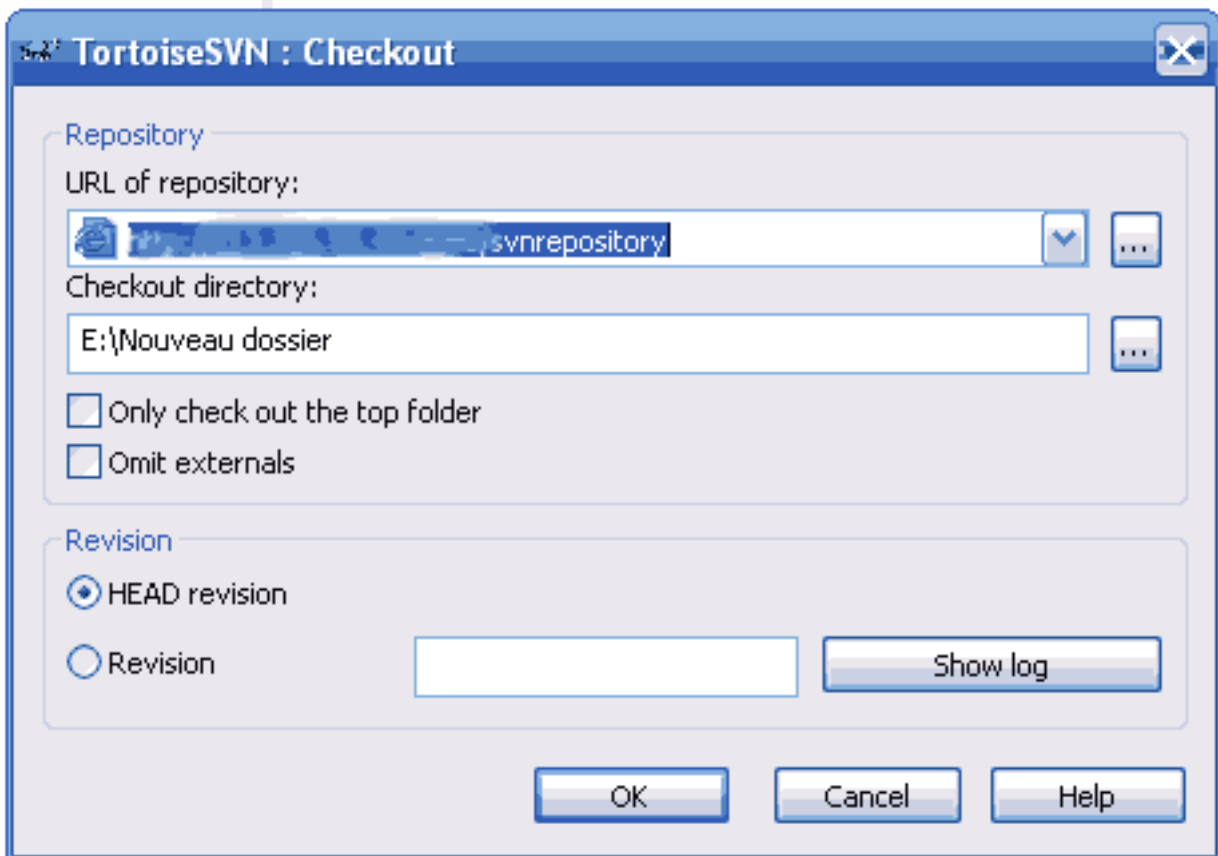
III-1 - Créer son espace de travail

Afin de créer sa copie de travail, les utilisateurs doivent créer un répertoire vide. Puis, dans le menu contextuel svn, ils doivent sélectionner **checkout**.



checkout

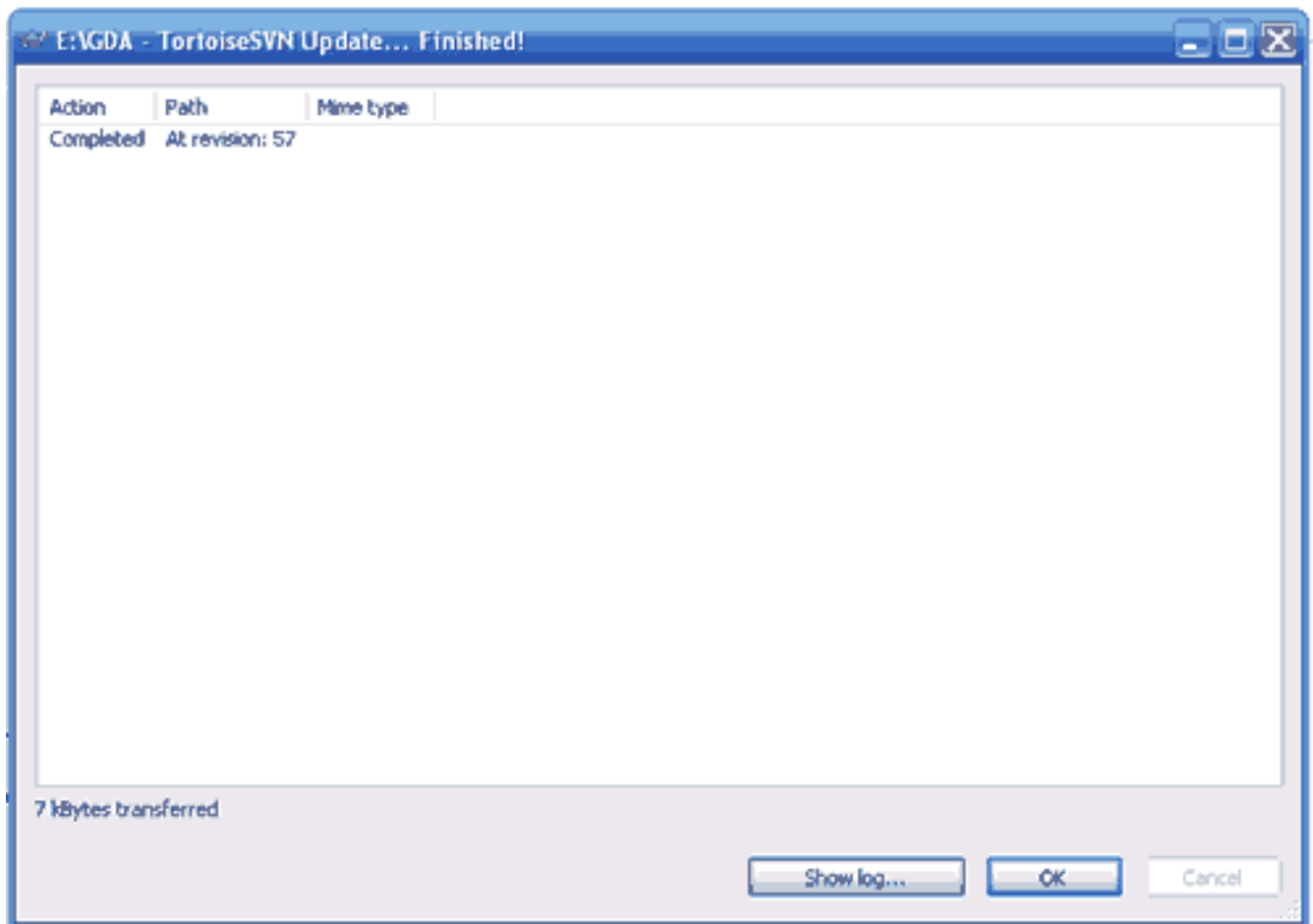
Une fenêtre s'ouvre alors. Vous devez donner l'URL du répertoire de dépôt et le nom du répertoire de destination (qui doit être vide). Les autres cases peuvent être laissées par défaut. Vous remarquerez notamment qu'il vous est possible de faire un checkout d'une version précise. Par défaut vous prendrez la version HEAD, qui est la dernière.



checkout

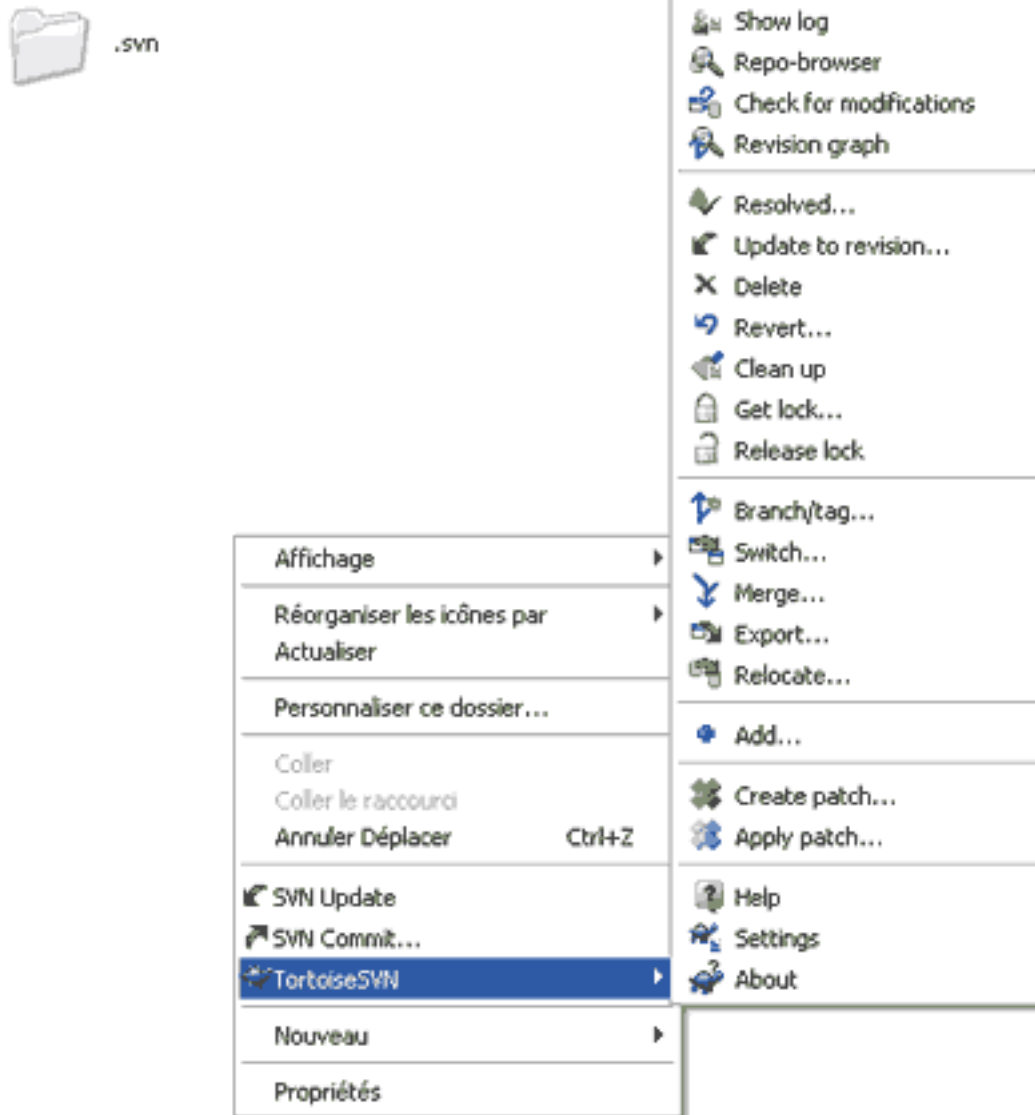
Une fois cette opération effectuée, une fenêtre s'ouvre et vous liste tous les fichiers créés sur votre copie de travail. Vous obtenez aussi le numéro de la révision actuelle.

(Note : l'exemple est une fenêtre d'update vide car aucune mise à jour n'a été récupéré. La fenêtre est cependant identique.)



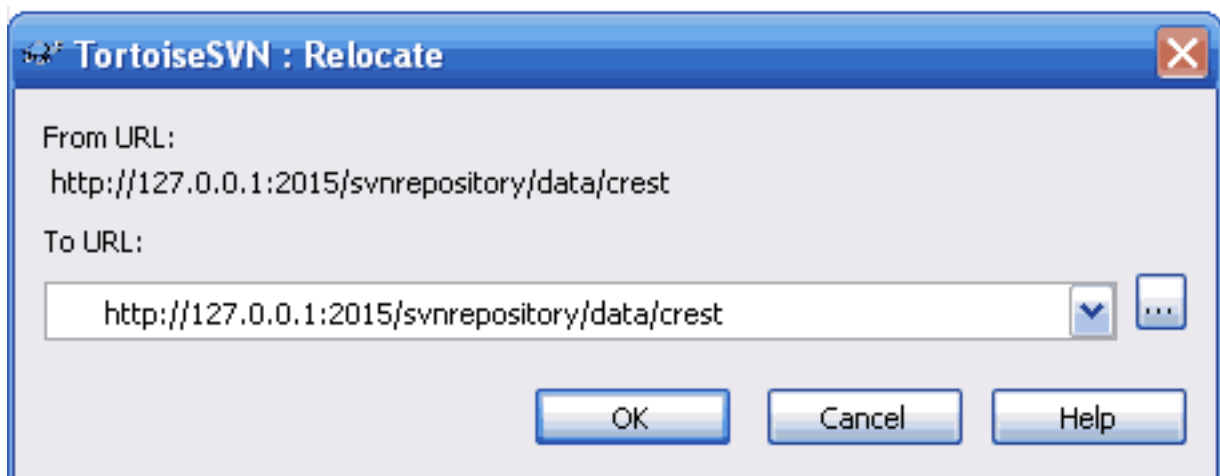
checkout

Une opération essentielle à connaître pour maintenir votre copie de travail est la relocalisation. En effet, si le serveur change d'adresse, vous aurez besoin de relocaliser votre copie de travail. L'option relocate se trouve dans le menu contextuel :



relocalisation

Une fenêtre s'ouvre alors afin de vous demander la nouvelle adresse. Comme vous le voyez sur l'exemple, vous pouvez ne relocaliser qu'un répertoire en particulier.



relocalisation

III-2 - Icône de la copie de travail

TortoiseSvn vous a rajouté des options dans le menu contextuel de l'explorateur mais il a aussi rajouté la gestion des icônes reflétant l'état des fichiers.



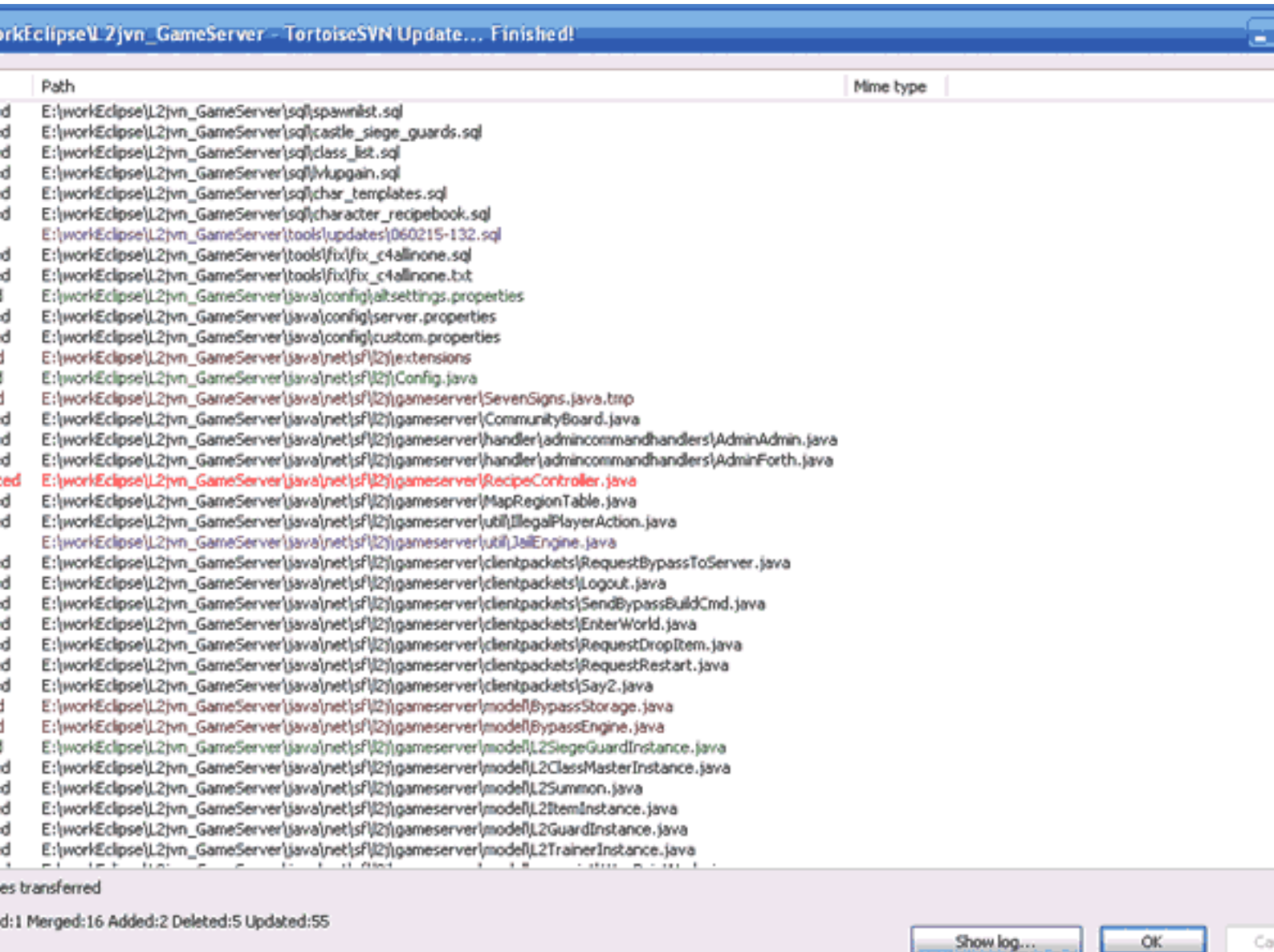
icônes

- Ajout : ces fichiers vont être ajoutés au référentiel Subversion
- Delete : ces fichiers vont être supprimés du référentiel
- Conflit : ces fichiers de votre copie de travail sont en conflit avec la version du répertoire de dépôt
- Dans Subversion : ces fichiers sont sous contrôle de version
- Locké : ces fichiers sont verrouillés
- Modifié : ces fichiers ont été modifiés sur votre copie de travail
- Lecture seule : ces fichiers ne peuvent être modifiés

Ces icônes sont cependant modifiables, celles-ci correspondent au style par défaut (XP classic).

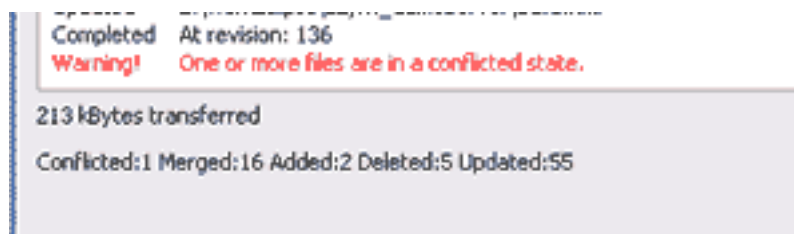
III-3 - Mettre à jour sa copie de travail

Nous allons maintenant mettre à jour notre version de travail. Cette mise à jour se fait avec l'option "update" du menu contextuel. La fenêtre qui va s'ouvrir alors va nous fournir un ensemble d'informations. Nous allons ainsi connaître le détail de la mise à jour fichier par fichier, le type de mise à jour (ajout, suppression, modification) et les éventuels conflits avec notre version locale. Subversion est capable de gérer pour vous la fusion de documents lorsque les différences entre la version locale et la version distante ne portent pas sur les mêmes lignes.



mise à jour

Les dernières informations de cette fenêtre vous donnent un bilan de la mise à jour, l'état de la mise à jour et le numéro de version du référentiel récupéré. Dans notre exemple, notre mise à jour a entraîné un conflit sur au moins un fichier.



conflit

L'une des options intéressantes de cette fenêtre est le bouton "show log". Cette fonctionnalité est aussi accessible par le menu contextuel. Elle permet de retrouver toutes les modifications d'un fichier ou d'une arborescence, les dates de modifications et les auteurs. Une version étant un propagation d'un ensemble de fichier, nous sommes en mesure de voir tous les fichiers qui ont été modifiés en même temps que celui que nous observons et un commentaire de la personne qui a propagé cette modification.

TortoiseSVN : Log Messages - E:\workEclipseV2\free_GameServer_c4\java

From: 10/03/2006 To: 03/05/2006 Messages, authors and paths

Revision	Actions	Author	Date	Message
284		DaDummy	14:32:26, mercredi 3 mai 2006	removed useless debug print
282		evil33t	19:04:21, lundi 1 mai 2006	few minor fixes for subclass fixes .)
281		evil33t	16:32:44, lundi 1 mai 2006	updated to l2jc4 rev 1879
280		evil33t	16:04:51, lundi 1 mai 2006	updated to l2jc4 rev 1878
278		evil33t	15:22:04, lundi 1 mai 2006	now all players can learn fishing skills ^.^
276		evil33t	15:03:42, lundi 1 mai 2006	This will fix/add: 1 Incorrect/Merged Skills 2 Buff Modifiers Carring Between Classes 3 New :
271		evil33t	17:17:28, dimanche 30 avril 2006	updated to l2jc4 rev 1872
270		evil33t	03:56:47, samedi 29 avril 2006	reverted a few fisherman skill functions to reenale skill learning
265		evil33t	00:06:47, samedi 29 avril 2006	fix for fishing
263		evil33t	23:54:46, vendredi 28 avril 2006	updated to l2jc4 rev 1861
260		evil33t	23:09:44, vendredi 28 avril 2006	updated to l2jc4 rev 1860
258		evil33t	22:56:20, vendredi 28 avril 2006	l2jc4 rev 1858
256		evil33t	22:48:38, vendredi 28 avril 2006	MotherTree Zone
255		evil33t	22:39:55, vendredi 28 avril 2006	updated to l2jc4 rev 1857
253		evil33t	22:35:47, vendredi 28 avril 2006	seven signs update Festival's fee Festival's monsters drop enabled
251		evil33t	22:30:01, vendredi 28 avril 2006	l2jc4 rev 1856

removed useless debug print

Action	Path	Copy from path	Revision
Modified	./L2_GameServer_c4/java/net/sf/l2j/gameserver/model/l2Multicell.java		

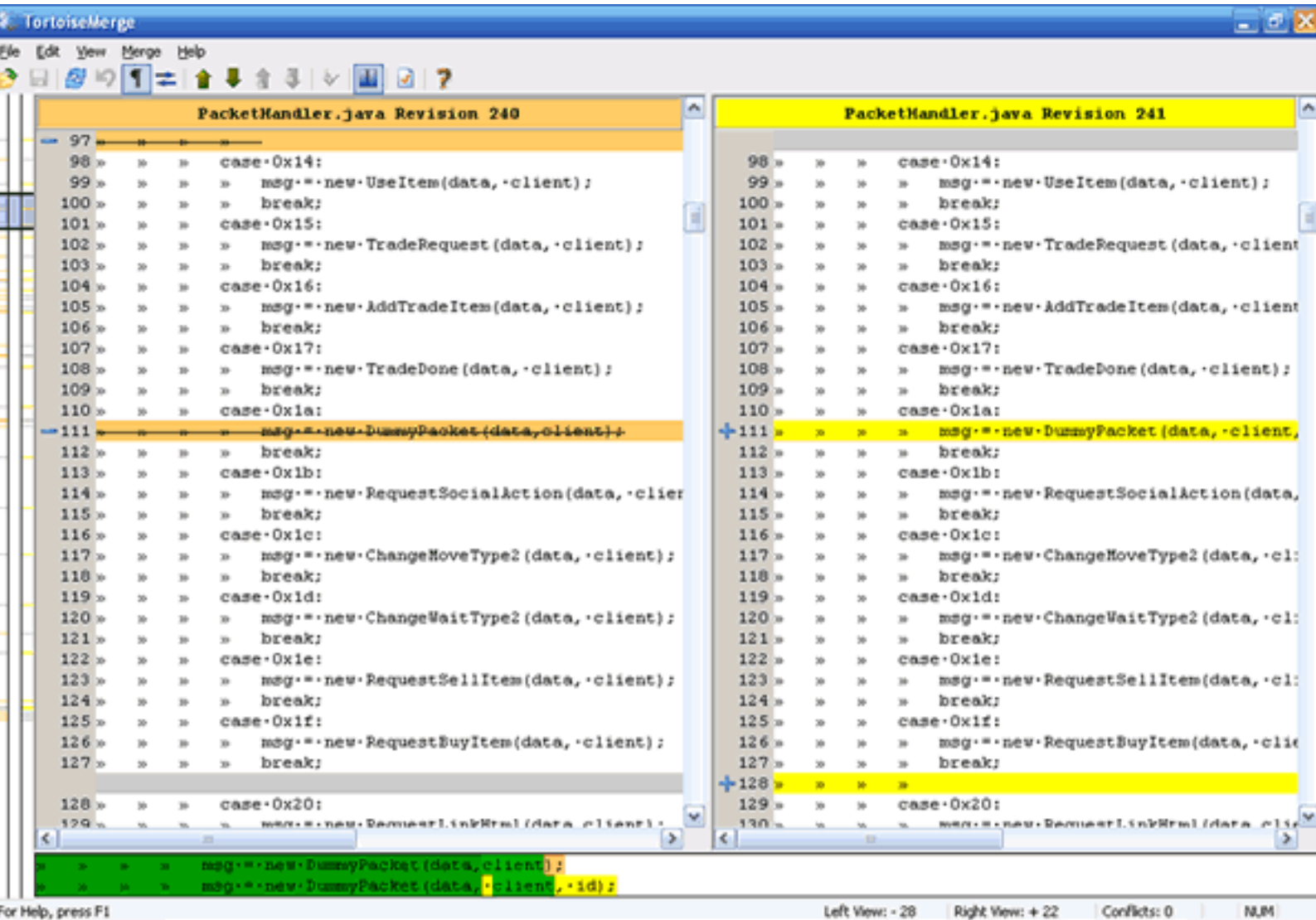
☒ Hide unrelated changed paths

Show All Next 100 ☐ Stop on copy/rename

Statistics Help OK

versions

Sur les deux écrans que nous venons d'étudier, nous avons la possibilité de cliquer sur le nom des fichiers et de visualiser les différences entre la version précédente et la version pointée. TortoiseSvn va alors ouvrir un visualiseur fourni avec l'interface. Ce visualiseur nous fournit des informations sur les différences de façon intuitive à partir d'un code couleurs.

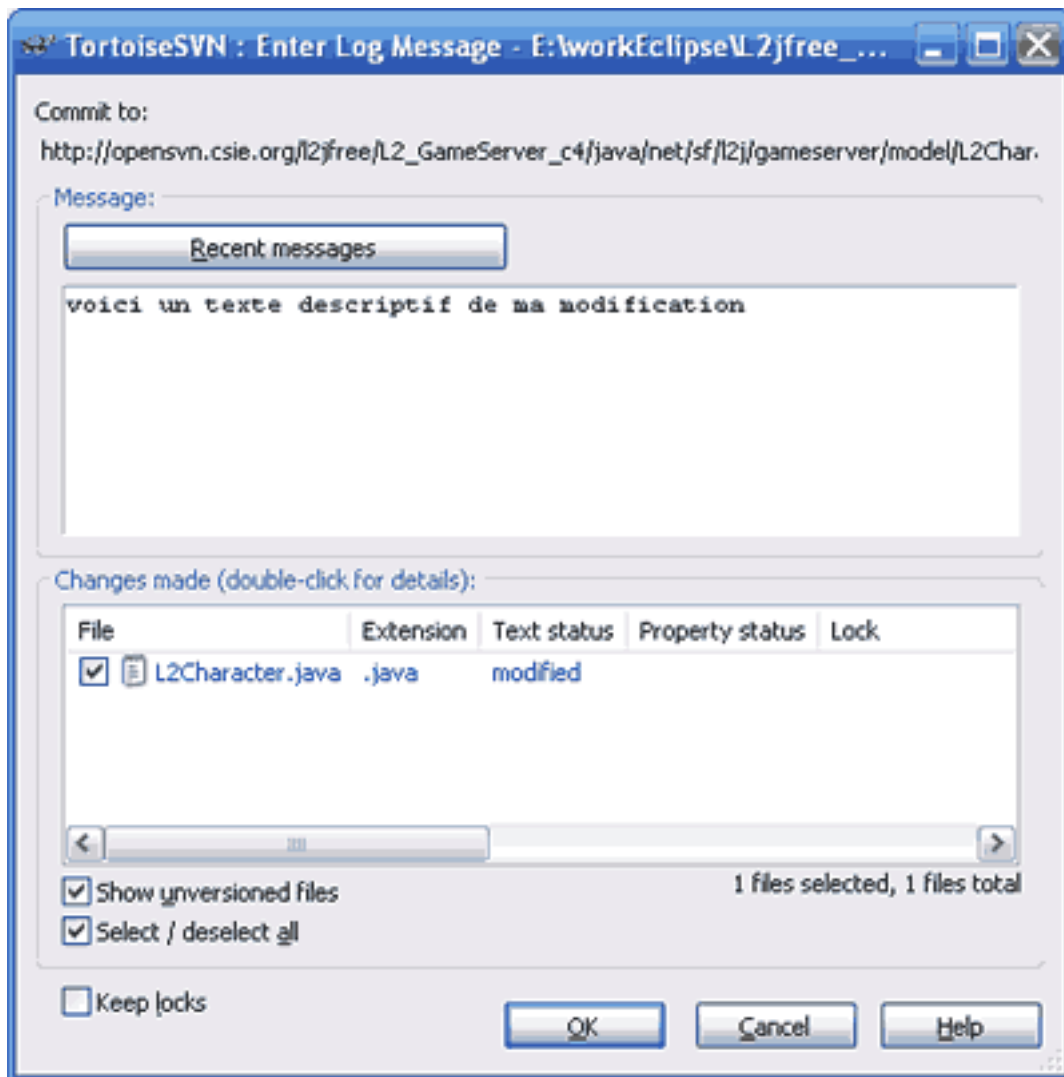


diff

III-4 - Propager ses modifications

Nous avons vu comment recevoir le fruit du travail des autres, voilà maintenant comment propager le nôtre. En pratique il existe deux méthodes, la propagation directe et la propagation par soumission de patch. Sur certains projets, typiquement les projets open source, le référentiel est ouvert en lecture à tout le monde mais seuls les développeurs ont accès en écriture. Les développeurs sont donc les seuls à pouvoir propager directement. Mais les personnes voulant contribuer de façon occasionnelle peuvent modifier leur copie de travail et créer leur patch pour l'envoyer à l'équipe de développement. Celle-ci se chargera d'appliquer le patch si nécessaire.

Voyons tout d'abord la méthode directe. Faites un clic droit sur un fichier que vous avez modifié. Ce fichier doit avoir une icône rouge comme vu plus haut. Sélectionnez : "commit". Une fenêtre apparaît. Ici vous devez saisir un texte décrivant la version que vous envoyez sur le référentiel. L'écran du bas liste les fichiers concernés par la propagation. Il vous est encore possible de ne pas propager certains fichiers en les décochant.



mise à jour

Pour la propagation par patch, l'opération est sensiblement la même. Dans le menu contextuel vous choisissez "create patch". Une fenêtre s'ouvre dans laquelle vous choisissez les fichiers. Vous avez alors la possibilité de créer un fichier .patch ou .diff. Ce fichier est exploitable par les développeurs qui pourront l'appliquer à leur copie de travail. Ce fichier diff ou patch adopte un format qui se rapproche d'un diff unique. En voici un exemple :

```
Index: E:/workEclipse/L2jfree_GameServer_c4/java/net/sf/l2j/gameserver/model/L2Character.java
=====
--- E:/workEclipse/L2jfree_GameServer_c4/java/net/sf/l2j/gameserver/model/L2Character.java (révision
285)
+++ E:/workEclipse/L2jfree_GameServer_c4/java/net/sf/l2j/gameserver/model/L2Character.java (copie de
travail)
@@ -48,6 +48,7 @@
import net.sf.l2j.gameserver.model.L2Skill.SkillType;
import net.sf.l2j.gameserver.model.actor.appearance.CharAppearance;
import net.sf.l2j.gameserver.model.actor.instance.L2ArtefactInstance;
+import net.sf.l2j.gameserver.model.actor.instance.L2SummonInstance;
import net.sf.l2j.gameserver.model.actor.instance.L2BoatInstance;
import net.sf.l2j.gameserver.model.actor.instance.L2DoorInstance;
import net.sf.l2j.gameserver.model.actor.instance.L2MonsterInstance;
@@ -3618,6 +3619,16 @@
    }
}

+
+    else if(this instanceof L2SummonInstance)
+    {
```

```
+      ((L2SummonInstance)this).displayHitMessage(damage,crit,miss);  
+    }  
+    else if(this instanceof L2PetInstance)  
+    {  
+      ((L2PetInstance)this).displayHitMessage(damage,crit,miss);  
+    }  
+  
+  
+    // If L2Character target is a L2PcInstance, send a system message  
+    if (target instanceof L2PcInstance)  
+    {
```

Conclusion

Voici quelques liens qui pourront vous être utiles pour trouver les outils cités dans cet article.

- **Apache** : serveur web. Si vous travaillez sur des sites Web, n'hésitez pas à utiliser **Wamp** contenant apache2, mysql5, php5.
- **Subversion** : gestionnaire de version. Ce logiciel ne sera pas nécessaires sur les postes clients.
- **TortoiseSvn** : interface graphique pour Subversion.

Et voici quelques articles **Développez.com** qui traite de gestion de version.

- **Un autre tutoriel Subversion**
- **introduction à CVS**
- **Utiliser CVS**

Cet article ne se veut pas exhaustif sur la question. Il a un but didactique afin de répondre a des besoins simples et traite essentiellement de points que j'utilise moi-même régulièrement. Toutefois, si vous désirez l'enrichir de vos remarques, vous pouvez me contacter par MP forum.

De multiples captures d'écran de cet article ont été faites à partir d'un dépôt ouvert à tous d'un projet open source. Ils ont permis de fournir un éventail d'écrans représentatif.