

# **TortoiseSVN**

**Un client Subversion pour Windows**

**Version 1.6.16**

**Stefan Küng  
Lübbe Onken  
Simon Large**

---

# **TortoiseSVN: Un client Subversion pour Windows: Version 1.6.16**

par Stefan Küng, Lübbe Onken, et Simon Large  
traduction: Jérémy Badier (jeremy.badier@gmail.com)

Publié le 2011/01/21 21:21:17 (r20750)

---

---

# Table des matières

Préface .....	xi
1. Public .....	xi
2. Guide de lecture .....	xi
3. TortoiseSVN est gratuit ! .....	xii
4. Communauté .....	xii
5. Remerciements .....	xii
6. Terminologie utilisée dans ce document .....	xii
1. Introduction .....	1
1.1. Qu'est-ce que TortoiseSVN ? .....	1
1.2. L'historique de TortoiseSVN .....	1
1.3. Les fonctionnalités de TortoiseSVN .....	1
1.4. Installer TortoiseSVN .....	2
1.4.1. Configuration requise .....	2
1.4.2. Installation .....	3
1.4.3. Packs de langue .....	3
1.4.4. Vérificateur d'orthographe .....	3
2. Concepts de base du contrôle de version .....	5
2.1. Le référentiel .....	5
2.2. Modèles de gestion de version .....	5
2.2.1. Le problème du partage de fichier .....	6
2.2.2. La solution Verrouiller-Modifier-Déverrouiller .....	6
2.2.3. La solution Copier-Modifier-Fusionner .....	7
2.2.4. Que fait Subversion ? .....	9
2.3. Subversion en action .....	9
2.3.1. Copies de travail .....	9
2.3.2. URL de référentiel .....	11
2.3.3. Révisions .....	12
2.3.4. Comment les copies de travail suivent le référentiel .....	13
2.4. Résumé .....	14
3. Le référentiel .....	15
3.1. Création de référentiel .....	15
3.1.1. Créer un référentiel avec le client de ligne de commande .....	15
3.1.2. Créer le référentiel avec TortoiseSVN .....	15
3.1.3. Accès local au référentiel .....	16
3.1.4. Accéder à un référentiel situé dans un partage réseau .....	16
3.1.5. Disposition du référentiel .....	17
3.2. Sauvegarde de référentiel .....	18
3.3. Scripts de hook côté serveur .....	19
3.4. Liens d'extraction .....	19
3.5. Accéder au référentiel .....	20
3.6. Serveur basé sur Svnserve .....	20
3.6.1. Introduction .....	20
3.6.2. Installer svnserve .....	21
3.6.3. Exécuter svnserve .....	21
3.6.4. Authentification de base avec svnserve .....	23
3.6.5. Sécuriser le serveur avec SASL .....	24
3.6.6. Authentification avec svn+ssh .....	25
3.6.7. Autorisation basée sur le chemin avec svnserve .....	25
3.7. Serveur basé sur Apache .....	26
3.7.1. Introduction .....	26
3.7.2. Installer Apache .....	26
3.7.3. Installer Subversion .....	27
3.7.4. Configuration .....	28
3.7.5. Plusieurs référentiels .....	30
3.7.6. Autorisation basée sur le chemin .....	30

---

3.7.7. Authentification avec un domaine Windows .....	31
3.7.8. Plusieurs sources d'authentification .....	32
3.7.9. Sécuriser le serveur avec SSL .....	33
3.7.10. Utiliser des certificats avec des hôtes SSL virtuels .....	35
4. Guide d'utilisation quotidienne .....	37
4.1. Pour commencer .....	37
4.1.1. Recouvrement d'icônes .....	37
4.1.2. Menus contextuels .....	37
4.1.3. Glisser-déposer .....	40
4.1.4. Raccourcis communs .....	40
4.1.5. Authentification .....	41
4.1.6. Maximiser les fenêtres .....	42
4.2. Importer des données dans un référentiel .....	42
4.2.1. Importer .....	42
4.2.2. Importer en place .....	43
4.2.3. Fichiers spéciaux .....	44
4.3. Extraire une copie de travail .....	44
4.3.1. Profondeur d'extraction .....	45
4.4. Livrer vos changements au référentiel .....	47
4.4.1. La boîte de dialogue Livrer .....	47
4.4.2. Listes de changements .....	49
4.4.3. Exclure des éléments de la livraison .....	49
4.4.4. Commentaires de livraison .....	49
4.4.5. Progression de la Livraison .....	51
4.5. Mettre à jour votre copie de travail avec les changements des autres .....	52
4.6. Résoudre des conflits .....	54
4.6.1. Conflit de fichiers .....	54
4.6.2. Conflits dans l'arborescence .....	55
4.7. Obtenir des information sur le statut .....	58
4.7.1. Recouvrement d'icônes .....	58
4.7.2. Les colonnes de TortoiseSVN dans l'explorateur Windows .....	59
4.7.3. Statut local et distant .....	60
4.7.4. Voir les différences .....	62
4.8. Listes de changements .....	62
4.9. La boîte de dialogue du Journal de révision .....	64
4.9.1. Appeler la boîte de dialogue du Journal de révision .....	65
4.9.2. La boîte de dialogue du Journal de révision .....	65
4.9.3. Obtenir des informations supplémentaires .....	66
4.9.4. Obtenir plus de commentaires .....	71
4.9.5. Révision de la Copie de Travail Courante .....	71
4.9.6. Fonctionnalités de Suivi des Fusions .....	72
4.9.7. Changer le commentaire et l'auteur .....	73
4.9.8. Filtrer les commentaires .....	73
4.9.9. Informations statistiques .....	74
4.9.10. Mode hors ligne .....	76
4.9.11. Refraîchissement de l'affichage .....	77
4.10. Voir les différences .....	77
4.10.1. Différences de fichier .....	77
4.10.2. Options de fins de ligne et d'espacement .....	78
4.10.3. Comparer des répertoires .....	79
4.10.4. Comparaison des images en utilisant TortoiseIDiff .....	80
4.10.5. Outils de différenciation/fusion externes .....	81
4.11. Ajouter de nouveaux fichiers et répertoires .....	82
4.12. Copier/Déplacer/Renommer des Fichiers et des Dossiers .....	83
4.13. Ignorer des fichiers et des répertoires .....	84
4.13.1. L'utilisation des pattern matching dans la liste des fichier à ignorer .....	85
4.14. Supprimer, déplacer et renommer .....	86
4.14.1. Supprimer des fichiers et des dossiers .....	86

---

4.14.2. Déplacer des fichiers et des dossiers .....	87
4.14.3. Modifier la casse dans le nom d'un fichier. ....	88
4.14.4. Gestion des conflits de nom de fichier. ....	88
4.14.5. Réparer les renommages de fichier .....	89
4.14.6. Supprimer les fichiers non versionnés .....	89
4.15. Annuler les changements .....	89
4.16. Nettoyer .....	91
4.17. Configuration des projets .....	91
4.17.1. Propriétés Subversion .....	92
4.17.2. Propriétés du projet TortoiseSVN .....	96
4.18. Eléments externes .....	98
4.18.1. Répertoires externes .....	98
4.18.2. Fichiers externes .....	100
4.19. Brancher / Étiqueter .....	101
4.19.1. Créer une branche ou une étiquette .....	101
4.19.2. Extraire ou aller sur... .....	103
4.20. Fusionner .....	104
4.20.1. Fusionner une plage de révisions .....	105
4.20.2. Réintégrer une branche .....	106
4.20.3. Fusionner deux arbres différents .....	107
4.20.4. Options de fusion .....	108
4.20.5. Prévisualiser les résultats de la fusion .....	109
4.20.6. Suivi des fusions .....	110
4.20.7. Gérer les conflits durant la fusion. ....	110
4.20.8. Fusionner une branche complétée .....	111
4.20.9. Branche de maintenance d'une fonctionnalité .....	112
4.21. Verrouiller .....	112
4.21.1. Comment le verrouillage fonctionne dans Subversion .....	113
4.21.2. Obtenir un verrou .....	113
4.21.3. Retirer un verrou .....	114
4.21.4. Vérifier le statut des verrous .....	115
4.21.5. Mettre les fichiers non verrouillés en Lecture seule .....	115
4.21.6. Les scripts hook de verrouillage .....	116
4.22. Créer et appliquer des patches .....	116
4.22.1. Créer un patch .....	116
4.22.2. Appliquer un patch .....	117
4.23. Qui a changé quelle ligne ? .....	117
4.23.1. Annoter pour les fichiers .....	118
4.23.2. Annoter les différences .....	120
4.24. l'explorateur de référentiel .....	120
4.25. Graphiques de révision .....	123
4.25.1. Graphiques des révisions .....	124
4.25.2. Changer l'affichage .....	124
4.25.3. Utiliser le Graphique de révisions .....	126
4.25.4. Refraîchissement de l'affichage .....	127
4.25.5. Visualisation des arborescences .....	127
4.26. Exporter une copie de travail Subversion .....	128
4.26.1. Retirer une copie de travail du contrôle de version .....	129
4.27. Relocaliser une copie de travail .....	129
4.28. Intégration avec des systèmes de gestion de bug / gestion d'incidents .....	130
4.28.1. Ajouter des numéros d'incidents aux messages de log .....	131
4.28.2. Récupérer des informations depuis le gestionnaire d'incidents .....	133
4.29. Intégration avec des explorateur de référentiel de type web. ....	134
4.30. Configuration de TortoiseSVN .....	135
4.30.1. Configuration générale .....	135
4.30.2. Options du Graphe des Révisions .....	143
4.30.3. Configuration du recouvrement d'icônes .....	146
4.30.4. Configuration du réseau .....	149

---

4.30.5. Réglages des programmes externes .....	151
4.30.6. Configuration des données sauvegardées .....	154
4.30.7. Mise en Cache des messages de log .....	155
4.30.8. Scripts hook côté client .....	159
4.30.9. Configuration de TortoiseBlame .....	163
4.30.10. Réglages dans le registre .....	163
4.30.11. Dossiers de travail de Subversion .....	165
4.31. Étape Finale .....	165
5. Le programme SubWCRev .....	166
5.1. La ligne de commande SubWCRev .....	166
5.2. Substitution de mot-clés .....	167
5.3. Exemple de mot-clé .....	167
5.4. Interface COM .....	168
6. IBUGtraqProvider interface .....	171
6.1. L'interface de IBUGtraqProvider .....	171
6.2. L'interface de IBUGtraqProvider2 .....	172
A. Foire aux questions (FAQ) .....	175
B. Comment faire pour... .....	176
B.1. Déplacer/copier beaucoup de fichiers en une fois .....	176
B.2. Forcer les utilisateurs à entrer un commentaire .....	176
B.2.1. Script hook sur le serveur .....	176
B.2.2. Propriétés de projet .....	177
B.3. Mettre à jour les fichiers sélectionnés à partir du référentiel .....	177
B.4. Annuler des révisions dans le référentiel .....	177
B.4.1. Utiliser la boîte de dialogue du journal de révision .....	177
B.4.2. Utiliser la boîte de dialogue fusionner .....	177
B.4.3. Utiliser svndumpfilter .....	178
B.5. Compare deux révisions d'un fichier ou d'un répertoire .....	178
B.6. Inclure un sous-projet commun .....	178
B.6.1. Utiliser svn:externals .....	179
B.6.2. Utiliser une copie de travail nichée .....	179
B.6.3. Utiliser un emplacement relatif .....	179
B.7. Créer un raccourci vers un référentiel .....	180
B.8. Ignorer les fichiers déjà versionnés .....	180
B.9. Retirer une copie de travail du contrôle de version .....	180
B.10. Retirer une copie de travail .....	180
C. Trucs Utiles Pour Les Administrateurs .....	181
C.1. Déployer TortoiseSVN via les stratégies de groupe .....	181
C.2. Rediriger la vérification de mise à niveau .....	181
C.3. Mettre la variable d'environnement SVN_ASP_DOT_NET_HACK .....	182
C.4. Désactiver les entrées du menu contextuel .....	182
D. Automatiser TortoiseSVN .....	185
D.1. Commandes de TortoiseSVN .....	185
D.2. Commandes de TortoiseIDiff .....	188
E. Référence croisée de l'interface en ligne de commande .....	190
E.1. Conventions et règles de base .....	190
E.2. Commandes de TortoiseSVN .....	190
E.2.1. Extraire .....	190
E.2.2. Mettre à jour .....	190
E.2.3. Mettre à jour à la révision .....	191
E.2.4. Livrer .....	191
E.2.5. Voir les différences .....	191
E.2.6. Voir le journal .....	192
E.2.7. Vérifier les modifications .....	192
E.2.8. Graphique de révision .....	192
E.2.9. Explorateur de référentiel .....	192
E.2.10. Éditer les conflits .....	193
E.2.11. Résolu .....	193

---

E.2.12. Renommer .....	193
E.2.13. Supprimer .....	193
E.2.14. Revenir en arrière .....	193
E.2.15. Nettoyer .....	193
E.2.16. Obtenir un verrou .....	193
E.2.17. Relâcher un verrou .....	194
E.2.18. Branche/Etiquette .....	194
E.2.19. Aller sur... ..	194
E.2.20. Fusionner .....	194
E.2.21. Exporter .....	194
E.2.22. Relocaliser .....	195
E.2.23. Créer un référentiel ici .....	195
E.2.24. Ajouter .....	195
E.2.25. Importer .....	195
E.2.26. Annoter .....	195
E.2.27. Ajouter à la liste des ignorés .....	195
E.2.28. Créer un patch .....	195
E.2.29. Appliquer un patch .....	195
F. Détails de l'implémentation .....	196
F.1. Recouvrement d'icônes .....	196
G. Sécuriser Svnserve grâce à SSH .....	198
G.1. Configurer un Serveur Linux .....	198
G.2. Configurer un Serveur Windows .....	198
G.3. Client SSH à utiliser avec TortoiseSVN .....	199
G.4. Création des certificats OpenSSH .....	199
G.4.1. Créer des clés en utilisant ssh-keygen .....	199
G.4.2. Créer des clés en utilisant PuTTYgen .....	199
G.5. Tester en utilisant PuTTY .....	200
G.6. Tester SSH avec TortoiseSVN .....	200
G.7. Variantes de Configuration SSH .....	201
Glossaire .....	203
Index .....	207

---

## Liste des illustrations

2.1. Un système Client/Serveur typique .....	5
2.2. Le problème à éviter .....	6
2.3. La solution Verrouiller-Modifier-Déverrouiller .....	7
2.4. La solution Copier-Modifier-Fusionner .....	8
2.5. ...Suite du modèle Copier-Modifier-Fusionner .....	8
2.6. Le système de fichiers du référentiel .....	10
2.7. Le référentiel .....	12
3.1. Le menu TortoiseSVN pour les dossiers non versionnés .....	15
4.1. L'Explorateur montrant le recouvrement d'icônes .....	37
4.2. Menu contextuel pour un répertoire sous contrôle de version .....	38
4.3. Menu fichier de l'Explorateur pour un raccourci dans un répertoire non versionné .....	39
4.4. Menu pour un répertoire sous contrôle de version lors d'un glisser-déplacer avec le bouton droit .....	40
4.5. Boîte de dialogue d'authentification .....	41
4.6. La boîte de dialogue Importer .....	43
4.7. La boîte de dialogue Extraire .....	45
4.8. La boîte de dialogue Livrer .....	47
4.9. Le vérificateur d'orthographe de la boîte de dialogue Livrer .....	50
4.10. La boîte de dialogue de progression montrant une livraison en cours .....	51
4.11. la boîte de dialogue de progression montrant une mise à jour terminée .....	52
4.12. L'Explorateur montrant le recouvrement d'icônes .....	58
4.13. Vérifier les modifications .....	60
4.14. Fenêtre de livraison avec les listes de modification .....	63
4.15. La boîte de dialogue du Journal de révision .....	65
4.16. Le panneau supérieur de la boîte de dialogue du Journal de révision avec le menu contextuel .....	66
4.17. Menu contextuel du panneau supérieur avec 2 révisions sélectionnées .....	69
4.18. Le panneau inférieur de la boîte de dialogue du Journal avec le menu contextuel .....	70
4.19. La Fenêtre du Journal de révision Montrant les Révisions Fusionnées .....	72
4.20. Histogramme de livraisons par auteur .....	74
4.21. Camembert de livraisons par auteur .....	75
4.22. Graphique de livraisons par date .....	76
4.23. La boîte de dialogue Comparer les révisions .....	79
4.24. Le visualiseur de différences d'images .....	80
4.25. Menu contextuel de l'explorateur pour les fichiers non versionnés .....	82
4.26. Menu pour un répertoire sous contrôle de version lors d'un glisser-déplacer avec le bouton droit .....	83
4.27. Menu contextuel de l'explorateur pour les fichiers non versionnés .....	84
4.28. Menu contextuel de l'explorateur pour les fichiers non versionnés .....	86
4.29. La boîte de dialogue Revenir en arrière .....	90
4.30. Page de propriétés de l'explorateur, onglet Subversion .....	92
4.31. Page de propriété de subversion .....	93
4.32. Ajouter des propriétés .....	94
4.33. La boîte de dialogue Branche/Étiquette .....	102
4.34. La boîte de dialogue Aller sur .....	104
4.35. Assistant de fusion - Réintégration .....	107
4.36. Assitant de de fusion - Fusion d'arborescence .....	108
4.37. La boîte de dialogue de conflit de fusion .....	111
4.38. La boîte de dialogue de fusion .....	112
4.39. La boîte de dialogue Verrouiller .....	114
4.40. La boîte de dialogue Vérifier les modifications .....	115
4.41. La boîte de dialogue Créer un patch .....	116
4.42. La boîte de dialogue Annoter .....	118
4.43. TortoiseBlame .....	119
4.44. l'explorateur de référentiel .....	121

---

4.45. Un graphique de révision .....	123
4.46. La fenêtre extraction-depuis-une-URL .....	128
4.47. La boîte de dialogue Relocaliser .....	129
4.48. Exemple de fenêtre de gestionnaire d'incidents .....	134
4.49. La boîte de dialogue Configuration, page Général .....	136
4.50. La boîte de dialogue de Configuration, Page Menu Contextuel .....	138
4.51. La boîte de dialogue Configuration, page Boîtes de dialogue 1 .....	139
4.52. La boîte de dialogue Configuration, page Boîtes de dialogue 2 .....	141
4.53. La boîte de dialogue Configuration, page Couleurs .....	142
4.54. La boîte de dialogue Configuration, page graphique de révision .....	143
4.55. La boîte de dialogue Configuration, page des couleur du graphe de révision .....	144
4.56. La Boîte de Dialogue Configuration, Page des Icônes de Recouvrement .....	146
4.57. La boîte de dialogue Configuration, page Ensemble d'icônes .....	149
4.58. La boîte de dialogue Configuration, page Réseau .....	149
4.59. La boîte de dialogue Configuration, page Visualisateur de différence .....	151
4.60. La boîte de dialogue Configuration, Boîte de dialogue Comparaison/fusion avancée .....	153
4.61. La boîte de dialogue Configuration, Page Données sauvegardées .....	154
4.62. La boîte de dialogue de Configuration, Page de Mise en Cache des Logs .....	156
4.63. La Fenêtre de propriétés, Statistiques d'Utilisation de la Mémoire Cache .....	158
4.64. La boîte de dialogue Configuration, page Scripts hook .....	159
4.65. La fenêtre de paramétrage, configuration des scripts de hook .....	160
4.66. La Fenêtre de Propriétés, Page d'Intégration d'un Gestionnaire d'Incidents .....	162
4.67. La boîte de dialogue ce configuration, page de bannissement. ....	163
C.1. La boîte de dialogue Mettre à jour .....	181

---

## Liste des tableaux

2.1. URL d'accès au référentiels .....	11
3.1. Réglages du <code>httpd.conf</code> d'Apache .....	28
5.1. Liste des commutateurs de ligne de commande disponibles .....	166
5.2. Liste des commutateurs de ligne de commande disponibles .....	167
5.3. Les méthodes COM/automation sont supportées .....	168
C.1. Entrées du menu et leurs valeurs .....	182
D.1. Liste des commandes et des options disponibles .....	186
D.2. Liste des options disponibles .....	188

---

# Préface



# TortoiseSVN

- Travaillez-vous dans une équipe ?
- Vous est-il jamais arrivé de travailler sur un fichier, et quelqu'un d'autre travaillait sur le même fichier au même moment ? Avez-vous perdu vos changements de ce fichier à cause de cela ?
- Avez-vous jamais sauvegardé un fichier, et ensuite voulu annuler les changements effectués ? Avez-vous jamais souhaité voir à quoi ressemblait un fichier il y a quelques temps ?
- Avez-vous jamais trouvé un bug dans votre projet et voulu savoir quand ce bug a été introduit dans vos fichiers ?

Si vous avez répondu « oui » à l'une de ces questions, alors TortoiseSVN est fait pour vous ! Continuez seulement cette lecture pour découvrir comment TortoiseSVN peut vous aider dans votre travail. Ce n'est pas si difficile.

## 1. Public

This book is written for computer literate folk who want to use Subversion to manage their data, but are uncomfortable using the command line client to do so. Since TortoiseSVN is a windows shell extension it's assumed that the user is familiar with the windows explorer and knows how to use it.

## 2. Guide de lecture

This [Préface](#) explains a little about the TortoiseSVN project, the community of people who work on it, and the licensing conditions for using it and distributing it.

Cette [Chapitre 1, Introduction](#) explique ce qu'est TortoiseSVN, ce qu'il fait, ses origines et les bases pour l'installer sur votre PC.

Dans [Chapitre 2, Concepts de base du contrôle de version](#) nous donnons une courte introduction au système de contrôle de révision *Subversion* qui est à la base de TortoiseSVN. C'est emprunté à la documentation du projet Subversion et cela explique les différentes approches au contrôle de version et comment Subversion fonctionne.

Le chapitre [Chapitre 3, Le référentiel](#) explique comment mettre en place un référentiel local, ce qui est utile pour tester Subversion et TortoiseSVN en utilisant un seul ordinateur. Il explique aussi un peu l'administration d'un référentiel qui est similaire pour les référentiels hébergés sur un serveur. Il ya également une section expliquant comment mettre en place un serveur.

La section [Chapitre 4, Guide d'utilisation quotidienne](#) est la plus importante puisqu'elle explique toutes les fonctionnalités principales de TortoiseSVN et comment les utiliser. Elle est sous forme de tutoriel qui indique comment extraire une copie de travail, la modifier, livrer les changements effectués, etc. Il traite de sujets de plus en plus avancés.

[Chapitre 5, Le programme SubWCRev](#) est un programme installé avec TortoiseSVN permettant d'extraire des informations de votre copie de travail et de les écrire dans un fichier. Ce qui est utile pour ajouter des informations de compilation à vos projets.

La section [Annexe B, Comment faire pour...](#) répond à quelques questions courantes concernant les tâches qui ne sont pas expliquées ailleurs.

La section *Annexe D, Automatiser TortoiseSVN* montre comment les boîtes de dialogue de TortoiseSVN peuvent être appelées en ligne de commande. C'est utile pour faire des scripts interactifs.

La *Annexe E, Référence croisée de l'interface en ligne de commande* fait le lien entre les commandes de TortoiseSVN et leurs équivalents en ligne de commande du client Subversion `svn` .exe.

### 3. TortoiseSVN est gratuit !

TortoiseSVN est gratuit. Vous n'avez pas à payer pour l'avoir, et vous pouvez l'utiliser comme vous le souhaitez. Il est développé sous licence GP (GPL).

TortoiseSVN is an Open Source project. That means you have full read access to the source code of this program. You can browse it on this link <http://code.google.com/p/tortoisesvn/source/browse/>. You will be prompted to enter username and password. The username is `guest`, and the password must be left blank. The most recent version (where we're currently working) is located under `/trunk/`, and the released versions are located under `/tags/`.

### 4. Communauté

Both TortoiseSVN and Subversion are developed by a community of people who are working on those projects. They come from different countries all over the world and work together to create wonderful programs.

### 5. Remerciements

Tim Kemp  
pour avoir fondé le projet TortoiseSVN

Stefan Küng  
for the hard work to get TortoiseSVN to what it is now

Lübbe Onken  
pour les belles icônes, le logo, la chasse aux bugs, la traduction et le soin apporté à la documentation

Simon Large  
for helping with the documentation and bug hunting

Le manuel de Subversion  
pour l'introduction grandiose à Subversion et son chapitre 2 que nous avons copié ici

Le projet Tigris Style  
pour certains des styles réutilisés dans cette documentation

Nos contributeurs  
for the patches, bug reports and new ideas, and for helping others by answering questions on our mailing list.

Nos donateurs  
pour les nombreuses heures de bonheur avec la musique qu'ils nous ont envoyé

### 6. Terminologie utilisée dans ce document

Pour rendre la lecture des documents plus facile, les noms de tous les écrans et des menus de TortoiseSVN sont marqués dans une police différente. La boîte de dialogue *Journal* par exemple.

Un choix de menu est indiqué par une flèche. TortoiseSVN → Voir le journal veut dire : sélectionnez *Voir le journal* à partir du menu contextuel *TortoiseSVN*.

Quand un menu contextuel local apparaît dans une des boîtes de dialogue de TortoiseSVN, on le montre comme cela : Menu contextuel → Enregistrer sous...

Les boutons de l'interface utilisateur sont indiqués comme ceci : Appuyer sur OK pour continuer.

Les actions utilisateur sont indiquées en utilisant une police en gras. **Alt+A** : appuyez sur la touche **A** en maintenant la touche **Alt** de votre clavier enfoncée. Glisser click droit : appuyez sur le bouton droit de la souris et en le maintenant enfoncé *faites glisser* les éléments vers le nouvel emplacement.

La sortie système et l'entrée au clavier sont indiquées aussi avec une police différente.



### **Important**

Les notes importantes sont marquées avec une icône.



### **Astuce**

Astuces pour vous rendre la vie plus facile.



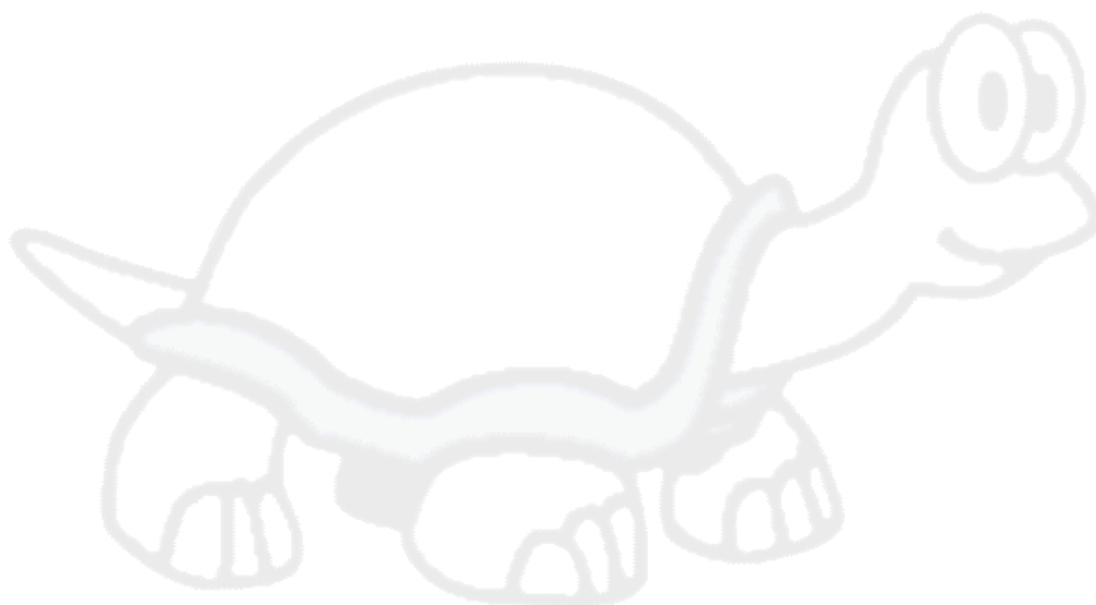
### **Attention**

Endroits où vous devez faire attention à ce que vous faites.



### **Avertissement**

Where extreme care has to be taken, data corruption or other nasty things may occur if these warnings are ignored.



---

# Chapitre 1. Introduction

Le contrôle de version est l'art de gérer les changements de l'information. Cela a longtemps été un outil critique pour les programmeurs, qui passent typiquement leur temps à faire de petites modifications au logiciel et à défaire ensuite ces changements le lendemain. Imaginez une équipe de ces programmeurs travaillant concurremment - et peut-être même simultanément sur les mêmes fichiers ! - et vous pouvez voir pourquoi un bon système est nécessaire pour *gérer le chaos potentiel*.

## 1.1. Qu'est-ce que TortoiseSVN ?

TortoiseSVN is a free open-source client for the *Subversion* version control system. That is, TortoiseSVN manages files and directories over time. Files are stored in a central *repository*. The repository is much like an ordinary file server, except that it remembers every change ever made to your files and directories. This allows you to recover older versions of your files and examine the history of how and when your data changed, and who changed it. This is why many people think of Subversion and version control systems in general as a sort of « time machine ».

Quelques systèmes de contrôle de version sont aussi des systèmes de gestion de configuration logicielle (GCL). Ces systèmes sont spécifiquement conçus pour gérer des arborescences de code source et ont beaucoup de fonctionnalités spécifiques au développement de logiciel - comme la compréhension de langages de programmation en natif, ou des outils d'approvisionnement pour construire le logiciel. Subversion, cependant, n'est pas un de ces systèmes ; c'est un système général qui peut être utilisé pour gérer *n'importe quelle* collection de fichiers, y compris du code source.

## 1.2. L'historique de TortoiseSVN

In 2002, Tim Kemp found that Subversion was a very good version control system, but it lacked a good GUI client. The idea for a Subversion client as a Windows shell integration was inspired by the similar client for CVS named TortoiseCVS.

Tim studied the source code of TortoiseCVS and used it as a base for TortoiseSVN. He then started the project, registered the domain `tortoisesvn.org` and put the source code online. During that time, Stefan Küng was looking for a good and free version control system and found Subversion and the source for TortoiseSVN. Since TortoiseSVN was still not ready for use then he joined the project and started programming. Soon he rewrote most of the existing code and started adding commands and features, up to a point where nothing of the original code remained.

As Subversion became more stable it attracted more and more users who also started using TortoiseSVN as their Subversion client. The user base grew quickly (and is still growing every day). That's when Lübke Onken offered to help out with some nice icons and a logo for TortoiseSVN. And he takes care of the website and manages the translation.

## 1.3. Les fonctionnalités de TortoiseSVN

Qu'est-ce qui fait de TortoiseSVN un si bon client Subversion ? Voici une courte liste des fonctionnalités.

Intégration dans le shell

TortoiseSVN integrates seamlessly into the Windows shell (i.e. the explorer). This means you can keep working with the tools you're already familiar with. And you do not have to change into a different application each time you need functions of the version control!

And you are not even forced to use the Windows Explorer. TortoiseSVN's context menus work in many other file managers, and in the File/Open dialog which is common to most standard Windows applications. You should, however, bear in mind that TortoiseSVN is intentionally developed as extension for the Windows Explorer. Thus it is possible that in other applications the integration is not as complete and e.g. the icon overlays may not be shown.

#### Recouvrement d'icônes

Le statut de chaque fichier et de chaque répertoire versionnés est indiqué par des petites icônes de revouvrement. De cette façon vous pouvez voir tout de suite quel est le statut de votre copie de travail.

#### Accès facile aux commandes de Subversion

Toutes les commandes de Subversion sont disponibles à partir du menu contextuel de l'explorateur. TortoiseSVN y ajoute son propre sous-menu.

Puisque TortoiseSVN est un client Subversion, nous voudrions aussi vous montrer certaines des fonctionnalités de Subversion :

#### Répertoires versionnés

CVS suit seulement à la trace l'histoire de fichiers individuels, mais Subversion met en oeuvre un système de fichiers « virtuel » versionné qui suit à la trace les changements sur des arborescences entières à travers le temps. Les fichiers *et* les répertoires sont versionnés. En conséquence, il y a du coté client de vraies commandes **déplacer** et **copier** qui fonctionnent sur les fichiers et les répertoires.

#### Livraisons atomiques

Une livraison va sur le référentiel complètement, ou pas du tout. Cela permet aux développeurs de construire et livrer les changements comme des morceaux logiques.

#### Metadonnées versionnées

Chaque fichier et chaque répertoire a un jeu invisible de « propriétés » attachées. Vous pouvez inventer et stocker n'importe quelle paire arbitraire clef/valeur que vous souhaitez. Les propriétés sont versionnées dans le temps, comme le contenu du fichier.

#### Choix de couches réseau

Subversion a une notion abstraite de l'accès au référentiel, le rendant facile à mettre en oeuvre à travers de nouveaux mécanismes de réseau. Le serveur réseau « avancé » de Subversion est un module pour le serveur Web Apache, qui utilise une variante de HTTP appelée WebDAV/DeltaV. Cela donne un grand avantage à Subversion en stabilité et en interopérabilité et fournit des différentes fonctionnalités clés gratuitement : authentification, autorisation, compression de fil et navigation de référentiel, par exemple. Un processus de serveur Subversion plus petit, autonome est aussi disponible. Ce serveur utilise un protocole personnalisé qui peut être facilement tunnelé par ssh.

#### Gestion cohérente des données

Subversion exprime les différences de fichier en utilisant un algorithme de différenciation binaire, qui travaille identiquement sur les fichiers textes (lisibles par l'homme) et les fichiers binaires (illisibles par l'homme). Les deux types de fichiers sont stockés également compressés dans le référentiel, et les différences sont transmises dans les deux directions à travers le réseau.

#### Embranchements et étiquetages efficaces

Le coût de l'embranchement et de l'étiquetage n'a pas besoin d'être proportionnel à la taille de projet. Subversion crée des branches et des étiquettes en copiant simplement le projet, en utilisant un mécanisme semblable à un lien dur. Ainsi ces opérations prennent seulement un temps très petit, constant et un espace très petit dans le référentiel.

#### Hackabilité

Subversion n'a aucun bagage historique ; il est mis en oeuvre comme une collection de bibliothèques C partagées avec des API bien définies. Cela fait que Subversion est extrêmement maintenable et utilisable par d'autres applications et d'autres langages.

## 1.4. Installer TortoiseSVN

### 1.4.1. Configuration requise

TortoiseSVN runs on Windows 2000 SP2, Windows XP or higher. Windows 98, Windows ME and Windows NT4 are no longer supported since TortoiseSVN 1.2.0, but you can still download the older versions if you really need them.

If you encounter any problems during or after installing TortoiseSVN please refer to [Annexe A, Foire aux questions \(FAQ\)](#) first.

## 1.4.2. Installation

TortoiseSVN comes with an easy to use installer. Double click on the installer file and follow the instructions. The installer will take care of the rest.



### Important

Vous devez être administrateur pour installer TortoiseSVN.

## 1.4.3. Packs de langue

The TortoiseSVN user interface has been translated into many different languages, so you may be able to download a language pack to suit your needs. You can find the language packs on our [translation status page](http://tortoisesvn.net/translation_status) [http://tortoisesvn.net/translation\_status]. And if there is no language pack available yet, why not join the team and submit your own translation ;-)

Each language pack is packaged as a .exe installer. Just run the install program and follow the instructions. Next time you restart, the translation will be available.

## 1.4.4. Vérificateur d'orthographe

TortoiseSVN inclut un vérificateur d'orthographe qui vous permet de vérifier vos commentaires de livraison. C'est particulièrement utile si le la langue du projet n'est pas votre langue maternelle. Le vérificateur d'orthographe utilise les mêmes fichiers de dictionnaire que [OpenOffice](http://openoffice.org) [http://openoffice.org] et [Mozilla](http://mozilla.org) [http://mozilla.org].

The installer automatically adds the US and UK English dictionaries. If you want other languages, the easiest option is simply to install one of TortoiseSVN's language packs. This will install the appropriate dictionary files as well as the TortoiseSVN local user interface. Next time you restart, the dictionary will be available too.

Ou vous pouvez installer les dictionnaires vous-même. Si vous avez OpenOffice ou Mozilla installés, vous pouvez copier ces dictionnaires, qui se trouvent dans les dossiers d'installation de ces applications. Autrement, vous devez télécharger les fichiers de dictionnaire requis depuis <http://wiki.services.openoffice.org/wiki/Dictionaries>

Une fois que vous avez les fichiers de dictionnaire, vous devez probablement les renommer pour que les noms de fichier contiennent seulement les caractères locaux. Exemple :

- fr\_FR.aff
- fr\_FR.dic

Ensuite copiez les simplement dans le sous-dossier bin du dossier d'installation de TortoiseSVN. Normalement ce sera C:\Program Files\TortoiseSVN\bin. Si vous ne voulez pas encombrer le sous-dossier bin, vous pouvez placer vos fichiers de vérification d'orthographe dans C:\Program Files\TortoiseSVN\Languages à la place. Si ce dossier n'est pas là, vous devez le créer d'abord. La prochaine fois vous démarrez TortoiseSVN, le vérificateur d'orthographe sera disponible.

Si vous installez plusieurs dictionnaires, TortoiseSVN utilise ces règles pour choisir lequel utiliser.

1. Vérifier le réglage `tsvn:projectlanguage`. Référez-vous à [Section 4.17, « Configuration des projets »](#) pour des informations concernant les propriétés de projet.

2. Si aucune langue de projet n'est indiquée, ou si cette langue n'est pas installée, essayer la langue correspondant aux options régionales de Windows.
3. Si les options régionales exactes de Windows ne marchent pas, essayer la langue de « Base », i.e. de\_CH (Allemand Suisse) devient de\_DE (Allemand).
4. Si aucune des règles ci-dessus ne marche, alors la langue par défaut est l'anglais, qui est inclus avec l'installation standard.

---

# Chapitre 2. Concepts de base du contrôle de version

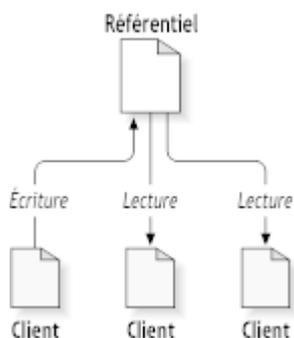
Ce chapitre est une version légèrement modifiée du même chapitre dans le manuel de Subversion. Une version en ligne du manuel de Subversion est disponible ici : <http://svnbook.red-bean.com/>.

Ce chapitre est une introduction courte, occasionnelle à Subversion. Si le contrôle de version est nouveau pour vous, ce chapitre est certainement pour vous. Nous commençons par une discussion sur les concepts généraux du contrôle de version, nous ferons notre chemin vers les idées spécifiques derrière Subversion et nous montrons quelques exemples simples de Subversion en utilisation.

Bien que les exemples dans ce chapitre montrent des gens partageant des collections de code source de programme, gardez à l'esprit que Subversion peut gérer n'importe quel sorte de collection de fichier - il n'est pas limité à l'aide de programmeurs.

## 2.1. Le référentiel

Subversion est un système centralisé pour partager l'information. Son coeur est un *référentiel*, qui est un dépôt central de données. Le référentiel stocke l'information sous forme d'une *arborescence de système de fichiers* - une hiérarchie typique de fichiers et de répertoires. N'importe quel nombre de *clients* se connecte au référentiel et ensuite lit ou écrit ces fichiers. En écrivant des données, un client rend l'information disponible aux autres ; en lisant des données, le client reçoit l'information des autres.



**Figure 2.1. Un système Client/Serveur typique**

Alors pourquoi est-ce si intéressant ? Jusqu'ici, cela ressemble à la définition d'un serveur de fichiers typique. Et en effet, le référentiel *est* une sorte de serveur de fichiers, mais il n'est pas de votre genre habituel. Ce qui rend le référentiel de Subversion spécial est qu'il *se rappelle de chaque changement* jamais écrit : chaque changement de chaque fichier, et même les changements de l'arborescence des répertoires elle-même, comme l'ajout, la suppression et le réarrangement des fichiers et des répertoires.

Quand un client lit des données du référentiel, il voit normalement seulement la dernière version de l'arborescence. Mais le client peut aussi voir les états *précédents*. Par exemple, un client peut poser des questions historiques comme, « que contenait ce répertoire mercredi dernier ? », ou « qui sont les dernières personnes à avoir changé ce fichier et quels changements ont-elles fait ? » C'est ce genre de questions qui sont au coeur de n'importe quel *système de contrôle de version* : des systèmes qui sont conçus pour enregistrer et suivre les changements des données au cours du temps.

## 2.2. Modèles de gestion de version

Tous les systèmes de contrôle de version doivent résoudre le même problème fondamental : comment le système permettra-t-il aux utilisateurs de partager l'information, mais les empêchera accidentellement de

se marcher sur les pieds ? Il est trop facile pour les utilisateurs d'écraser accidentellement les changements de chacun sur le référentiel.

### 2.2.1. Le problème du partage de fichier

Considérons ce scénario : supposons que nous avons deux collaborateurs, Harry et Sally. Ils décident chacun d'éditer le même fichier du référentiel en même temps. Si Harry sauvegarde ses changements sur le référentiel d'abord, il est possible qu'ensuite (quelques moments plus tard) Sally puisse accidentellement les écraser avec sa propre nouvelle version du fichier. Tandis que la version d'Harry du fichier ne sera pas perdue pour toujours (parce que le système se rappelle chaque changement), les changements qu'Harry a fait *ne seront pas* dans la version plus récente du fichier de Sally, parce qu'elle n'a jamais vu les changements d'Harry pour commencer. Le travail d'Harry est effectivement encore perdu - ou du moins manquant de la dernière version du fichier - et probablement par accident. C'est certainement une situation que nous voulons éviter!

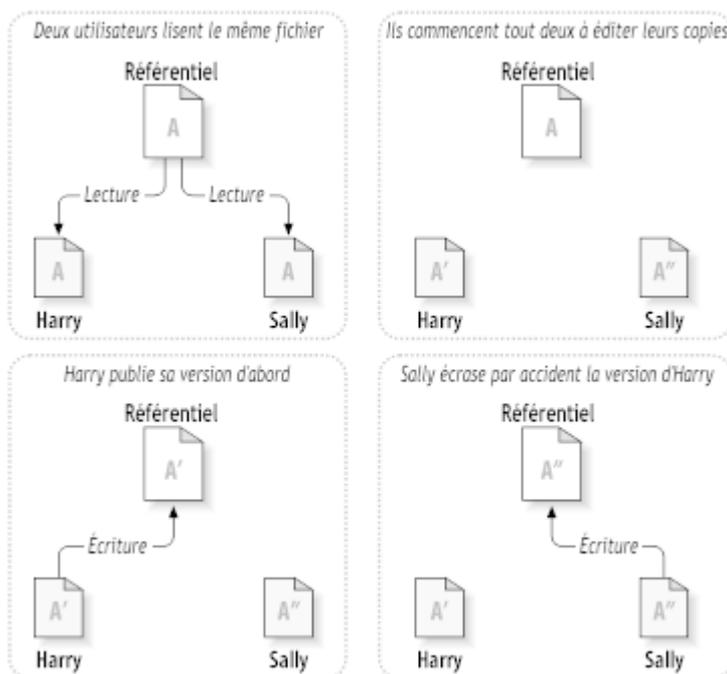
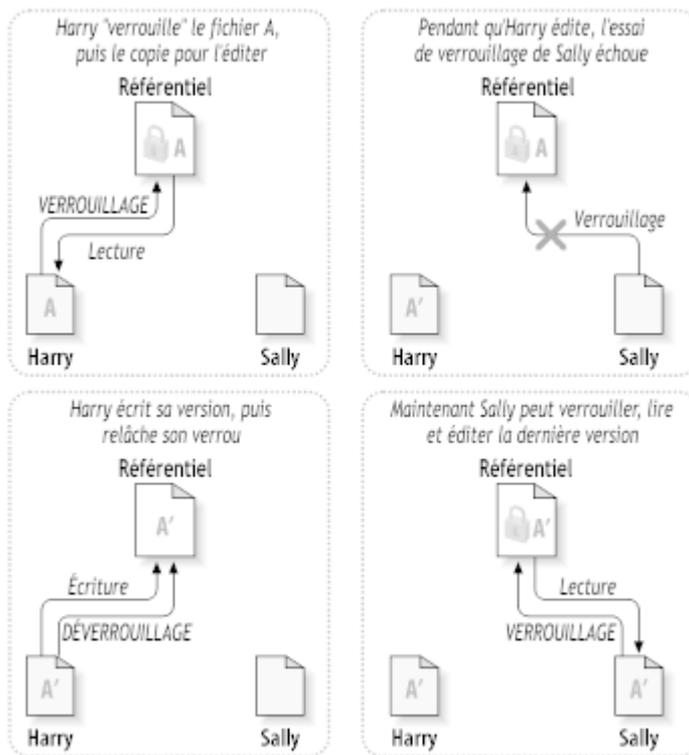


Figure 2.2. Le problème à éviter

### 2.2.2. La solution Verrouiller-Modifier-Déverrouiller

Beaucoup de systèmes de contrôle de version utilisent le modèle *verrouiller-modifier-déverrouiller* pour aborder ce problème, qui est une solution très simple. Dans un tel système, le référentiel ne permet qu'à une seule personne de changer un fichier à la fois. D'abord Harry doit *verrouiller* le fichier avant qu'il ne puisse commencer à y faire des changements. Le verrouillage d'un fichier s'apparente alors à l'emprunt d'un livre dans une bibliothèque ; si Harry a verrouillé un fichier, alors Sally ne peut pas le modifier. Si elle essaye de verrouiller le fichier, le référentiel refusera la requête. Tout qu'elle peut faire est lire le fichier et attendre qu'Harry finisse ses changements et relâche le verrou. Dès qu'Harry déverrouille le fichier, son tour est fini, et Sally peut alors prendre son tour en le verrouillant et en y apportant ses modifications.



**Figure 2.3. La solution Verrouiller-Modifier-Déverrouiller**

Le problème avec le modèle "verrouiller-modifier-déverrouiller" est qu'il est un peu restrictif et devient souvent un barrage pour les utilisateurs :

- *Le verrouillage peut causer des problèmes administratifs.* Parfois Harry verrouillera un fichier et ensuite l'oubliera. En attendant, parce que Sally attend toujours pour éditer le fichier, ses mains sont liées. Et ensuite Harry va en vacances. Maintenant Sally doit appeler un administrateur pour relâcher le verrou d'Harry. La situation se finit en causant beaucoup de retard inutile et de temps gaspillé.
- *Le verrouillage peut causer une sérialisation inutile.* Et si Harry édite le début d'un fichier texte et Sally veut simplement éditer la fin du même fichier ? Ces changements ne se chevauchent pas du tout. Ils pourraient facilement éditer le fichier simultanément et aucun grand mal n'arriverait, à supposer que les changements aient été correctement fusionnés ensemble. Ils n'ont aucun besoin de prendre leur tour dans cette situation.
- *Le verrouillage peut créer un faux sentiment de sécurité.* Disons qu'Harry verrouille et édite le fichier A, tandis que Sally verrouille et édite le fichier B en même temps. Mais supposons qu'A et B dépendent l'un de l'autre et les changements faits à chacun sont sémantiquement incompatibles. Soudainement A et B ne marchent désormais plus ensemble. Le système de verrouillage était impuissant à empêcher le problème - encore il a fourni d'une façon ou d'une autre un sentiment de fausse sécurité. C'est facile pour Harry et Sally de s'imaginer qu'en verrouillant les fichiers, chacun commence une tâche sûre, isolée et les interdit ainsi de discuter de leurs changements incompatibles dès le début.

### 2.2.3. La solution Copier-Modifier-Fusionner

Subversion, CVS et les autres systèmes de contrôle de version utilisent un modèle *copier-modifier-fusionner* comme une alternative au verrouillage. Dans ce modèle, le client de chaque utilisateur lit le référentiel et crée une *copie de travail* personnelle du fichier ou du projet. Les utilisateurs travaillent alors en parallèle, modifiant leurs copies privées. Finalement, les copies privées sont fusionnées ensemble dans une version nouvelle, finale. Le système de contrôle de version aide souvent avec la fusion, mais en fin de compte un être humain est responsable pour qu'elle se produise correctement.

Voici un exemple. Disons qu'Harry et Sally ont chacun une copie de travail du même projet, extrait du référentiel. Ils travaillent en même temps et modifient localement le même fichier A. Sally sauvegarde

ses changements dans le référentiel d'abord. Quand Harry essaie de sauvegarder ses changements, le référentiel l'informe que son fichier A est *périmé*. Autrement dit, dans le référentiel, ce fichier A a été modifié depuis qu'il a été extrait. Donc Harry demande à son client de *fusionner* les nouveaux changements du fichier A du référentiel avec sa copie de travail. Il y a des chances que les changements de Sally ne se chevauchent pas avec les siens ; ainsi une fois qu'il a les deux changements intégrés, il sauvegarde sa copie de travail sur le référentiel.

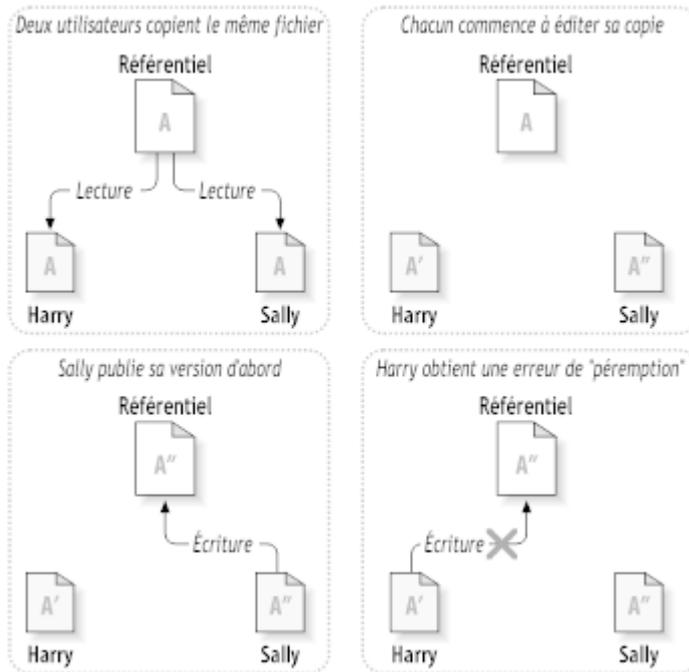


Figure 2.4. La solution Copier-Modifier-Fusionner

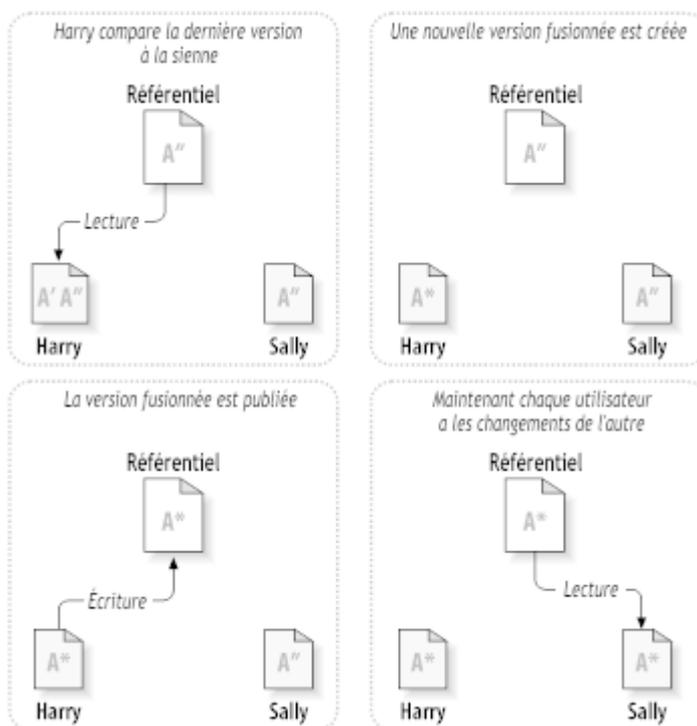


Figure 2.5. ...Suite du modèle Copier-Modifier-Fusionner

Mais que se passe-t-il si les changements de Sally *chevauchent* les changements d'Harry ? Que se passe-t-il alors ? Cette situation est appelée un *conflit* et habituellement, ce n'est pas vraiment un problème. Quand Harry demande à son client de fusionner les derniers changements du référentiel vers sa copie de travail, sa copie du fichier A est marquée d'une façon ou d'une autre comme étant dans un état de conflit : il sera capable de voir les deux jeux de changements en conflit et choisira manuellement entre eux. Notez que le logiciel ne peut pas résoudre automatiquement les conflits ; seuls les êtres humains sont capables de comprendre et de faire les choix intelligents nécessaires. Une fois qu'Harry a manuellement résolu les changements se chevauchant (peut-être en discutant du conflit avec Sally !), il peut sans risque sauvegarder le fichier fusionné sur le référentiel.

Le modèle copier-modifier-fusionner peut sembler un peu chaotique, mais en pratique, il fonctionne de manière extrêmement fluide. Les utilisateurs peuvent travailler en parallèle, n'ayant jamais à s'attendre. Quand ils travaillent sur les mêmes fichiers, il s'avère que la plupart de leurs changements simultanés ne se chevauchent pas du tout ; les conflits sont peu fréquents. Et le temps que cela prend pour résoudre des conflits est moindre que le temps perdu par un système de verrouillage.

À la fin, tout cela se réduit à un facteur critique : la communication entre les utilisateurs. Quand les utilisateurs communiquent mal, les conflits tant syntaxiques que sémantiques augmentent. Aucun système ne peut forcer les utilisateurs à communiquer parfaitement et aucun système ne peut détecter les conflits sémantiques. Ainsi il n'y a aucune raison d'être apaisé par une fausse promesse qu'un système de verrouillage empêchera d'une façon ou d'une autre des conflits ; en pratique, le verrouillage semble inhiber la productivité plus qu'autre chose.

Il y a une situation commune où le modèle verrouiller-modifier-déverrouiller s'en sort mieux et c'est quand vous avez des fichiers qui ne sont pas fusionnables. Par exemple, si votre référentiel contient quelques images graphiques et deux personnes modifient une image en même temps, il n'y a aucun moyen de fusionner ces changements. Soit Harry, soit Sally perdra ses changements.

#### 2.2.4. Que fait Subversion ?

Subversion utilise la solution copier-modifier-fusionner par défaut et dans des nombreux cas c'est tout ce dont vous aurez jamais besoin. Cependant, à partir de la version 1.2, Subversion supporte aussi le verrouillage de fichier, si vous avez des fichiers non-fusionnables, ou si vous êtes simplement forcés à une politique de verrouillage par le management, Subversion fournira encore les fonctionnalités dont vous avez besoin.

### 2.3. Subversion en action

#### 2.3.1. Copies de travail

Vous a déjà lu au sujet des copies de travail ; maintenant nous allons vous montrer comment le client Subversion les crée et les utilise.

Une copie de travail de Subversion est une arborescence de répertoires ordinaire sur votre système local, contenant une collection de fichiers. Vous pouvez éditer ces fichiers comme vous le souhaitez, et si ce sont des fichiers de code source, vous pouvez compiler votre programme de la façon habituelle. Votre copie de travail est votre propre secteur de travail privé : Subversion n'incorporera jamais les changements d'autres personnes, ni ne rendra vos propres changements disponibles aux autres, jusqu'à ce que vous lui disiez explicitement de le faire.

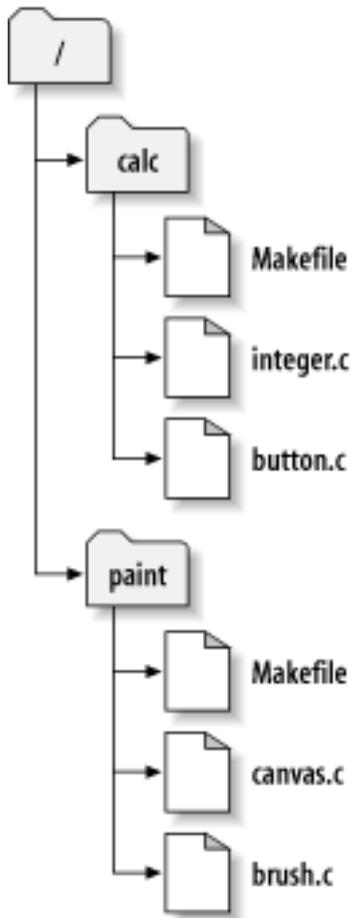
Après avoir fait des changements dans votre copie de travail et avoir vérifié qu'ils fonctionnent correctement, Subversion vous fournit des commandes pour *publier* vos changements et ainsi les rendre disponibles aux autres personnes travaillant avec vous sur le projet. Si d'autres personnes publient leurs changements dans le référentiel, Subversion vous fournit des commandes pour fusionner ces changements dans votre répertoire de travail.

Une copie de travail contient aussi quelques fichiers supplémentaires, créés et entretenus par Subversion, pour l'aider à effectuer ces commandes. Particulièrement chaque répertoire dans votre copie de travail

contient un sous-répertoire nommé `.svn`, aussi connu comme le *répertoire administratif* de la copie de travail. Les fichiers de chaque répertoire administratif aident Subversion à reconnaître quels fichiers contiennent des changements non publiés et quels fichiers sont périmés par rapport au travail des autres.

Un référentiel typique de Subversion contient souvent les fichiers (ou le code source) pour plusieurs projets ; d'habitude, chaque projet est un sous-répertoire dans l'arborescence du système de fichiers du référentiel. Dans cette optique, la copie de travail d'un utilisateur correspondra d'habitude à un sous-arbre particulier du référentiel.

Par exemple, supposons que vous avez un référentiel contenant deux projets d'application.



**Figure 2.6. Le système de fichiers du référentiel**

En d'autres termes, la racine du référentiel a deux sous-répertoires : `paint` et `calc`.

Pour obtenir une copie de travail, vous devez *extraire* une sous-arborescence du référentiel. (Le terme *extraire* peut faire penser à un quelconque verrouillage ou à une réservation des ressources, mais il n'en est rien ; il crée simplement une copie privée du projet pour vous).

Supposons que vous faites des changements sur `button.c`. Puisque le répertoire `.svn` se rappelle la date de modification et le contenu original du fichier, Subversion peut dire que vous avez modifié le fichier. Cependant, Subversion ne rend pas vos changements publics jusqu'à ce que vous le lui disiez explicitement. L'acte de publier vos changements est plus communément appelé *livrer* (ou *valider*) les changements au référentiel.

Pour publier vos changements aux autres, vous pouvez utiliser la commande de Subversion **livrer**.

Maintenant vos changements sur `button.c` ont été livrés au référentiel ; si un autre utilisateur extrait une copie de travail de `/calc`, il verra vos changements dans la dernière version du fichier.

Supposons que vous avez une collaboratrice, Sally, qui a extrait une copie de travail de `/calc` en même temps que vous. Quand vous livrez votre changement sur `button.c`, la copie de travail de Sally est laissée inchangée ; Subversion modifie seulement les copies de travail à la demande de l'utilisateur.

Pour actualiser son projet, Sally peut demander à Subversion de *mettre à jour* sa copie de travail, en utilisant la commande de Subversion **mettre à jour**. Cela incorporera vos changements dans sa copie de travail, en même temps que d'autres qui ont été livrés depuis qu'elle l'a extrait.

Notez que Sally n'a pas dû spécifier les fichiers à mettre à jour; Subversion utilise l'information dans le répertoire `.svn` et la nouvelle information dans le référentiel, pour décider quels fichiers doivent être actualisés.

### 2.3.2. URL de référentiel

Les référentiels de Subversion peuvent être accédés par beaucoup de méthodes différentes - sur le disque local, ou par des protocoles de réseau divers. Un emplacement de référentiel, cependant, est toujours une URL. Le schéma d'URL indique la méthode d'accès :

Schéma	Méthode d'accès
<code>file://</code>	Accès direct au référentiel sur disque local ou réseau.
<code>http://</code>	Accès via le protocole WebDAV à un serveur Apache avec Subversion.
<code>https://</code>	Même chose que <code>http://</code> , mais avec cryptage SSL.
<code>svn://</code>	Accès TCP/IP non authentifié via un protocole personnalisé à un serveur <code>svnserve</code> .
<code>svn+ssh://</code>	Accès TCP/IP authentifié, crypté via un protocole personnalisé à un serveur <code>svnserve</code> .

#### Tableau 2.1. URL d'accès au référentiels

Pour la plupart, les URL de Subversion utilisent la syntaxe standard, autorisant la définition des noms serveur et des numéros de port dans l'URL. La méthode d'accès `file://` est normalement utilisée pour les accès locaux, bien qu'elle puisse être utilisée avec des chemins UNC pour un hôte non local. L'URL prend donc la forme `file://nomhote/chemin/vers/referentiel`. Pour la machine locale, la partie `nomhote` de l'URL doit être soit absente, soit `localhost`. Pour cette raison, les chemins locaux apparaissent normalement avec trois slashes, `file:///chemin/vers/referentiel`.

Ainsi, les utilisateurs du système de fichier `file:` sur les plate-formes Windows devront utiliser une syntaxe « standard » non-officielle pour avoir accès aux référentiels qui sont sur la même machine, mais sur un disque différent du disque de travail. Les deux syntaxes de d'URL suivantes marcheront ; X est le disque sur lequel est hébergé le référentiel :

```
file:///X:/chemin/vers/référentiel
...
file:///X|/chemin/vers/référentiel
...
```

Notez qu'une URL utilise des slashes ordinaires bien que la forme native d'un chemin (non-URL) utilise des antislashes sous Windows.

Vous pouvez sans risque avoir accès à un référentiel FSFS via un partage réseau, mais vous *ne pouvez pas* avoir accès à un référentiel BDB de cette façon.



#### Avertissement

Ne créez pas ou n'accédez pas à un référentiel Berkeley DB sur un partage réseau. Il *ne peut pas* exister sur un système de fichiers distant. Même si vous faites mapper le disque

réseau à une lettre de disque. Si vous essayez d'utiliser Berkeley DB sur un partage réseau, les résultats sont imprévisibles - vous pouvez voir des erreurs mystérieuses tout de suite, ou cela peut s'écouler des mois avant que vous ne découvriez que votre base de données du référentiel est subtilement corrompue.

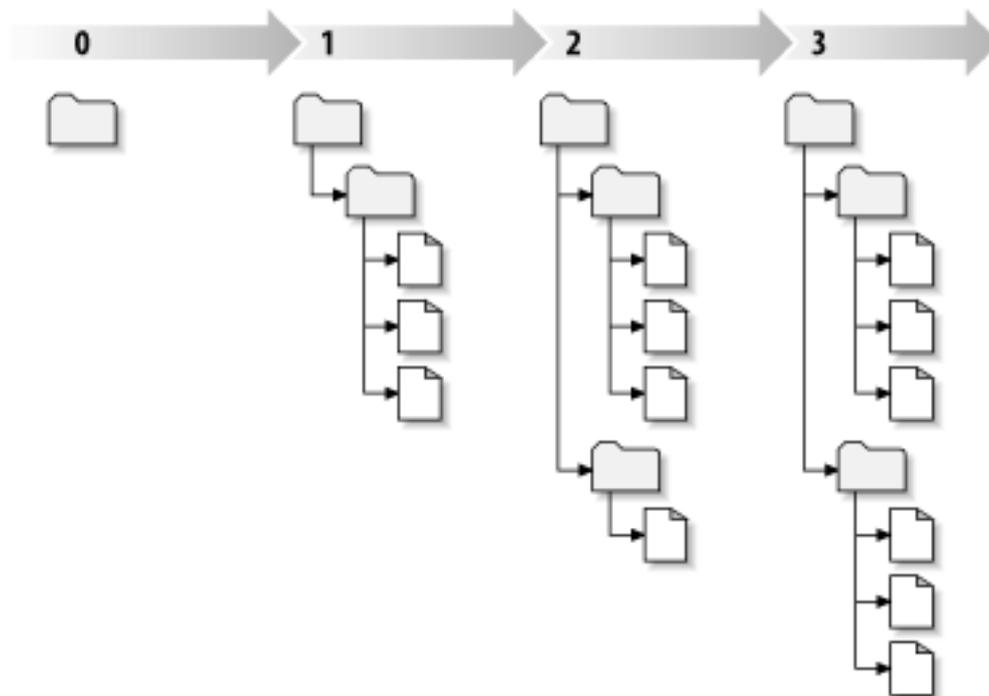
### 2.3.3. Révisions

Une opération **svn commit** peut publier les changements de n'importe quel nombre de fichiers et de répertoires comme une seule transaction atomique. Dans votre copie de travail, vous pouvez changer le contenu des fichiers, créer, supprimer, renommer et copier des fichiers et des répertoires et ensuite livrer le jeu complet de changements comme une unité.

Dans le référentiel, chaque livraison est traitée comme une transaction atomique : tous les changements de la livraison ont lieu, ou aucun n'a lieu. Subversion essaye de conserver cette atomicité face aux plantages du programme, pannes système, problèmes de réseau et autres actions utilisateur.

Chaque fois que le référentiel accepte une livraison, cela crée un nouvel état de l'arborescence du système de fichiers, appelé une *révision*. À chaque révision est assignée un entier naturel unique, plus grand que le numéro de la révision précédente d'une unité. La révision initiale d'un référentiel récemment créé est numérotée zéro et ne consiste en rien d'autre qu'un répertoire racine vide.

Une façon agréable de visualiser le référentiel est une série d'arbres. Imaginez un tableau de numéros de révision, commençant à 0, s'étirant de gauche à droite. Chaque numéro de révision a un arbre du système de fichiers s'accrochant au-dessous de lui et chaque arbre est un « instantané » de la façon à laquelle le référentiel ressemble après chaque livraison.



**Figure 2.7. Le référentiel**

#### Numéros de révision globaux

À la différence de beaucoup d'autres systèmes de contrôle de version, les numéros de révision de Subversion s'appliquent à *toute l'arborescence*, et non à des fichiers individuels. Chaque numéro de révision sélectionne un arbre entier, un état particulier du référentiel après un certain changement livré. Une autre façon d'y penser est que cette révision N représente l'état du système de fichiers du

référentiel après que la Nième livraison. Quand un utilisateur de Subversion parle de ``la révision 5 de foo.c'', il veut dire vraiment ``foo.c comme il apparaît à la révision 5.'' Remarquez qu'en général, les révisions N et M d'un fichier ne diffèrent *pas* nécessairement !

Il est important de noter que les copies de travail ne correspondent pas toujours à une seule révision dans le référentiel ; elles peuvent contenir des fichiers de plusieurs révisions différentes. Par exemple, supposez que vous extrayiez une copie de travail d'un référentiel dont la révision la plus récente est 4 :

```
calc/Makefile:4
    integer.c:4
    button.c:4
```

À l'heure actuelle, ce répertoire de travail correspond exactement à la révision 4 dans le référentiel. Cependant, supposez que vous faites un changement sur `button.c` et que vous livrez ce changement. En admettant qu'aucune autre livraison n'ait eu lieu, votre livraison créera la révision 5 du référentiel, et votre copie de travail ressemblera maintenant à cela :

```
calc/Makefile:4
    integer.c:4
    button.c:5
```

Supposons que, à ce point, Sally livre une modification sur `integer.c`, créant la révision 6. Si vous utilisez **svn update** pour actualiser votre copie de travail, elle ressemblera alors à ceci :

```
calc/Makefile:6
    integer.c:6
    button.c:6
```

Les changements de Sally sur `integer.c` apparaîtront dans votre copie de travail et votre changement seront toujours présent dans `button.c`. Dans cet exemple, le texte `Makefile` est identique dans des révisions 4, 5 et 6, mais Subversion marquera votre copie de travail de `Makefile` avec la révision 6 pour indiquer qu'il est toujours d'actualité. Ainsi, après avoir fait une mise à jour propre au sommet de votre copie de travail, elle correspondra généralement exactement à une révision dans le référentiel.

### 2.3.4. Comment les copies de travail suivent le référentiel

Pour chaque fichier dans un répertoire de travail, Subversion enregistre deux informations essentielles dans le secteur administratif `.svn/` :

- sur quelle révision votre fichier de travail est basé (cela s'appelle la *révision de travail* du fichier), et
- un enregistrement d'horodatage quand la copie locale a été mise à jour en dernier par le référentiel.

Avec ces informations, en parlant au référentiel, Subversion peut dire dans lequel des quatre états suivants se trouve un fichier de travail :

#### Inchangé et courant

Le fichier est inchangé dans le répertoire de travail et aucun changement sur ce fichier n'a été livré au référentiel depuis sa révision de travail. Une **livraison** du fichier ne fera rien et une **mise à jour** du fichier ne fera rien.

#### Changé localement et courant

Le fichier a été changé dans le répertoire de travail et aucun changement sur ce fichier n'a été livré au référentiel depuis sa révision de base. Il y a des changements locaux qui n'ont pas été livrés au référentiel, ainsi une **livraison** du fichier réussira à publier vos changements et une **mise à jour** du fichier ne fera rien.

#### Inchangé et périmé

Le fichier n'a pas été changé dans le répertoire de travail, mais il a été changé dans le référentiel. Le fichier devrait être mis à jour éventuellement, pour l'actualiser avec la révision publique. Une **livraison** du fichier ne fera rien et une **mise à jour** du fichier intègrera les derniers changements dans votre copie de travail.

#### Changé localement et périmé

Le fichier a été modifié dans le répertoire de travail, et dans le référentiel. Une **livraison** du fichier échouera avec une erreur *périmé*. Le fichier devrait d'abord être mis à jour ; la commande **mise à jour** essaiera de fusionner les deux versions. Si Subversion réussit pas à faire correctement la fusion, il laisse l'utilisateur résoudre le conflit.

## 2.4. Résumé

Nous avons couvert un certain nombre de concepts fondamentaux de Subversion dans ce chapitre :

- Nous avons présenté les notions du référentiel central, la copie de travail du client et le tableau d'arbres de révision de référentiel.
- Nous avons vu quelques exemples simples comment deux collaborateurs peuvent utiliser Subversion pour publier et recevoir des changements l'un de l'autre, en utilisant le modèle "copier-modifier-fusionner".
- Nous avons un peu parlé de la façon dont Subversion suit à la trace et gère l'information dans une copie de travail.

---

# Chapitre 3. Le référentiel

Peu importe le protocole que vous utilisez pour avoir accès à vos référentiels, vous devez toujours créer au moins un référentiel. Cela peut être fait soit avec le client de ligne de commande de Subversion soit avec TortoiseSVN.

Si vous n'avez pas encore créé de référentiel Subversion, il est temps de le faire maintenant.

## 3.1. Création de référentiel

Vous pouvez créer un référentiel avec le backend FSFS ou avec le format de base de données Berkeley (BDB) plus ancien. Le format de FSFS est plus rapide et plus facile à gérer, il fonctionne maintenant sur les partages réseau et sous Windows 98 sans problème. Le format BDB est considéré comme plus stable car il est utilisé depuis plus longtemps ; cependant, FSFS est maintenant utilisé depuis plusieurs années, donc cet argument est dépassé. Lisez [Choosing a Data Store](http://svnbook.red-bean.com/en/1.5/svn.reposadmin.planning.html#svn.reposadmin.basics.backends) [http://svnbook.red-bean.com/en/1.5/svn.reposadmin.planning.html#svn.reposadmin.basics.backends] dans le manuel de Subversion pour plus d'informations.

### 3.1.1. Créer un référentiel avec le client de ligne de commande

1. Créez un dossier vide avec le nom SVN (par exemple D:\SVN\), qui est utilisé comme racine pour tous vos référentiels.
2. Créez un autre dossier MonNouveauRéférentiel dans D:\SVN\.
3. Ouvrez l'invite de commande (ou la boîte DOS), allez vers D:\SVN\ et tapez

```
svnadmin create --fs-type bdb MonNouveauRéférentiel
```

ou

```
svnadmin create --fs-type fsfs MonNouveauRéférentiel
```

Maintenant vous avez un nouveau référentiel situé à D:\SVN\MonNouveauRéférentiel.

### 3.1.2. Créer le référentiel avec TortoiseSVN

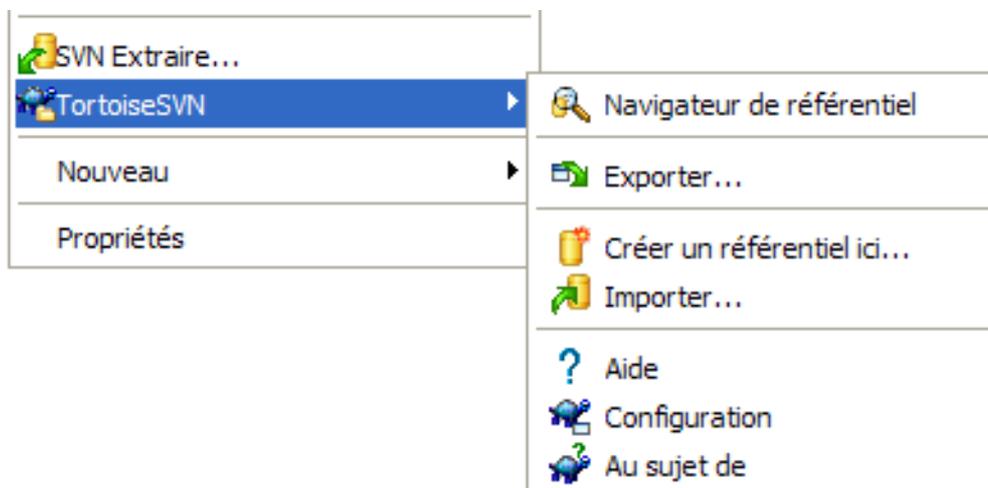


Figure 3.1. Le menu TortoiseSVN pour les dossiers non versionnés

1. Ouvrez l'explorateur Windows
2. Créez un nouveau dossier et nommez-le par exemple `SVNréférentiel`
3. Faites un clic droit sur le dossier nouvellement créé et sélectionnez TortoiseSVN → Créer un référentiel ici...

Un référentiel est alors créé à l'intérieur du nouveau dossier. *N'écrivez pas ces fichiers vous-même !!!* Si vous avez des erreurs assurez vous que le dossier est vide et qu'il n'est pas protégé en écriture.



### Astuce

TortoiseSVN ne permet plus de créer des référentiels BDB, cependant vous pouvez toujours le faire en ligne de commande. Les référentiels FSFS sont généralement plus aisés à maintenir et, pour des raisons de compatibilité entre les différentes versions de BDB, nous permet de maintenir plus aisément TortoiseSVN.

Future versions of TortoiseSVN will not support `file://` access to BDB repositories due to these compatibility issues, although it will of course always support this repository format when accessed via a server through the `svn://`, `http://` or `https://` protocols. For this reason, we strongly recommend that any new repository which must be accessed using `file://` protocol is created as FSFS.

Bien sûr, nous recommandons également de ne pas utiliser d'accès de type `file://`, hormis à des fins de test. Utiliser un serveur est plus sécurisé et plus fiable pour toute utilisation mutualisée.

### 3.1.3. Accès local au référentiel

Pour accéder à votre référentiel local vous avez besoin du chemin vers ce dossier. Souvenez-vous juste que Subversion s'attend à des chemins de référentiels dans la forme `file:///C:/SVNréférentiel/`. Notez l'utilisation de slashes.

Pour avoir accès à un référentiel placé sur une partage réseau vous pouvez soit utiliser le mappage de disque, ou vous pouvez utiliser le chemin UNC. Pour les chemins UNC, la forme est `file://NomDuServeur/chemin/vers/référentiel/`. Notez qu'il y a seulement 2 slashes au début ici.

Avant SVN 1.2, les chemins UNC devaient être donnés dans la forme plus obscure `file:///\\NomDuServeur/chemin/vers/référentiel` Cette forme est toujours supportée, mais n'est pas recommandée.



### Avertissement

Ne créez pas ou n'accédez pas à un référentiel Berkeley DB sur un partage réseau. Il *ne peut pas* exister sur un système de fichiers distant. Même si vous mappez le disque réseau sur une lettre de disque. Si vous essayez d'utiliser Berkeley DB sur un partage réseau, les résultats seront imprévisibles - vous pouvez voir des erreurs mystérieuses tout de suite, ou il peut s'écouler des mois avant que vous ne découvriez que votre base de données du référentiel est corrompue.

### 3.1.4. Accéder à un référentiel situé dans un partage réseau

Au moins en théorie, il est possible d'utiliser un référentiel FSFS sur un partage réseau en utilisant le protocole `file://`, ce n'est certainement *pas* recommandé. En fait, nous le déconseillons *fortement*, et ne supportons pas un tel usage.

Premièrement vous donnez le droit d'accès en écriture dans le référentiel à tous les utilisateurs, de manière à ce qu'aucun ne puisse supprimer ou rendre complètement inutilisable de quelque manière que ce soit le référentiel.

Ensuite, tous les protocoles d'échange de fichiers ne supportent pas le verouillage dont Subversion a besoin, donc votre référentiel peut être corrompu. Ca ne devrait pas arriver tout de suite, mais un jour deux utilisateurs vont essayer d'accéder au référentiel en même temps.

Troisièmement, les droits des fichiers doivent être fixés exactement de cette manière. Vous pouvez vous en tirer sur un partage Windows, mais SAMBA est particulièrement compliqué.

Les accès de type `file://` sont réservés à des fins de test et de debug ou pour une utilisation mono-utilisateur. Lorsque vous voulez partager votre référentiel vous devez *vraiment* mettre en place un vrai serveur, et ce n'est pas aussi compliqué qu'il n'y paraît. Lisez [Section 3.5, « Accéder au référentiel »](#) pour guider votre choix.

### 3.1.5. Disposition du référentiel

Avant que vous n'importiez vos données dans le référentiel vous devriez d'abord penser à la façon dont vous voulez organiser vos données. Si vous utilisez une des dispositions recommandées, ce sera beaucoup plus facile plus tard.

Il y a quelques standards, des manières recommandées d'organiser un référentiel. La plupart des personnes créent un répertoire `trunk` contenant la « version principale » de développement, un répertoire `branches` qui contient les copies de travail parallèles et un répertoire `tags` qui contient les versions stables. Si un référentiel ne contient qu'un seul projet, ces répertoires sont créés à la base du référentiel :

```
/trunk
/branches
/tags
```

Si un référentiel contient de multiples projets, les gens indexent souvent leur disposition par branche :

```
/trunk/paint
/trunk/calc
/branches/paint
/branches/calc
/tags/paint
/tags/calc
```

...ou par projet:

```
/paint/trunk
/paint/branches
/paint/tags
/calc/trunk
/calc/branches
/calc/tags
```

L'indexation par projet a un sens si les projets ne sont pas étroitement liés et si chacun est extrait individuellement. Pour des projets liés où vous pouvez vouloir extraire tous les projets en une fois, ou où les projets sont tous liés ensemble dans un unique package de distribution, il est souvent mieux d'indexer par branche. De cette façon vous avez seulement un tronc à extraire et les relations entre les sous-projets sont plus facilement visibles.

Si vous adoptez une approche de niveau supérieur `/trunk /tags /branches`, il va sans dire que vous devez copier le tronc en entier pour chaque branche et chaque étiquette et d'une certaine façon, cette structure offre le plus de flexibilité.

Pour les projets sans rapport vous pouvez préférer utiliser des référentiels séparés. Quand vous livrez des changements, c'est le numéro de révision du référentiel entier qui change, pas le numéro de révision du projet. Avoir 2 projets sans rapport qui partagent un référentiel peut signifier de grands écarts dans les numéros de révision. Les projets Subversion et TortoiseSVN apparaissent à la même adresse hôte, mais sont des référentiels complètement séparés permettant le développement indépendant et aucune confusion sur les numéros de génération.

Bien sûr, vous êtes libre d'ignorer ces dispositions communes. Vous pouvez créer n'importe quel sorte de variation, ce qui marche le mieux pour vous ou votre équipe. Rappelez-vous que quoi que vous choisissiez, ce n'est pas un engagement permanent. Vous pouvez réorganiser votre référentiel à tout moment. Parce que les branches et les étiquettes sont des répertoires ordinaires, TortoiseSVN peut les déplacer ou les renommer comme vous le souhaitez.

Passer d'une disposition à une autre ne revient qu'à exécuter une série de mouvements côté serveur ; si vous n'aimez pas la façon dont les choses sont organisées dans le référentiel, jonglez juste avec les répertoires.

Si vous n'avez pas encore créé la structure de base de votre référentiel, vous devriez le faire maintenant. Il y a deux manière de le faire. Si vous souhaitez simplement une structure `/trunk /tags /branches`, vous pouvez utiliser l'explorateur de référentiel pour créer les trois répertoires (dans trois livraisons différentes). Si vous souhaitez créer une arborescence plus complexe, il sera plus simple de le faire directement sur le disque puis de l'importer dans une seule livraison, comme suit:

1. créez un nouveau répertoire vide sur votre disque dur
2. créez votre structure de dossier de niveau supérieure désirée à l'intérieur de ce dossier - n'y mettez pas encore de fichiers !
3. import this structure into the repository via a right click on the folder and selecting TortoiseSVN → Import... This will import your temp folder into the repository root to create the basic repository layout.

Notez que le nom du dossier que vous importez n'apparaît pas dans le référentiel, seulement son contenu. Par exemple, créez la structure de dossier suivante :

```
C:\Temp\Nouveau\trunk
C:\Temp\Nouveau\branches
C:\Temp\Nouveau\tags
```

Importez `C:\Temp\Nouveau` dans la racine du référentiel, qui ressemblera alors à cela :

```
/trunk
/branches
/tags
```

## 3.2. Sauvegarde de référentiel

Quel que soit le type de référentiel que vous utilisez, il est extrêmement important que vous maintenez des sauvegardes régulières et que vous vérifiez la sauvegarde. Si le serveur tombe, vous pouvez être capable d'avoir accès à une version récente de vos fichiers, mais sans le référentiel tout votre historique est perdu pour toujours.

Le moyen le plus simple (mais non recommandé) est juste de copier le dossier du référentiel sur le médium de secours. Cependant, vous devez être absolument sûr qu'aucun processus n'ait accès aux données. Dans ce contexte, accès veut dire *pas* d'accès du tout. Un référentiel BDB est écrit même quand l'opération semble seulement exiger la lecture, comme l'obtention du statut. Si votre référentiel est accédé pendant la copie, (navigateur Internet laissé ouvert, WebSVN, etc) la sauvegarde sera sans valeur.

La méthode recommandée est d'exécuter

```
svnadmin hotcopy chemin/vers/référentiel chemin/vers/sauvegarde --clean-logs
```

pour créer une copie de votre référentiel d'une manière sûre. Sauvegardez alors la copie. L'option `--clean-logs` n'est pas exigée, mais supprime les fichiers de journal superflus quand vous sauvegardez un référentiel BDB, qui peut économiser un certain espace.

L'outil `svnadmin` est installé automatiquement quand vous installez le client de ligne de commande de Subversion. Si vous installez les outils de ligne de commande sur un PC Windows, la meilleure façon est de télécharger la version d'installateur Windows. Il est compressé plus efficacement que la version `.zip`, donc le téléchargement est plus petit et il s'occupe de configurer les chemins pour vous. Vous pouvez télécharger la dernière version du client de ligne de commande de Subversion à partir de <http://subversion.apache.org/getting.html>.

### 3.3. Scripts de hook côté serveur

Un script hook est un programme déclenché par un certain événement du référentiel, comme la création d'une nouvelle révision ou la modification d'une propriété non versionnée. Chaque hook fournit suffisamment d'informations pour dire à quel événement il se rapporte, sur quelle(s) cible(s) il doit fonctionner et le nom utilisateur de la personne qui a déclenché l'événement. Selon le résultat du hook ou le statut retourné, le programme hook peut continuer l'action, l'arrêter, ou la suspendre d'une certaine façon. Veuillez vous référer au chapitre sur *Scripts de Hook* [<http://svnbook.red-bean.com/en/1.5/svn.reposadmin.create.html#svn.reposadmin.create.hooks>] dans le manuel de Subversion pour plus de détails sur la façon dont les hooks sont mis en place.

Ces scripts de hook sont exécutés par le serveur qui héberge le référentiel. TortoiseSVN permet également de configurer des scripts de hook à exécuter côté client en réponse à certains événements. Voir [Section 4.30.8, « Scripts hook côté client »](#) pour plus d'information.

Des exemples de scripts hook peuvent être trouvés dans le répertoire `hooks` du référentiel. Ces exemples de scripts sont appropriés pour des serveurs Unix/Linux, mais doivent être modifiés si votre serveur est basé sur Windows. Le hook peut être un fichier batch ou un exécutable. L'exemple ci-dessous montre un fichier batch qui pourrait être utilisé pour mettre en oeuvre un hook de `pre-revprop-change`.

```
rem Autorise seulement les changements sur les commentaires.
if "%4" == "svn:log" exit 0
echo Property '%4' ne peut pas être modifié >&2
exit 1
```

Notez que tout ce qui est envoyé sur `stdout` est abandonné. Si vous voulez qu'un message apparaisse dans la boîte de dialogue du rejet de la livraison vous devez l'envoyer sur `stderr`. Dans un fichier batch, c'est réalisé en utilisant `>&2`

### 3.4. Liens d'extraction

Si vous voulez donner l'accès à votre référentiel Subversion à d'autres utilisateurs, vous pouvez vouloir mettre un lien vers celui-ci sur votre site Web. Une façon de rendre ceci plus accessible est d'inclure un *lien d'extraction* pour les autres utilisateurs de TortoiseSVN.

Quand vous installez TortoiseSVN, il enregistre un nouveau protocole `tsvn:`. Quand un utilisateur de TSVN clique sur un lien de ce type, la boîte de dialogue d'extraction s'ouvrira automatiquement avec l'URL du référentiel déjà remplie.

Pour inclure un tel lien dans votre page html, vous devez ajouter du code de ce type :

```
<a href="tsvn:http://nom.de.domaine.du.project.org/svn/trunk">
</a>
```

Bien sûr ce sera encore mieux si vous avez inclus une image appropriée. Vous pouvez utiliser le *logo de Tortoise SVN* [<http://tortoisesvn.tigris.org/images/TortoiseCheckout.png>] ou vous pouvez vous servir de votre propre image.

```
<a href="tsvn:http://nom.de.domaine.du.projet.org/svn/trunk">
<img src=TortoiseCheckout.png></a>
```

Vous pouvez aussi faire pointer le raccourci vers une révision spécifique, par exemple

```
<a href="tsvn:http://nom.de.domaine.du.projet.org/svn/trunk?100">
</a>
```

## 3.5. Accéder au référentiel

Pour utiliser TortoiseSVN (ou un autre client Subversion), vous avez besoin d'un hébergement pour vos référentiels. Vous pouvez soit les stocker localement et y accéder en utilisant le protocole `file://`, soit les placer sur un serveur et y accéder avec les protocoles `http://` ou `svn://`. Ces deux protocoles peuvent aussi être cryptés. Vous utiliserez alors `https://`, `svn+ssh://`, ou encore `svn://` avec SASL.

Si votre serveur est chez un hébergeur public comme *Google Code* [<http://code.google.com/hosting/>] ou qu'il a déjà été configuré entièrement par quelqu'un d'autre. Allez à la page [Chapitre 4, Guide d'utilisation quotidienne](#).

Si vous n'avez pas de serveur, si vous travaillez seul et/ou si vous souhaitez juste tester Subversion et TortoiseSVN, alors les référentiels locaux sont probablement la meilleure solution. Créez juste un référentiel sur votre ordinateur comme décrit dans [Chapitre 3, Le référentiel](#). Vous pouvez sauter le reste de ce chapitre et aller directement au [Chapitre 4, Guide d'utilisation quotidienne](#) pour savoir comment commencer à l'utiliser.

Mettre en place un référentiel destiné à être utilisé par plusieurs utilisateurs sur un partage réseau n'est pas conseillé. Lisez [Section 3.1.4, « Accéder à un référentiel situé dans un partage réseau »](#) pour savoir pourquoi nous pensons que ce n'est pas une bonne idée. Mettre en place un serveur n'est pas aussi compliqué qu'il n'y paraît, sera beaucoup plus sûr et probablement plus rapide.

Les sections suivantes sont des guides pas à pas d'installation de ce type de serveur sur une machine Windows. Bien sûr vous pouvez aussi l'installer sur une machine Linux, mais c'est en dehors du contexte de ce guide. Des informations plus détaillées sur les options du serveur Subversion et sur la manière de faire le bon choix d'architecture sont disponibles dans le livre Subversion, dans la section [Configuration du Serveur](#) [<http://svnbook.red-bean.com/en/1.5/svn.serverconfig.html>].

## 3.6. Serveur basé sur Svnserve

### 3.6.1. Introduction

Subversion est livré avec Svnserve - un serveur stand-alone léger qui utilise une variante du protocole TCP/IP pour se connecter. C'est la solution idéale pour de petites installations, ou si un bon gros Apache ne peut être utilisé.

Dans la plupart des cas svnserve est plus facile à installer et fonctionne plus rapidement que le serveur utilisant Apache, cela dit il y manque quelques fonctionnalités. D'autant plus maintenant qu'est inclus le support SASL.

### 3.6.2. Installer svnserve

1. Récupérez la dernière version de Subversion à cet endroit <http://subversion.apache.org/getting.html>. Vous pouvez également récupérer le programme d'installation prépackagé chez CollabNet ici : <http://www.collab.net/downloads/subversion>. Ce programme d'install mettra en place svnserve comme service Windows, et inclura quelques outils utiles si vous comptez utiliser SASL pour sécuriser votre serveur.
2. Si vous déjà une version de Subversion d'installée, et que svnserve est lancé, vous devrez l'arrêter avant de continuer.
3. Exécutez le programme d'installation de Subversion. Si vous l'exécutez sur votre serveur (ce qui est recommandé) vous pouvez passer l'étape 4.
4. Ouvrez l'explorateur Windows, allez au répertoire d'installation de Subversion (habituellement C:\Program Files\Subversion) et dans le répertoire bin, trouvez les fichiers `svnserve.exe`, `intl3_svn.dll`, `libapr.dll`, `libapriconv.dll`, `libapriutil.dll`, `libdb*.dll`, `libeay32.dll` et `ssleay32.dll` - copiez ces fichiers, ou copiez simplement tout le contenu du répertoire bin, dans un répertoire sur votre serveur, par exemple `c:\svnserve`

### 3.6.3. Exécuter svnserve

Maintenant que svnserve est installé, vous avez besoin qu'il tourne sur votre serveur. L'approche la plus simple est d'exécuter ce qui suit à partir d'un interpréteur de commandes DOS ou de créer un raccourci Windows :

```
svnserve.exe --daemon
```

svnserve démarrera maintenant en attendant des requêtes entrantes sur le port 3690. Le commutateur `--daemon` indique à svnserve de fonctionner comme un processus de démon, donc il existera toujours jusqu'à ce qu'il soit arrêté manuellement.

Si vous n'avez pas encore créé de référentiel, suivez les instructions données avec le paramétrage du serveur Apache [Section 3.7.4, « Configuration »](#).

Pour tester que svnserve fonctionne, utilisez TortoiseSVN → Explorateur de référentiel pour voir un référentiel.

En supposant que votre référentiel est placé dans `c:\repos\TestRepo` et votre serveur est appelé `localhost`, entrer :

```
svn://localhost/repos/TestRepo
```

quand l'explorateur de référentiel le demande.

Vous pouvez aussi augmenter la sécurité et gagner du temps en entrant l'URL avec svnserve en utilisant le commutateur `--root` pour indiquer l'emplacement racine et limiter l'accès à un répertoire spécifique sur le serveur :

```
svnserve.exe --daemon --root disque:\chemin\vers\racine\référentiel
```

En utilisant le test précédent comme guide, svnserve fonctionnerait maintenant comme :

```
svnserve.exe --daemon --root c:\référentiels
```

Et dans TortoiseSVN notre URL de l'explorateur de référentiel est maintenant raccourcie en :

```
svn://localhost/TestRéférentiel
```

Notez que le commutateur `--root` est aussi nécessaire si votre référentiel est placé sur une partition ou un disque différent de l'emplacement de `svnserve` sur votre serveur.

`Svnserve` s'occupera de plusieurs référentiels. Indiquez juste leur emplacement dans le répertoire racine, et accédez y en utilisant des chemins relatifs à la racine.



### Avertissement

Ne créez pas ou n'accédez pas à un référentiel Berkeley DB sur un partage réseau. Il *ne peut pas* exister sur un système de fichiers distant. Même si vous faites mapper le disque réseau à une lettre de disque. Si vous essayez d'utiliser Berkeley DB sur un partage réseau, les résultats sont imprévisibles - vous pouvez voir des erreurs mystérieuses tout de suite, ou cela peut s'écouler des mois avant que vous ne découvriez que votre base de données du référentiel est subtilement corrompue.

#### 3.6.3.1. Exécuter `svnserve` en tant que service

Exécuter `svnserve` par un utilisateur n'est généralement pas la meilleure solution. Cela revient à avoir toujours un utilisateur connecté sur votre serveur et vous rappeler de le relancer après un redémarrage. Une meilleure manière consiste à lancer `svnserve` en tant que service Windows. À partir de Subversion 1.4, `svnserve` peut être installé comme un service windows natif.

To install `svnserve` as a native windows service, execute the following command all on one line to create a service which is automatically started when windows starts.

```
sc create svnserve binpath= "c:\svnserve\svnserve.exe --service
--root c:\repos" displayname= "Subversion" depend= tcpip
start= auto
```

If any of the paths include spaces, you have to use (escaped) quotes around the path, like this:

```
sc create svnserve binpath= "
\"C:\Program Files\Subversion\bin\svnserve.exe\"
--service --root c:\repos" displayname= "Subversion"
depend= tcpip start= auto
```

You can also add a description after creating the service. This will show up in the Windows Services Manager.

```
sc description svnserve "Subversion server (svnserve)"
```

Notez la ligne de commande relativement atypique utilisée par `sc`. Dans la paire clé= valeur il ne doit pas y avoir d'espace entre la clé et le = mais il doit y en avoir un avant la valeur.



### Astuce

Microsoft recommande maintenant que les services soient exécutés soit par le compte Service Local soit par le compte Service Réseau. Référez-vous *The Services and Service Accounts Security Planning Guide* [<http://www.microsoft.com/technet/security/topics/serversecurity/serviceaccount/default.mspx>]. Pour créer le service sous le compte Service Local, ajoutez ce qui suit à l'exemple ci-dessus.

```
obj= "NT AUTHORITY\LocalService"
```

Notez que vous devriez donner au compte Service Local les droits appropriés tant pour Subversion et vos référentiels que pour toute application utilisée par les scripts hook. Le groupe intégré pour cela est appelé "SERVICE LOCAL".

A partir du moment où vous avez installé le service, vous devez aller dans la fenêtre de gestion des services pour le démarrer (juste cette fois ci, il se lancera ensuite automatiquement après le redémarrage du serveur).

Pour plus d'informations, consultez le document *Windows Service Support for Svnserve* [<http://svn.collab.net/repos/svn/trunk/notes/windows-service.txt>].

Si vous aviez une ancienne version de svnservice utilisant le wrapper `SVNService`, et que vous voulez maintenant utiliser le support natif, vous devez désinscrire le wrapper comme service (n'oubliez pas d'arrêter d'abord le service !). Utilisez simplement la commande

```
svnservice -remove
```

pour retirer l'entrée dans la base de registre.

### 3.6.4. Authentification de base avec svnservice

L'installation par défaut de svnservice fournit un accès anonyme en lecture seule. Cela signifie que vous pouvez utiliser une URL du type `svn://` pour extraire, mettre à jour, ou utiliser l'explorateur de référentiel dans TortoiseSVN pour voir le référentiel, mais vous ne pourrez faire aucune livraison.

Pour permettre l'accès en écriture à un référentiel, vous devez éditer le fichier `conf/svnservice.conf` dans votre répertoire de référentiel. Ce fichier contrôle la configuration du démon svnservice et contient aussi la documentation utile.

Vous pouvez activer l'accès en écriture anonyme en mettant simplement :

```
[general]
anon-access = write
```

Cependant, vous ne saurez pas qui a fait des changements à un référentiel, puisque la propriété `svn:author` sera vide. Vous serez aussi incapables de contrôler qui fait des changements à un référentiel. C'est un paramétrage quelque peu risqué !

Une façon de surmonter cela est de créer une base de données de mot de passe :

```
[general]
anon-access = none
auth-access = write
password-db = fichier_utilisateur
```

Où `fichier_utilisateur` est un fichier qui existe dans le même répertoire que `svnservice.conf`. Ce fichier peut vivre ailleurs dans votre système de fichiers (utile pour quand vous avez plusieurs référentiels qui exigent les mêmes droits d'accès) et peuvent être référencés en utilisant un chemin absolu, ou un chemin relatif au répertoire `conf`. Si vous incluez un chemin, il doit être écrit de `/la/manière/unix`. L'utilisation de `\` ou des lettres de disque ne marchera pas. Le `fichier_utilisateur` devrait avoir une structure de :

```
[users]
nom_utilisateur = mot_de_passe
...
```

Cet exemple interdira tout accès aux utilisateurs non authentifiés (anonymes) et donnera l'accès en lecture-écriture aux utilisateurs inscrits dans `fichier_utilisateur`.



### Astuce

Si vous gérez plusieurs référentiels avec la même base de données de mots de passe, l'utilisation d'un groupe d'identification facilitera la tâche des utilisateurs, dans la mesure où TortoiseSVN peut garder vos accréditations dans la mémoire cache vous n'avez à les saisir qu'une seule fois. Plus d'informations sont disponibles dans le manuel de Subversion dans les sections [Create a 'users' file and realm](http://svnbook.red-bean.com/en/1.5/svn.serverconfig.svnservice.html#svn.serverconfig.svnservice.auth.users) [http://svnbook.red-bean.com/en/1.5/svn.serverconfig.svnservice.html#svn.serverconfig.svnservice.auth.users] and [Client Credentials](http://svnbook.red-bean.com/en/1.5/svn.serverconfig.netmodel.html#svn.serverconfig.netmodel.credcache) [Caching](http://svnbook.red-bean.com/en/1.5/svn.serverconfig.netmodel.html#svn.serverconfig.netmodel.credcache) [http://svnbook.red-bean.com/en/1.5/svn.serverconfig.netmodel.html#svn.serverconfig.netmodel.credcache]

## 3.6.5. Sécuriser le serveur avec SASL

### 3.6.5.1. Qu'est-ce que SASL ?

La couche d'authentification et de sécurisation Cyrus est open source et développée par l'université Carnegie Mellon. Elle étoffe les protocoles réseau de capacités d'authentification et de cryptage, et tout comme Subversion à partir de sa version 1.5, svnservice et TortoiseSVN savent utiliser de cette librairie.

Pour plus de détails sur les options disponibles, consultez la section [Using svnservice with SASL](http://svnbook.red-bean.com/en/1.5/svn.serverconfig.svnservice.html#svn.serverconfig.svnservice.sasl) [http://svnbook.red-bean.com/en/1.5/svn.serverconfig.svnservice.html#svn.serverconfig.svnservice.sasl] du manuel de référence de Subversion. Si vous recherchez un moyen simple de mettre en place un système d'authentification et de cryptage sur un serveur Windows, de manière à pouvoir accéder à votre référentiel depuis le gros méchant Internet, continuez à lire.

### 3.6.5.2. Authentification SASL

Pour activer le mécanisme spécifique à SASL sur le serveur, vous aurez besoin de faire trois choses. Premièrement, créer une section `[sasl]` dans le fichier `svnservice.conf` de votre référentiel, avec ce couple clé-valeur:

```
use-sasl = true
```

Ensuite, créez un fichier nommé `svn.conf` au bon endroit - typiquement dans le répertoire d'installation de Subversion.

Troisièmement, créer deux clés de registre pour indiquer à SASL où trouver les éléments. Créez une clé nommée `[HKEY_LOCAL_MACHINE\SOFTWARE\Carnegie Mellon\Project Cyrus\SASL Library]` et insérez y deux valeurs de type chaîne de caractère: `SearchPath` ayant comme valeur le chemin contenant le greffon `sasl*.dll` (qui est normalement le répertoire d'installation de Subversion), et `ConfFile` ayant comme valeur le chemin contenant le fichier `svn.conf`. Si vous avez utilisé le programme d'installation de CollabNet, ces clés auront déjà été créées pour vous.

Ajoutez les lignes suivantes au fichier `svn.conf` :

```
pwcheck_method: auxprop
auxprop_plugin: sasldb
mech_list: DIGEST-MD5
sasldb_path: C:\TortoiseSVN\sasldb
```

La dernière ligne contient le chemin vers la base de données d'authentification, qui est un fichier nommé `sasldb`. Elle peut aller n'importe où, mais un choix adapté est le répertoire parent du référentiel. Assurez-vous que le service `svnserve` a le droit de lire ce fichier.

Si `svnserve` est démarré, redémarrez-le pour qu'il puisse prendre en compte les nouvelles valeurs de configuration.

À présent que tout est mis en place, il ne vous reste plus qu'à créer des utilisateurs ainsi que leur mot de passe. Pour ce faire, utilisez l'utilitaire `saslpaswd2`. Si vous avez utilisé le programme d'installation de CollabNet, cet utilitaire sera dans le répertoire d'installation. Utilisez une commande du style :

```
saslpaswd2 -c -f C:\TortoiseSVN\sasldb -u realm utilisateur
```

L'option `-f` donne le chemin vers la base de données, `realm` doit avoir la même valeur que celle définie dans le fichier `svnserve.conf` de votre référentiel, et `utilisateur` est exactement ce à quoi vous vous attendez. La valeur `realm` ne doit pas contenir d'espace.

Vous pouvez voir la liste des utilisateurs enregistrés dans la base de données en utilisant le programme `sasldblistusers2`.

### 3.6.5.3. Cryptage SASL

Pour activer ou désactiver des niveaux de cryptage, vous pouvez renseigner les deux valeurs suivantes dans le fichier `svnserve.conf` de votre référentiel :

```
[sasldb]  
use-sasl = true  
min-encryption = 128  
max-encryption = 256
```

Les options `min-encryption` et `max-encryption` contrôlent le niveau de cryptage requis par le serveur. Pour désactiver complètement ce cryptage, mettez ces deux valeurs à 0. Pour activer un checksum simple (i.e., protéger et garantir l'intégrité des données non cryptées), mettez les deux options à 1. Si vous voulez que le cryptage soit possible (mais pas nécessaire), mettez le minimum à 0, et le maximum à une valeur de quelques bits. Pour obliger sans condition le cryptage, les deux options à des valeurs supérieures à 1. Dans notre exemple précédent, on oblige les clients à faire un cryptage situé entre 128 bits et 256 bits.

### 3.6.6. Authentification avec svn+ssh

Une autre façon d'authentifier les utilisateurs avec un serveur basé sur `svnserve` est d'utiliser un shell sécurisé (SSH) pour tunneler les requêtes. Cette méthode n'est pas aussi facile à configurer que SASL, mais peut être utile dans certains cas.

Avec cette approche, `svnserve` n'est pas exécuté comme un processus démon, le shell sécurisé démarre `svnserve` pour vous plutôt, l'exécutant comme l'utilisateur SSH authentifié. Pour activer cela, vous avez besoin d'un démon shell sécurisé sur votre serveur.

Une méthode simple pour mettre en place votre serveur est donnée là : [Annexe G, Sécuriser Svnserve grâce à SSH](#). Vous pouvez trouver d'autres informations concernant SSH dans les FAQ en tapant « SSH » comme champ de recherche.

Plus d'informations sur `svnserve` dans [Version Control with Subversion](http://svnbook.red-bean.com) [http://svnbook.red-bean.com].

### 3.6.7. Autorisation basée sur le chemin avec svnserve

En démarrant avec Subversion 1.3, `svnserve` supporte le même arrangement d'autorisation à base de chemin `mod_authz_svn` qui est disponible avec le serveur Apache. Vous devez éditer le fichier `conf/svnserve.conf` dans votre répertoire de référentiel et ajouter une ligne se référant à votre fichier d'autorisation.

```
[general]  
authz-db = authz
```

Ici, `authz` est un fichier que vous créez pour définir les autorisations d'accès. Vous pouvez utiliser un fichier séparé pour chaque référentiel, ou vous pouvez utiliser le même fichier pour plusieurs référentiels. Lisez [Section 3.7.6, « Autorisation basée sur le chemin »](#) pour une description du format du fichier.

## 3.7. Serveur basé sur Apache

### 3.7.1. Introduction

La plus flexible de toutes les installations serveur possibles pour Subversion est basée sur Apache. Bien qu'un peu plus compliquée à mettre en place, elle offre des avantages que d'autres serveurs n'ont pas :

#### WebDAV

Le serveur Subversion basé sur Apache utilise le protocole WebDAV qui est aussi supporté par beaucoup d'autres programmes. Vous pourriez par exemple monter un tel référentiel comme un « Répertoire Web » dans l'explorateur Windows et y accéder ensuite comme n'importe quel autre dossier.

#### Parcourir le référentiel

Vous pouvez diriger votre navigateur à l'URL de votre référentiel et naviguer dans son contenu sans avoir de client Subversion installé. Cela donne accès à vos données à un cercle beaucoup plus large d'utilisateurs.

#### Authentification

Vous pouvez utiliser n'importe quel mécanisme d'identification que supporte Apache, y compris SSPI et LDAP.

#### Sécurité

Puisqu'Apache est très stable et sécurisé, vous obtenez automatiquement la même sécurité pour votre référentiel. Cela inclut le cryptage SSL.

### 3.7.2. Installer Apache

La première chose dont vous avez besoin avant d'installer Apache est d'un ordinateur avec Windows2000, WinXP+SP1, Windows 2003, Vista ou Windows Server 2008.



#### Avertissement

Veuillez noter que Windows XP sans le service pack 1 entrainera des problèmes de données réseau et pourrait donc corrompre votre référentiel !

1. Téléchargez la dernière version du serveur web Apache à partir de <http://httpd.apache.org/download.cgi>. Assurez-vous que vous téléchargez la version 2.2.x - la version 1.3.xx ne marchera pas !

Le programme d'installation d'Apache est disponible en cliquant sur le lien autres fichiers, allez ensuite dans le répertoire `binaries/win32`. Vous pouvez utiliser le fichier `msi apache-2.2.x-win32-x86-openssl-0.9.x.msi` (celui incluant le support OpenSSL).

2. Une fois que vous avez l'installateur Apache2, vous pouvez double-cliquer dessus et il vous guidera dans le processus d'installation. Assurez-vous que vous entrez l'URL du serveur correctement (si vous n'avez pas de nom de DNS pour votre serveur entrez seulement l'adresse IP). Je recommande d'installer Apache *pour Tous les Utilisateurs, sur le Port 80, en tant que Service*. Note : si vous avez déjà IIS ou un autre programme lancé qui écoute sur le port 80 l'installation pourrait échouer. Si cela arrive, allez au répertoire de programmes, `\Apache Group\Apache2\conf` et repérer le fichier `httpd.conf`.

Éditez ce fichier pour que `Listen 80` soit changé pour un port libre, par exemple `Listen 81`. Redémarrez alors l'installation - elle devrait cette fois se terminer sans problème.

3. Maintenant testez si le serveur web Apache fonctionne correctement en allant avec votre navigateur internet à `http://localhost/` - un site web préconfiguré devrait s'afficher.



## Attention

Si vous décidez d'installer Apache en tant que service, soyez averti que par défaut il fonctionnera en tant que compte du système local. Une pratique plus sécurisée pour vous serait de créer un compte séparé pour exécuter Apache.

Assurez-vous que le compte sur le serveur sur lequel Apache fonctionne possède une entrée explicite dans la liste de contrôle d'accès du répertoire du référentiel (clic-droit sur le répertoire | propriétés | sécurité), avec contrôle total. Autrement, les utilisateurs ne seront pas capables de livrer leurs changements.

Même si Apache fonctionne en tant que système local, vous avez toujours besoin d'une telle entrée (qui sera le compte SYSTEM dans ce cas).

Si cette permission n'est pas activé dans la configuration de votre serveur Apache, vos utilisateurs auront des messages d'erreur « Accès refusé », qui s'afficheront dans le journal des erreurs Apache comme erreur 500.

### 3.7.3. Installer Subversion

1. Téléchargez la dernière version des fichiers binaires Win32 de Subversion pour Apache. Soyez sûr de prendre la version correspondant à la version d'Apache que vous avez installé, dans le cas contraire vous aurez un message incompréhensible quand vous redémarrerez. Si votre version d'Apache est la 2.2.x allez à cette adresse : <http://subversion.tigris.org/servlets/ProjectDocumentList?folderID=8100>.
2. Exécutez l'installateur de Subversion et suivez les instructions. Si l'installateur de Subversion a reconnu que vous avez installé Apache, donc vous êtes presque au bout. S'il n'a pas pu trouver un serveur Apache alors vous devez faire quelques étapes complémentaires.

3.

En utilisant l'explorateur Windows, allez au répertoire d'installation de Subversion (d'habitude `c:\program files\Subversion`) et trouvez les fichiers `/httpd/mod_dav_svn.so` et `mod_authz_svn.so`. Copiez ces fichiers au répertoire de modules Apache (d'habitude `c:\program files\apache group\apache2\modules`).

4. Copiez les fichiers `/bin/libdb*.dll` et `/bin/intl3_svn.dll` du répertoire d'installation de Subversion au répertoire bin d'Apache.
5. Éditez le fichier de configuration d'Apache (d'habitude `C:\Program Files\Apache Group\Apache2\conf\httpd.conf`) avec un éditeur de texte comme le Bloc-notes et faites les changements suivants :

Décommentez (supprimez la marque '#') les lignes suivantes :

```
#LoadModule dav_fs_module modules/mod_dav_fs.so
#LoadModule dav_module modules/mod_dav.so
```

Ajoutez les deux lignes suivantes à la fin de la section `LoadModule`.

```
LoadModule dav_svn_module modules/mod_dav_svn.so
```

```
LoadModule authz_svn_module modules/mod_authz_svn.so
```

### 3.7.4. Configuration

Vous avez maintenant mis en place Apache et Subversion, mais Apache ne sait pas encore comment gérer les clients Subversion comme TortoiseSVN. Pour qu'Apache sache quelle URL sera utilisée pour les référentiels Subversion vous devez éditer le fichier de configuration d'Apache (d'habitude placé dans `c:\program files\apache group\apache2\conf\httpd.conf`) avec votre éditeur de texte préféré (par exemple le Bloc-notes) :

1. À la fin du fichier de configuration, ajoutez les lignes suivantes :

```
<Location /svn>
  DAV svn
  SVNListParentPath on
  SVNParentPath D:\SVN
  #SVNIndexXSLT "/svnindex.xsl"
  AuthType Basic
  AuthName "Subversion repositories"
  AuthUserFile passwd
  #AuthzSVNAccessFile svnaccessfile
  Require valid-user
</Location>
```

Elle correspondent à une configuration d'Apache pour prendre en compte des référentiels Subversion situés physiquement dans le répertoire `D:\SVN`. Les référentiels sont disponibles pour le monde extérieur à partir de l'URL : `http://MonServeur/svn/` . L'accès est limité aux utilisateurs/mots de passe listés dans le fichier `passwd`.

2. Pour créer le fichier `passwd`, ouvrez l'invite de commande (la boîte DOS) une nouvelle fois, allez dans le dossier `apache2` (habituellement `c:\program files\apache group\apache2`) et créez le fichier en tapant

```
bin\htpasswd -c passwd <nom de l'utilisateur>
```

Cela créera un fichier avec le nom `passwd` qui est utilisé pour l'authentification. Des utilisateurs supplémentaires peuvent être ajoutés avec

```
bin\htpasswd passwd <nom de l'utilisateur>
```

3. Redémarrer le service Apache une nouvelle fois.
4. Allez avec votre navigateur sur `http://MonServeur/svn/MonNouveauRéférentiel` (où `MonNouveauRéférentiel` est le nom du référentiel Subversion que vous avez créé auparavant). Si tout s'est bien passé, un nom d'utilisateur et un mot de passe vous seront demandés, puis vous pouvez voir le contenu de votre référentiel.

Une courte explication sur ce que vous venez juste de saisir :

Réglage	Explication
<Location /svn>	veut dire que les référentiels Subversion sont disponibles à partir de l'URL <code>http://MonServeur/svn/</code>
DAV svn	indique à Apache quel module sera responsable pour servir cette URL - dans ce cas le module de Subversion.

Réglage	Explication
SVNListParentPath on	Pour la version 1.3 de Subversion et supérieure, cette directive permet de lister tous les référentiels disponibles sous SVNParentPath.
SVNParentPath D:\SVN	indique à Subversion de chercher des référentiel sous D:\SVN
SVNIndexXSLT svnindex.xsl"	"/ Utilisé pour parcourir Internet avec un plus joli navigateur.
AuthType Basic	active l'authentification de base, c'est-à-dire Nom de l'utilisateur/mot de passe
AuthName "Subversion repositories"	est utilisé comme information chaque fois qu'un dialogue d'identification apparaît pour dire à l'utilisateur pour quoi l'authentification est demandée.
AuthUserFile passwd	spécifie quel fichier de mots de passe est utilisé pour l'authentification
AuthzSVNAccessFile	Emplacement du fichier d'accès pour les chemins à l'intérieur d'un référentiel Subversion
Require valid-user	spécifie que seuls les utilisateurs qui ont entré un nom d'utilisateur/mot de passe correct peuvent accéder à l'URL

**Tableau 3.1. Réglages du httpd.conf d'Apache**

Mais ce n'est qu'un exemple. Il y a beaucoup, beaucoup plus de possibilités que vous pouvez faire avec le serveur web Apache.

- Si vous voulez que votre référentiel ait l'accès en lecture pour tout le monde mais l'accès en écriture seulement pour des utilisateurs spécifiques vous pouvez changer la ligne

```
Require valid-user
```

```
en
```

```
<LimitExcept GET PROPFIND OPTIONS REPORT>
Require valid-user
</LimitExcept>
```

- Utiliser un fichier passwd limite et autorise l'accès à tous vos référentiels comme une unité. Si vous voulez plus de contrôle sur les utilisateurs par dossier à l'intérieur d'un référentiel vous pouvez décommenter la ligne

```
#AuthzSVNAccessFile svnaccessfile
```

et créer un fichier d'accès de Subversion. Apache s'assurera que seuls les utilisateurs valides sont capables d'avoir accès votre emplacement /svn et passera alors le nom d'utilisateur au module AuthzSVNAccessFile de Subversion pour qu'il puisse mettre en application un accès plus granulaire basé sur les règles inscrites dans le fichier d'accès de Subversion. Notez que les chemins sont spécifiés comme référentiel:chemin ou simplement chemin. Si vous ne spécifiez pas de référentiel particulier, cette règle d'accès s'appliquera à tous les référentiels sous SVNParentPath. Le format du fichier de stratégie d'autorisation utilisé par mod\_authz\_svn est décrit dans [Section 3.7.6, « Autorisation basée sur le chemin »](#)

- Pour parcourir le référentiel avec un navigateur Internet plus agréable, décommentez la ligne

```
#SVNIndexXSLT "/svnindex.xsl"
```

et mettez les fichiers `svnindex.xsl`, `svnindex.css` et `menucheckout.ico` dans le dossier racine (la plupart du temps `C:/Program Files/Apache Group/Apache2/htdocs`). Le nom de ce répertoire est fixé par la directive `DocumentRoot` dans votre fichier de configuration Apache.

Vous pouvez récupérer ces trois fichiers depuis notre référentiel : <http://tortoissvn.googlecode.com/svn/trunk/contrib/svnindex>. (Section 3, « TortoiseSVN est gratuit ! » explique comment accéder au référentiel des sources de TortoiseSVN).

Le fichier XSL du référentiel TortoiseSVN a une fonctionnalité pratique : Si vous parcourez le référentiel grâce à un navigateur Internet, chaque répertoire aura une icône sur sa droite. Si vous cliquez sur cette icône, la fenêtre de livraison de TortoiseSVN sera ouverte depuis cette URL.

### 3.7.5. Plusieurs référentiels

Si vous avez utilisé la directive `SVNParentPath` alors vous n'avez pas à changer le fichier de configuration d'Apache chaque fois que vous ajoutez un nouveau référentiel Subversion. Créez simplement le nouveau référentiel au même emplacement que le premier référentiel et vous avez fini ! Dans ma société, j'ai un accès direct à ce dossier spécifique sur le serveur via SMB (accès Windows normal aux fichiers). Donc je crée juste un nouveau dossier là, j'exécute la commande de TortoiseSVN TortoiseSVN → Créer un référentiel ici... et un nouveau projet a un foyer...

Si vous utilisez Subversion 1.3 ou supérieur, vous pouvez utiliser la directive `SVNListParentPath` on pour permettre à Apache de produire une liste de tous les projets disponibles si vous allez avec votre navigateur sur le chemin parent plutôt qu'à un référentiel spécifique.

### 3.7.6. Autorisation basée sur le chemin

Le module `mod_authz_svn` permet un contrôle précis des autorisations d'accès basé sur les noms des utilisateurs et les chemins de référentiel. C'est possible avec le serveur Apache et, à partir de Subversion 1.3, c'est aussi disponible avec `svnserve`.

Un fichier d'exemple ressemblerait à ceci :

```
[groups]
admin = john, kate
devteam1 = john, rachel, sally
devteam2 = kate, peter, mark
docs = bob, jane, mike
training = zak
# Accès par défaut à TOUS les référentiels
# Tout le monde peut lire, les admins peuvent écrire, Dan German est exclus.
[/]
* = r
@admin = rw
dangerman =
# Permet aux développeurs un accès total aux référentiels de leurs projets
[proj1:/]
@devteam1 = rw
[proj2:/]
@devteam2 = rw
[bigproj:/]
@devteam1 = rw
@devteam2 = rw
trevor = rw
# Donne aux personnes en charge de la documentation un accès en écriture sur tous l
```

```
[/trunk/doc]
@docs = rw
# Donne aux débutants un accès en écriture uniquement au référentiel d'entraînement
[TrainingRepos:/]
@training = rw
```

Notez que la vérification de chaque chemin peut être une opération coûteuse, particulièrement dans le cas du journal de révision. Le serveur vérifie chaque chemin changé dans chaque révision et le vérifie pour la lisibilité, qui peut être consommatrice de temps sur les révisions qui affectent de grands nombres de fichiers.

L'authentification et l'autorisation sont des processus séparés. Si un utilisateur veut obtenir l'accès à un chemin de référentiel, il doit satisfaire *tant* les pré-requis habituels d'authentification que les pré-requis d'autorisation du fichier d'accès.

### 3.7.7. Authentification avec un domaine Windows

Comme vous l'avez peut-être remarqué vous devez faire une entrée nom d'utilisateur/mot de passe dans le fichier `passwd` pour chaque utilisateur séparément. Et si (pour des raisons de sécurité) vous voulez que vos utilisateurs changent périodiquement leurs mots de passe vous devez faire le changement manuellement.

Mais il y a une solution pour ce problème - du moins si vous avez accès au référentiel de l'intérieur d'un réseau local avec un contrôleur de domaine Windows : `mod_auth_sspi` !

Le module SSPI original a été offert par Syneapps en incluant le code source. Mais son développement a été arrêté. Mais ne désespérez pas, la communauté l'a récupéré et l'a amélioré. Il a un nouveau foyer sur [SourceForge](http://sourceforge.net/projects/mod-auth-sspi/) [http://sourceforge.net/projects/mod-auth-sspi/].

- Téléchargez le module qui correspond à votre version d'Apache, copiez alors le fichier `mod_auth_sspi.so` dans le dossier contenant les modules Apache.
- Éditez le fichier de configuration d'Apache : ajoutez la ligne

```
LoadModule sspi_auth_module modules/mod_auth_sspi.so
```

à la section `LoadModule`. Assurez-vous d'insérer cette ligne *avant* la ligne

```
LoadModule auth_module modules/mod_auth.so
```

- Pour faire que les emplacements de Subversion utilisent ce type d'authentification, vous devez changer la ligne

```
AuthType Basic
```

```
en
```

```
AuthType SSPI
```

vous devez ajouter aussi

```
SSPIAuth On
SSPIAuthoritative On
SSPIDomain <domaincontroller>
SSPIOmitDomain on
```

```
SSPIUsernameCase lower  
SSPIPerRequestAuth on  
SSPIOfferBasic On
```

dans le bloc `<Location /svn>`. Si vous n'avez pas de contrôleur de domaine, laissez le nom du contrôle de domaine comme `<domaincontroller>`.

Notez que si vous vous authentifiez en utilisant SSPI, alors vous n'avez plus besoin de la ligne `AuthUserFile` pour définir un fichier de mot de passe. Apache authentifie votre nom d'utilisateur et votre mot de passe avec votre domaine Windows à la place. Vous devez mettre à jour la liste d'utilisateurs dans votre `svnaccessfile` pour faire aussi référence au `DOMAINE\nom d'utilisateur`.



### Important

L'authentification SSPI n'est disponible que pour les connexions sécurisées de type SSL (https). Si vous vous connectez à votre serveur en utilisant une connexion http simple, cela ne fonctionnera pas.

Pour activer SSL sur votre serveur, voir le chapitre : [Section 3.7.9, « Sécuriser le serveur avec SSL »](#)



### Astuce

Les fichiers `AuthzSVNAccessFile` de Subversion sont sensibles à la casse en ce qui concerne les noms d'utilisateur ("JUtilisateur" diffère "jutilisateur").

Dans le monde de Microsoft, les domaines Windows et les mots de passe ne sont pas sensibles à la casse. Cependant, certains administrateurs réseau aiment créer des comptes utilisateur en CamelCase (par exemple "JUtilisateur").

Cette différence peut vous mordre quand vous utilisez l'authentification SSPI lorsque le domaine Windows et les noms d'utilisateur sont passés à Subversion comme lorsque l'utilisateur les tape à l'invite. Internet Explorer passe souvent le nom de l'utilisateur à Apache en utilisant automatiquement la casse avec laquelle le compte a été créé.

Le résultat final est que vous pouvez avoir besoin d'au moins deux entrées dans votre `AuthzSVNAccessFile` pour chaque utilisateur - une entrée minuscule et une entrée dans la même casse que celle qu'Internet Explorer passe à Apache. Vous devrez aussi apprendre à vos utilisateurs à aussi taper leurs accreditations en utilisant les minuscules quand ils accèdent aux référentiels via TortoiseSVN.

Les journaux d'Erreurs et d'Accès d'Apache sont vos meilleurs amis dans le déchiffrement de problèmes comme ceux-ci puisqu'ils vous aideront à déterminer la chaîne du nom d'utilisateur passée par le module `AuthzSVNAccessFile` de Subversion. Vous pouvez avoir besoin d'expérimenter avec le format exact de la chaîne de l'utilisateur dans le `svnaccessfile` (par exemple `DOMAINE\utilisateur` contre `DOMAINE//utilisateur`) pour que tout fonctionne.

## 3.7.8. Plusieurs sources d'authentification

Il est aussi possible d'avoir plus d'une source d'authentification pour votre référentiel Subversion. Pour ce faire, vous devez rendre chaque type d'authentification non définitive, pour qu'Apache vérifie plusieurs sources pour la correspondance nom utilisateur/mot de passe.

Un scénario fréquent consiste à utiliser l'authentification par le domaine Windows et par fichier `passwd`, pour que vous puissiez fournir un accès SVN aux utilisateurs qui ne possèdent pas de login dans le domaine Windows.

- Pour activer l'authentification via le domaine Windows et par un fichier d'authentification `passwd`, ajoutez les lignes suivantes dans le bloc `<Location>` de votre fichier de configuration Apache :

```
AuthBasicAuthoritative Off
SSPIAuthoritative Off
```

Voici un exemple de configuration complète d'Apache pour une authentification combinant le domaine Windows et le fichier d'authentification `passwd` :

```
<Location /svn>
  DAV svn
  SVNListParentPath on
  SVNParentPath D:\SVN

  AuthName "Subversion repositories"
  AuthzSVNAccessFile svnaccessfile.txt

# NT Domain Logins.
  AuthType SSPI
  SSPIAuth On
  SSPIAuthoritative Off
  SSPIDomain <domaincontroller>
  SSPIOfferBasic On

# Htpasswd Logins.
  AuthType Basic
  AuthBasicAuthoritative Off
  AuthUserFile passwd

  Require valid-user
</Location>
```

### 3.7.9. Sécuriser le serveur avec SSL

Même si Apache 2.2.x permet le support OpenSSL, ce n'est pas activé par défaut. Vous devez l'activer manuellement.

1. Dans le fichier de configuration apache, décommentez les lignes:

```
#LoadModule ssl_module modules/mod_ssl.so

et en bas
```

```
#Include conf/extra/httpd-ssl.conf
```

changez alors la ligne (sur une seule ligne)

```
SSLMutex "file:C:/Program Files/Apache Software Foundation/\
Apache2.2/logs/ssl_mutex"
```

```
en
```

```
SSLMutex default
```

2. Ensuite vous devez créer un certificat SSL. Pour faire cela ouvrez une invite de commande (la boîte DOS) et allez au dossier d'Apache (par exemple `C:\program files\apache group\apache2`) et tapez la commande suivante :

```
bin\openssl req -config conf\openssl.cnf -new -out mon-serveur.csr
```

On vous demandera une phrase de mot de passe. N'utilisez pas s'il vous plaît de mots simples, mais des phrases entières, par exemple une partie d'une poésie. Plus la phrase est longue, mieux c'est. Vous devez aussi entrer l'URL de votre serveur. Toutes les autres questions sont facultatives mais nous vous recommandons de remplir celles-ci aussi.

Normalement le fichier `privkey.pem` est créé automatiquement, mais si ce n'est pas le cas vous devez exécuter cette commande pour le générer:

```
bin\openssl genrsa -out conf\privkey.pem 2048
```

Next type the commands

```
bin\openssl rsa -in conf\privkey.pem -out conf\server.key
```

and (on one line)

```
bin\openssl req -new -key conf\server.key -out conf\server.csr \  
-config conf\openssl.cnf
```

and then (on one line)

```
bin\openssl x509 -in conf\server.csr -out conf\server.crt  
-req -signkey conf\server.key -days 4000
```

This will create a certificate which will expire in 4000 days. And finally enter (on one line):

```
bin\openssl x509 -in conf\server.cert -out conf\server.der.crt  
-outform DER
```

These commands created some files in the Apache conf folder (`server.der.crt`, `server.csr`, `server.key`, `.rnd`, `privkey.pem`, `server.cert`).

3. Redémarrez le service Apache.
4. Allez avec votre navigateur à `https://servername/svn/project ...`



## SSL et Internet Explorer

Si vous sécurisez votre serveur avec SSL et utilisez l'authentification avec un domaine Windows vous vous apercevrez que la navigation dans le référentiel avec Internet Explorer ne marche plus. Ne vous inquiétez pas - c'est seulement qu'Internet Explorer n'est pas capable d'authentifier. Les autres navigateurs n'ont pas ce problème et TortoiseSVN et les autres clients Subversion sont toujours capable d'authentifier.

Si vous voulez toujours utiliser IE pour parcourir le référentiel vous pouvez soit :

- définissez une directive `<Location /path>` dans le fichier de configuration d'Apache et ajoutez `SSPIBasicPreferred On`. Cela permettra à IE d'authentifier à nouveau,

mais les autres navigateurs et Subversion ne seront pas capables d'authentifier pour cet emplacement.

- Offrir de naviguer avec une authentification non cryptée (sans SSL) aussi. Étrangement IE n'a pas de problèmes avec l'authentification si la connexion n'est pas sécurisée avec SSL.
- Dans l'installation "standard" de ssl, il y a souvent la déclaration suivante dans l'hôte SSL virtuel d'Apache :

```
SetEnvIf User-Agent ".*MSIE.*" \  
          nokeepalive ssl-unclean-shutdown \  
          downgrade-1.0 force-response-1.0
```

Il y a (avait ?) de bonnes raisons pour cette configuration, regardez [http://www.modssl.org/docs/2.8/ssl\\_faq.html#ToC49](http://www.modssl.org/docs/2.8/ssl_faq.html#ToC49) Mais si vous voulez l'authentification NTLM vous devez utiliser keepalive. Si vous décommentez entièrement SetEnvIf vous devriez être capables d'authentifier IE avec l'authentification windows avec SSL et Apache sur Win32 avec mod\_auth\_sspi inclus.



## Forcer l'accès SSL

Une fois que vous avez mis en place SSL pour rendre votre référentiel plus sécurisé, vous pourriez vouloir désactiver l'accès normal non-SSL (http) et permettre seulement l'accès https. Pour ce faire, vous devez ajouter une autre directive au bloc <Location> de Subversion : SSLRequireSSL.

Voici un exemple de bloc <Location> :

```
<Location /svn>  
  DAV svn  
  SVNParentPath D:\SVN  
  SSLRequireSSL  
  AuthType Basic  
  AuthName "Subversion repositories"  
  AuthUserFile passwd  
  #AuthzSVNAccessFile svnaccessfile  
  Require valid-user  
</Location>
```

### 3.7.10. Utiliser des certificats avec des hôtes SSL virtuels

Envoyé à la liste de diffusion TortoiseSVN par Nigel Green. Merci !

Avec certaines configurations de serveur, vous pouvez avoir à mettre en place un seul serveur avec 2 hôtes SSL virtuels: Le premier pour les accès internet publics, ne nécessitant aucun certificat côté client. Et un second, sécurisé avec certificat côté client, où tourne le serveur Subversion.

Dans le fichier de configuration de Apache, ajouter une directive `SSLVerifyClient Optional` à la section *per-server* (i.e. en dehors de balises `VirtualHost` et `Directory`) force Apache à demander un certificat au client au moment d'établir la connexion SSL. A cause d'un bug dans le module `mod_ssl` il est crucial de demander le certificat à ce moment car sinon il ne fonctionnera plus en cas de tentative de re-connexion SSL.

La solution est d'ajouter la directive suivante dans le répertoire du virtual host que vous voulez réserver à Subversion:

```
SSLRequire %{SSL_CLIENT_VERIFY} eq "SUCCESS"
```

Cette directive autorise l'accès au répertoire seulement si le certificat du client a été reçu et validé.

To summarise, the relevant lines of the Apache configuration are:

```
SSLVerifyClient Optional

### Virtual host configuration for the PUBLIC host
### (not requiring a certificate)

<VirtualHost 127.0.0.1:443>
  <Directory "pathtopublicfileroot">
    </Directory>
</VirtualHost>

### Virtual host configuration for SUBVERSION
### (requiring a client certificate)
<VirtualHost 127.0.0.1:443>
  <Directory "subversion host root path">
    SSLRequire %{SSL_CLIENT_VERIFY} eq "SUCCESS"
  </Directory>

  <Location /svn>
    DAV svn
    SVNParentPath /pathtorepository
  </Location>
</VirtualHost>
```

---

# Chapitre 4. Guide d'utilisation quotidienne

Ce document décrit l'utilisation quotidienne du client TortoiseSVN. Ce n'est *pas* une introduction aux systèmes de contrôle de version, *ni* une introduction à Subversion (SVN). C'est plutôt un endroit où vous pouvez regarder quand vous savez approximativement ce que vous voulez faire, mais vous ne vous rappelez pas tout à fait comment le faire.

If you need an introduction to version control with Subversion, then we recommend you read the fantastic book: *Version Control with Subversion* [<http://svnbook.red-bean.com/>]. Si vous avez besoin d'une introduction au contrôle de version avec Subversion, alors nous vous recommandons la lecture du livre fantastique : *Version Control with Subversion* [<http://svnbook.red-bean.com/>].

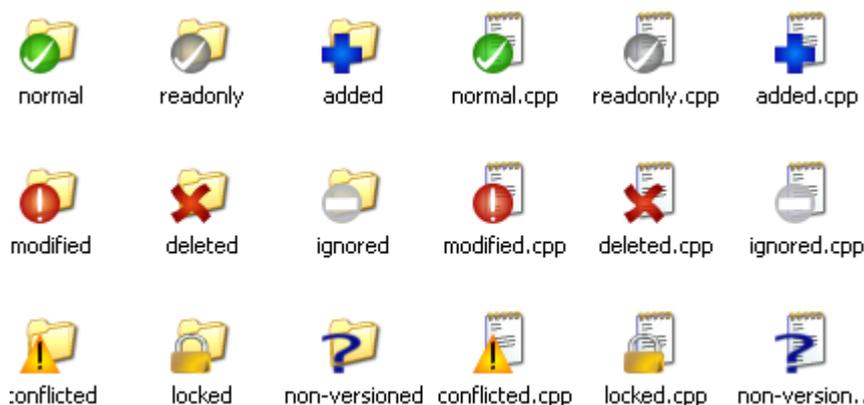
Ce document est aussi un travail en cours, de même que TortoiseSVN et Subversion. Si vous trouvez des erreurs, veuillez les annoncer sur la mailing list pour que nous puissions mettre à jour la documentation. Certaines copies d'écran dans le Guide d'Utilisation Quotidienne (GUQ) pourraient ne pas refléter l'état actuel du logiciel. Veuillez nous pardonner. Nous travaillons sur TortoiseSVN pendant notre temps libre.

Pour se servir au mieux du guide d'utilisation quotidienne :

- Vous devez avoir déjà installé TortoiseSVN.
- Vous devriez être familier avec les systèmes de contrôle de version.
- Vous devez connaître les bases de Subversion.
- Vous devriez avoir mis en place un serveur et/ou avoir accès à un référentiel Subversion.

## 4.1. Pour commencer

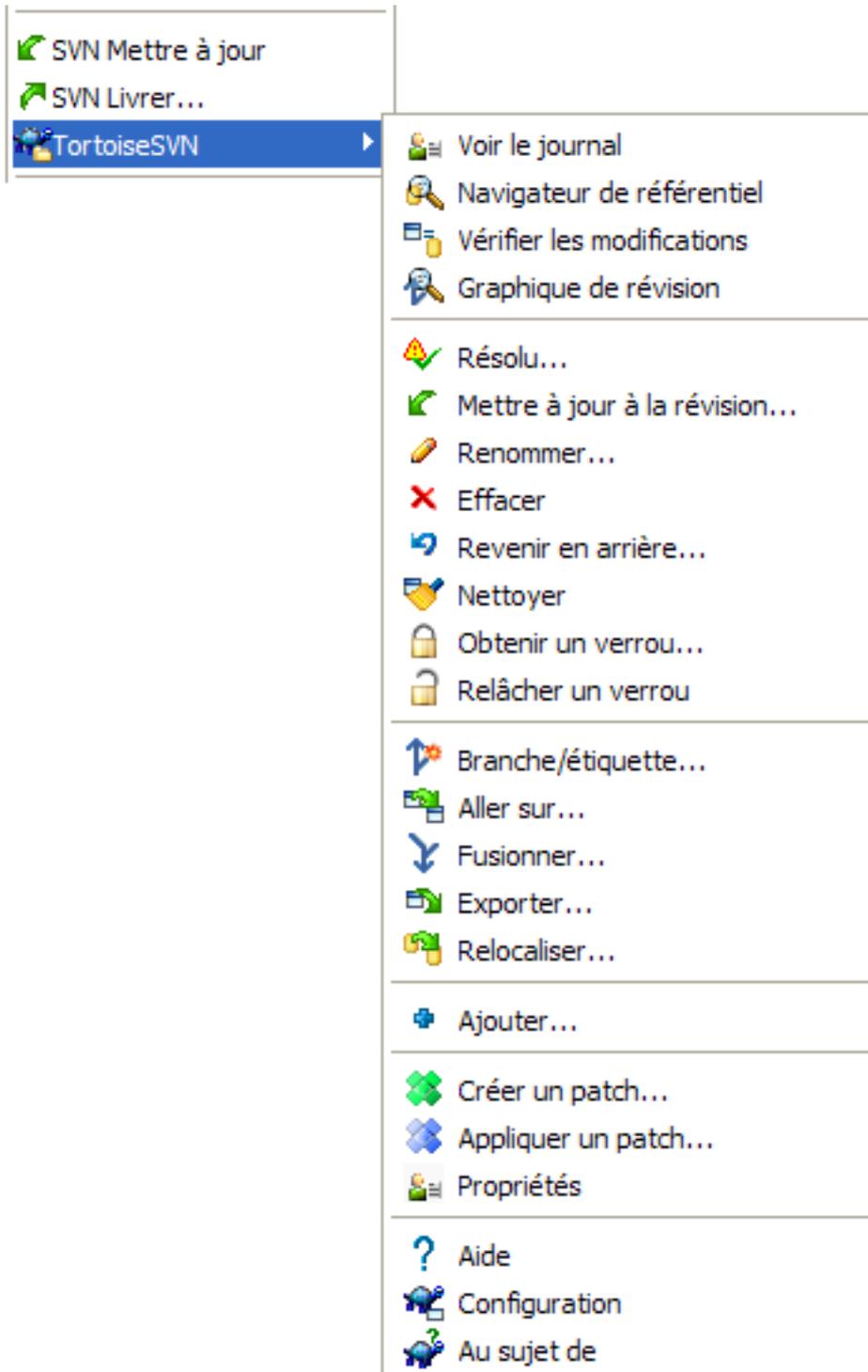
### 4.1.1. Recouvrement d'icônes



**Figure 4.1. L'Explorateur montrant le recouvrement d'icônes**

Une des fonctionnalités les plus visibles de TortoiseSVN sont les recouvrements d'icônes qui apparaissent sur les fichiers de votre copie de travail. Celles-ci vous montrent en un clin d'oeil quels fichiers ont été modifiés. Référez-vous à [Section 4.7.1, « Recouvrement d'icônes »](#) pour découvrir que représentent les différentes icônes.

### 4.1.2. Menus contextuels



**Figure 4.2. Menu contextuel pour un répertoire sous contrôle de version**

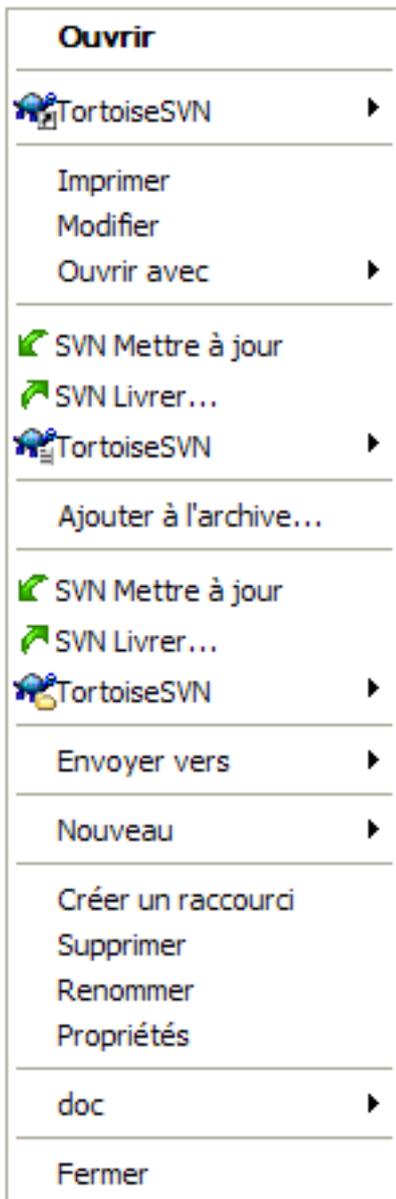
Toutes les commandes de TortoiseSVN sont invoquées à partir du menu contextuel de l'explorateur Windows. La plupart sont directement visibles quand vous faites un clic droit sur un fichier ou sur un dossier. Les commandes disponibles dépendent si le fichier ou le dossier ou son dossier parent sont sous contrôle de version ou non. Vous pouvez aussi voir le menu TortoiseSVN comme une partie du menu fichier de l'Explorateur.



## Astuce

Quelques commandes qui sont très rarement utilisées ne sont accessibles que dans l'extension du menu contextuel. Pour afficher cette extension, appuyer sur **Shift** au moment où vous faites le click-droit.

Dans certains cas, vous pouvez voir plusieurs entrées TortoiseSVN. Ce n'est pas un bug !

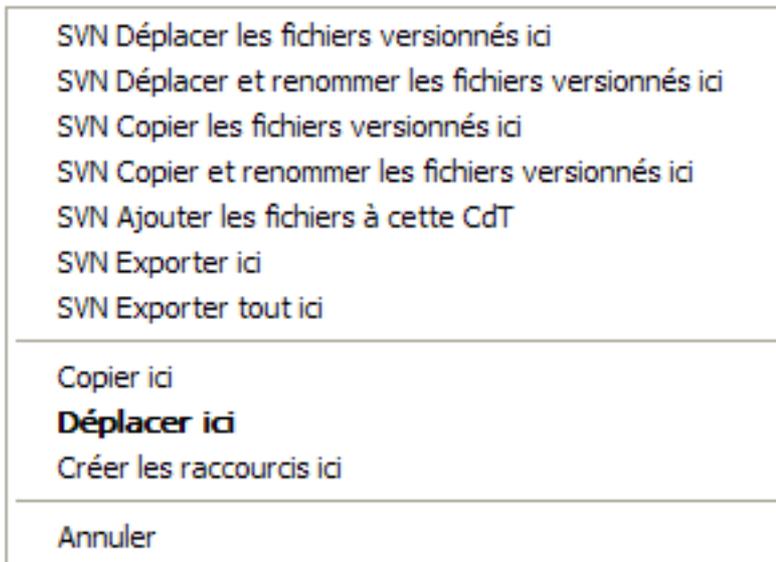


**Figure 4.3. Menu fichier de l'Explorateur pour un raccourci dans un répertoire non versionné**

Cet exemple est pour un raccourci non versionné dans un dossier versionné et dans le menu de fichier de l'Explorateur il y a *trois* entrées pour TortoiseSVN. L'une est pour le dossier, une autre pour le raccourci lui-même et la troisième pour l'objet sur lequel pointe le raccourci. Pour vous aider à les distinguer entre elles, les icônes ont un indicateur dans le coin inférieur droit pour montrer si l'entrée de menu est pour un fichier, un dossier, un raccourci ou pour des plusieurs éléments sélectionnés.

Si vous êtes un utilisateur de Windows 2000 vous verrez le menu contextuel sans icône. Nous savons que cela fonctionnait dans les versions précédentes, mais Microsoft has changed the way its icon handlers work for Vista, requiring us to use a different display method which unfortunately does not work on Windows 2000.

#### 4.1.3. Glisser-déposer



**Figure 4.4. Menu pour un répertoire sous contrôle de version lors d'un glisser-déplacer avec le bouton droit**

D'autres commandes sont disponibles par glisser-déposer, quand vous glissez-déposez avec le bouton droit des fichiers ou des dossiers vers un nouvel emplacement à l'intérieur des copies de travail ou quand vous glissez-déposez avec le bouton droit un fichier non versionné ou un dossier dans un répertoire qui est sous contrôle de version.

#### 4.1.4. Raccourcis communs

Quelques opérations communes ont des raccourcis Windows bien connus, mais qui n'apparaissent ni sur les boutons ni dans les menus. Si vous ne savez pas comment faire quelque chose d'évident, comme le rafraîchissement d'une vue, regardez ici.

F1

L'aide, bien sûr.

F5

Rafraîchir la vue courante. C'est peut-être la commande d'une touche la plus utile. Par exemple... Dans l'Explorateur cela rafraîchira le recouvrement des icônes sur votre copie de travail. Dans la boîte de dialogue de livraison il parcourra à nouveau la copie de travail pour voir ce qui pourrait être livré. Dans la boîte de dialogue du journal de révision, il entrera à nouveau en contact avec le référentiel pour vérifier s'il y a des changements plus récents.

Ctrl-A

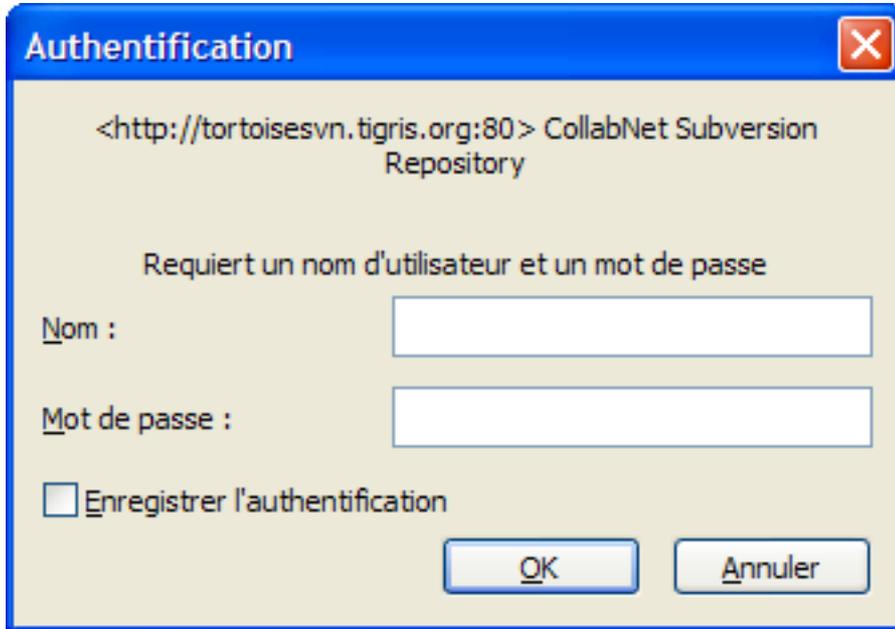
Sélectionner tout. Cela peut être utilisé si vous obtenez un message d'erreur et que vous voulez copier-coller dans un email. Utilisez Ctrl-A pour choisir le message d'erreur et ensuite...

Ctrl-C

... Copier le texte choisi.

#### 4.1.5. Authentification

Si le référentiel auquel vous essayez d'avoir accès est protégé par un mot de passe, une boîte de dialogue d'identification s'affichera.



**Figure 4.5. Boîte de dialogue d'authentification**

Entrez votre nom d'utilisateur et votre mot de passe. La case à cocher fera que TortoiseSVN stockera les accreditations dans le répertoire de Subversion par défaut : %APPDATA%\Subversion\auth dans trois sous-répertoires :

- `svn.simple` contains credentials for basic authentication (username/password).
- `svn.ssl.server` contient des certificats serveur SSL.
- `svn.username` contient les accreditations pour l'authentification avec le nom d'utilisateur uniquement (aucun mot de passe nécessaire).

Si vous voulez effacer le cache d'authentification pour tous les serveurs, vous pouvez le faire à partir de la page **Données Sauvegardées** de la boîte de dialogue de configuration de TortoiseSVN. Ce bouton effacera toutes les données d'authentification gardées en cache dans les répertoire `auth` de Subversion, aussi bien que les authentifications stockées par des versions précédemment installées de TortoiseSVN. Référez-vous à [Section 4.30.6, « Configuration des données sauvegardées »](#).

Certains préfèrent que leurs données d'identification soient supprimées à l'ouverture ou à la fermeture de Windows. Le moyen de le faire est d'utiliser un script de fermeture pour supprimer le répertoire %APPDATA%\Subversion\auth, i.e.

```
@echo off
rmdir /s /q "%APPDATA%\Subversion\auth"
```

Vous pouvez trouver le mode d'emploi pour installer ce type de script à cette adresse [windows-help-central.com](http://www.windows-help-central.com/windows-shutdown-script.html) [http://www.windows-help-central.com/windows-shutdown-script.html].

Pour plus d'informations sur la façon de mettre en place l'authentification et le contrôle d'accès pour votre serveur, référez-vous à [Section 3.5, « Accéder au référentiel »](#)

### 4.1.6. Maximiser les fenêtres

Beaucoup de fenêtres de TortoiseSVN ont nombre d'informations à afficher, mais il est souvent plus utile de maximiser leur largeur ou leur hauteur plutôt que de les afficher en plein écran. Pour ce faire, il y a des raccourcis dans le bouton **Maximiser**. Utilisez le bouton du milieu de la souris pour agrandir la hauteur, et le bouton droit de la souris pour élargir.

## 4.2. Importer des données dans un référentiel

### 4.2.1. Importer

If you are importing into an existing repository which already contains some projects, then the repository structure will already have been decided. If are importing data into a new repository then it is worth taking the time to think about how it will be organised. Read [Section 3.1.5, « Disposition du référentiel »](#) for further advice.

Cette section décrit la commande Importer, qui a été faite pour importer une arborescence de répertoires dans le référentiel en une action. Bien que cette commande fonctionne, elle a quelques défauts:

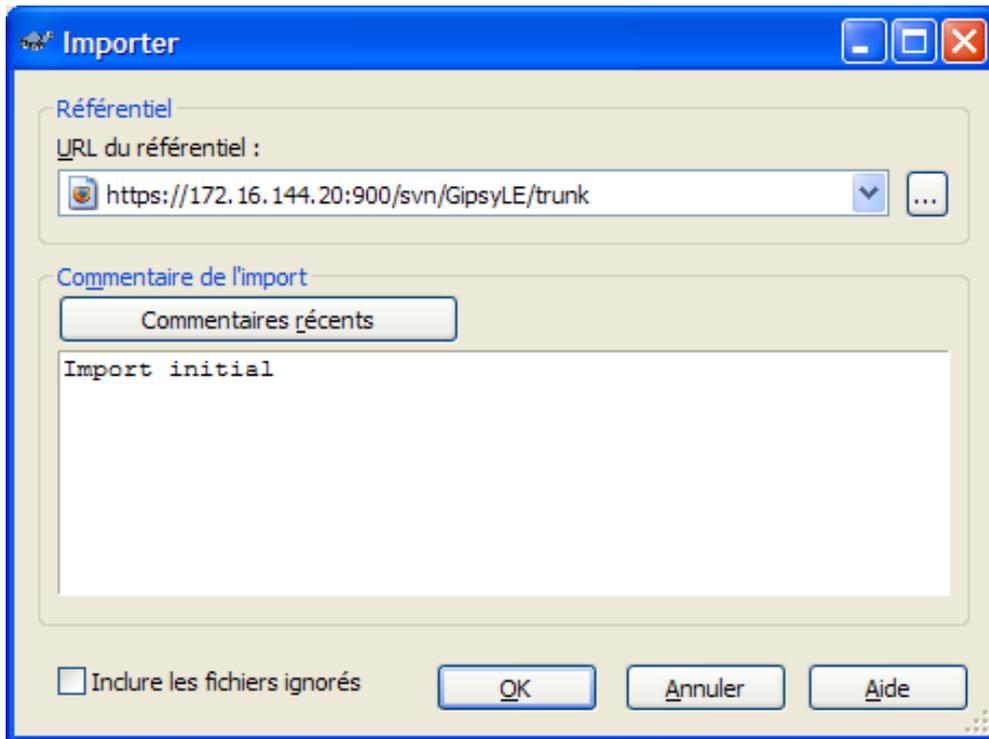
- Il n'y a aucun moyen de sélectionner des éléments à inclure, à part en utilisant le filtre global des fichiers à ignorer.
- Le répertoire importé ne devient pas une copie de travail. Vous devez faire une extraction pour en avoir une directement du serveur.
- Il est aisé de se tromper de répertoire à importer dans le référentiel.

Pour ces raisons nous recommandons d'utiliser la méthode en deux étapes décrite dans : [Section 4.2.2, « Importer en place »](#) au lieu de cette commande . Mais puisque vous êtes là, c'est le fonctionnement de la commande Importer...

Avant que vous n'importiez votre projet dans un référentiel, vous devriez :

1. Supprimez tous les fichiers qui ne sont pas nécessaires pour générer le projet (fichiers temporaires, fichiers produits par un compilateur par exemple \*.obj, binaires compilés...)
2. Organisez les fichiers en dossiers et sous-dossiers. Bien qu'il soit possible de renommer/déplacer les fichiers plus tard, il est fortement recommandé de fixer la structure de votre projet juste avant l'importation !

Choisissez maintenant le dossier de niveau supérieur de votre structure de répertoire de projet dans l'explorateur Windows et faite un clic droit pour ouvrir le menu contextuel. Choisissez la commande TortoiseSVN → Importer... qui fait s'afficher une boîte de dialogue :



**Figure 4.6. La boîte de dialogue Importer**

Dans la boîte de dialogue, vous devez entrer l'URL du référentiel dans lequel vous souhaitez importer le projet. Il est important de comprendre que le répertoire local que vous importez n'apparaîtra pas dans le référentiel, mais juste son contenu. Par exemple, si vous avez la structure suivante :

```
C:\Projects\Widget\source
C:\Projects\Widget\doc
C:\Projects\Widget\images
```

et que vous importez C:\Projects\Widget dans `http://monnomdedomaine.com/svn/trunk` vous serez alors surpris de voir que vos sous répertoires iront directement dans trunk au lieu d'être dans un répertoire Widget. Vous devez spécifier le sous répertoire comme faisant partie de l'URL, `http://monnomdedomaine.com/svn/trunk/Widget-X`. Notez que la commande importer créera automatiquement les sous répertoires dans le référentiel s'ils n'existent pas.

Le message d'importation est utilisé comme un commentaire.

Par défaut, les fichiers et les dossiers qui correspondent au filtre d'exclusion ne sont *pas* importés. Pour ignorer ce comportement, vous pouvez utiliser la case à cocher **Inclure les fichiers ignorés**. Référez-vous à [Section 4.30.1, « Configuration générale »](#) pour plus d'informations sur la configuration d'un filtre d'exclusion.

Dès que vous appuyez sur OK TortoiseSVN importe dans le référentiel l'arborescence complète des répertoires, fichiers compris. Le projet est maintenant sous contrôle de version dans le référentiel. Veuillez noter que le dossier que vous avez importé n'est *PAS* sous contrôle de version ! Pour obtenir une *copie de travail* sous contrôle de version, vous devez faire une extraction de la version que vous venez d'importer. Ou vous renseigner sur la manière d'importer un répertoire déjà en place.

## 4.2.2. Importer en place

Si vous disposez déjà d'un référentiel, et que vous souhaitez y ajouter une nouvelle arborescence, suivez les étapes suivantes :

1. Utiliser l'explorateur de référentiel pour créer un nouveau projet directement dans le référentiel.
2. Extrayez le nouveau répertoire parent du répertoire que vous voulez importer. Un message va s'afficher, indiquant que le répertoire n'est pas vide. Vous disposez maintenant d'un répertoire parent versionné, dont le contenu n'est pas sous contrôle de version.
3. Sélectionnez TortoiseSVN → Ajouter... sur le répertoire sous contrôle de version pour ajouter un ou des éléments. Vous pouvez ajouter ou supprimer des fichiers, activer les propriétés `svn:ignore` sur les répertoires et faire toutes les modifications dont vous avez besoin.
4. Livrez le répertoire de plus haut niveau, et vous aurez une nouvelle arborescence versionnée, et une copie de travail locale, créée depuis votre répertoire existant.

### 4.2.3. Fichiers spéciaux

Parfois vous avez besoin d'avoir un fichier sous contrôle de version qui contient des données spécifiques aux utilisateurs. Cela signifie que vous avez un fichier que chaque développeur/utilisateur doit modifier pour convenir à son paramétrage local. Mais versionner un tel fichier est difficile car chaque utilisateur livrerait son propre fichier à chaque fois.

Dans de tels cas nous suggérons d'utiliser des fichiers *templates*. Vous créez un fichier qui contient toutes les données dont vos développeurs auront besoin, ajoutez ce fichier au contrôle de version et laissez les développeurs extraire ce fichier. Alors, chaque développeur doit *faire une copie* de ce fichier et la renommer. Après cela, modifier la copie n'est plus un problème.

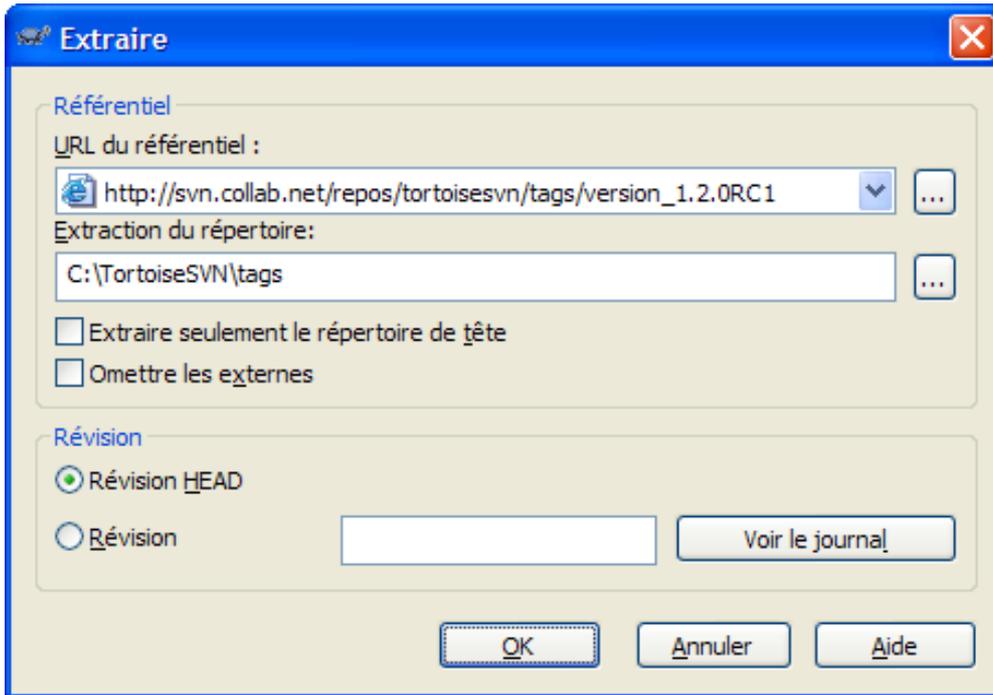
Comme exemple, vous pouvez regarder le script de construction de TortoiseSVN. Il appelle un fichier nommé `TortoiseVars.bat` qui n'existe pas dans le référentiel, seul le fichier `TortoiseVars.tmpl` y est. Ce dernier sert de modèle afin que chaque développeur crée son propre `TortoiseVars.bat`. À l'intérieur de ce fichier, nous avons ajouté des commentaires pour que les utilisateurs voient quelles lignes ils doivent éditer et changer selon leur paramétrage local pour qu'il fonctionne.

Afin de ne pas déranger les utilisateurs, nous avons aussi ajouté le fichier `TortoiseVars.bat` à la liste des ignorés de son dossier parent, c'est-à-dire que nous avons mis la propriété Subversion `svn:ignore` pour inclure ce nom de fichier. De cette façon il n'apparaîtra pas comme non versionné à chaque livraison.

## 4.3. Extraire une copie de travail

Pour obtenir une copie de travail vous devez faire une *extraction* à partir d'un référentiel.

Choisissez un répertoire dans l'explorateur Windows où vous voulez placer votre copie de travail. Faites un clic droit pour afficher le menu contextuel et choisissez la commande TortoiseSVN → Extraire..., qui affiche la boîte de dialogue suivante :



**Figure 4.7. La boîte de dialogue Extraire**

Si vous entrez un nom de dossier qui n'existe pas, alors il sera créé.

### 4.3.1. Profondeur d'extraction

Vous pouvez choisir la *profondeur* que vous voulez extraire, ce qui vous permet de spécifier la profondeur de récursion dans les sous répertoires. Si vous voulez juste une partie d'une grosse arborescence, vous pouvez n'extraire que le répertoire de plus haut niveau, et mettre à jour les sous répertoires petit à petit.

Totalement récursive

Extraire l'arborescence complète, incluant récursivement tous les sous répertoires.

Descendants directs, y compris les répertoires

Extrait le répertoire spécifié, tous les fichiers et sous répertoires compris, mais ne remplit pas les sous répertoires.

Juste les fichiers

Extrait le répertoire spécifié, y compris les fichiers mais n'extrait aucune sous répertoire.

Uniquement cet élément.

Extraire juste le répertoire. Ne pas le remplir avec les répertoires et fichiers enfants.

Copie de travail

Se souvenir de la profondeur dans la version de travail. Cette option n'est pas utilisée dans la fenêtre d'extraction, mais c'est la valeur qui sera utilisée par défaut pour toutes les autres fenêtres ayant cette option.

Exclure

Utilisé pour réduire la profondeur de la copie de travail déjà rempli. Cette option n'est disponible que dans la fenêtre Mettre à jour à la révision.

Si vous n'extrayez qu'une portion de l'arborescence (en décochant la case *récursivement*), vous pouvez lister d'autres sous répertoires via l'explorateur de référentiel ([Section 4.24, « l'explorateur de référentiel »](#)) ou la fenêtre de vérification des modifications ([Section 4.7.3, « Statut local et distant »](#)).

In the repository browser, Right click on the checked out folder, then use TortoiseSVN → Repo-Browser to bring up the repository browser. Find the sub-folder you would like to add to your working

copy, then use **Context menu** → **Update item to revision...** That menu will only be visible if the selected item does not exist yet in your working copy, but the parent item does exist.

Dans la fenêtre de vérification des modifications, cliquez d'abord sur le bouton **Vérifier le référentiel**. La fenêtre affichera tous les fichiers et répertoires du référentiel mais que vous n'avez pas exporté comme étant ajouté à distance. Faites un Click droit sur le(s) répertoire(s) que vous voudriez ajouter à votre CdT, puis utilisez **Menu contextuel** → **Mise à jour**.

Cette fonctionnalité est très utile lorsque vous ne voulez récupérer qu'une partie d'une grosse arborescence, tout en gardant le côté pratique d'une mise à jour d'un seul élément. Supposez que vous avez une grosse arborescence avec des sous répertoires nommés `Projet01` à `Projet99`, et que vous ne vouliez récupérer que `Projet03`, `Projet25` et `Projet76/SousProj`. Suivez ces étapes :

1. Extrait le répertoire parent avec la profondeur « Juste cet élément » Vous avez maintenant un répertoire de haut niveau vide.
2. Sélectionnez le nouveau répertoire et utilisez la commande **TortoiseSVN** → **Explorateur de référentiel** pour voir le contenu du référentiel.
3. Faites un click droit sur `Project03` et **Menu contextuel** → **Mettre à jour à la révision...** Laissez les paramètres par défaut et cliquez sur **OK**. Ce répertoire est a présent complet.

Répéter le même processus pour `Project25`..

4. Allez dans `Project76/SubProj` et faites de même. Cette fois ci, notez que le répertoire `Project76` est vide excepté `SubProj`, lequel est complètement rempli. Subversion a créé pour vous les répertoires intermédiaires sans les remplir.



### Modifier la profondeur de la copie de travail

Une fois que vous avez extrait une copie de travail à une certaine profondeur vous pouvez modifier cette profondeur à postériori en utilisant **Menu contextuel** → **Mettre à jour l'élément à la révision...**



### Utilisant un serveur ancien

Les serveurs de version inférieure à 1.5 ne supportent pas la notion de profondeur de la copie de travail, ils n'interprètent donc pas efficacement les requêtes l'utilisant. Les commande fonctionneront, mais un vieux serveur retournera toutes les données, laissant le client filtrer ce qui n'est pas requis, ce qui augmentera le trafic sur le réseau. Si possible, vous devriez mettre à jour votre serveur.

Si le projet contient des références à des projets externes que vous ne voulez *pas* extraire en même temps, utilisez la case à cocher **Omettre les références externes** .



### Important

If **Omit externals** is checked, or if you wish to increase the depth value, you will have to perform updates to your working copy using **TortoiseSVN** → **Update to Revision...** instead of **TortoiseSVN** → **Update**. The standard update will include all externals and keep the existing depth.

Il est recommandé de n'extraire que la partie `trunk` de l'arborescence de répertoire, ou un de ses sous répertoires. Si vous spécifiez le chemin parent de l'arborescence de répertoire dans l'URL alors vous

pourriez vous retrouver avec un disque dur rempli puisque vous obtiendrez une copie de l'arborescence du référentiel en entier incluant chaque branche et chaque étiquette de votre projet !



## Exporter

Parfois vous pouvez avoir envie de créer une copie locale sans aucun de ces répertoires `.svn`, pour créer un paquet zippé de vos sources par exemple. Lisez [Section 4.26](#), « [Exporter une copie de travail Subversion](#) » pour savoir comment le faire.

## 4.4. Livrer vos changements au référentiel

Envoyer les changements que vous avez faits dans votre copie de travail est connue comme *livrer* les changements. Mais avant de livrer vous devez vous assurer que votre copie de travail est à jour. Vous pouvez soit utiliser directement TortoiseSVN → Mettre à jour. Ou vous pouvez utiliser TortoiseSVN → Vérifier les modifications d'abord, pour voir quels fichiers ont été changés localement ou sur le serveur.

### 4.4.1. La boîte de dialogue Livrer

Si votre copie de travail est à jour et s'il n'y a aucun conflit, vous êtes prêts à livrer vos changements. Choisissez n'importe quel fichier et/ou dossier que vous voulez livrer, puis TortoiseSVN → Livrer....

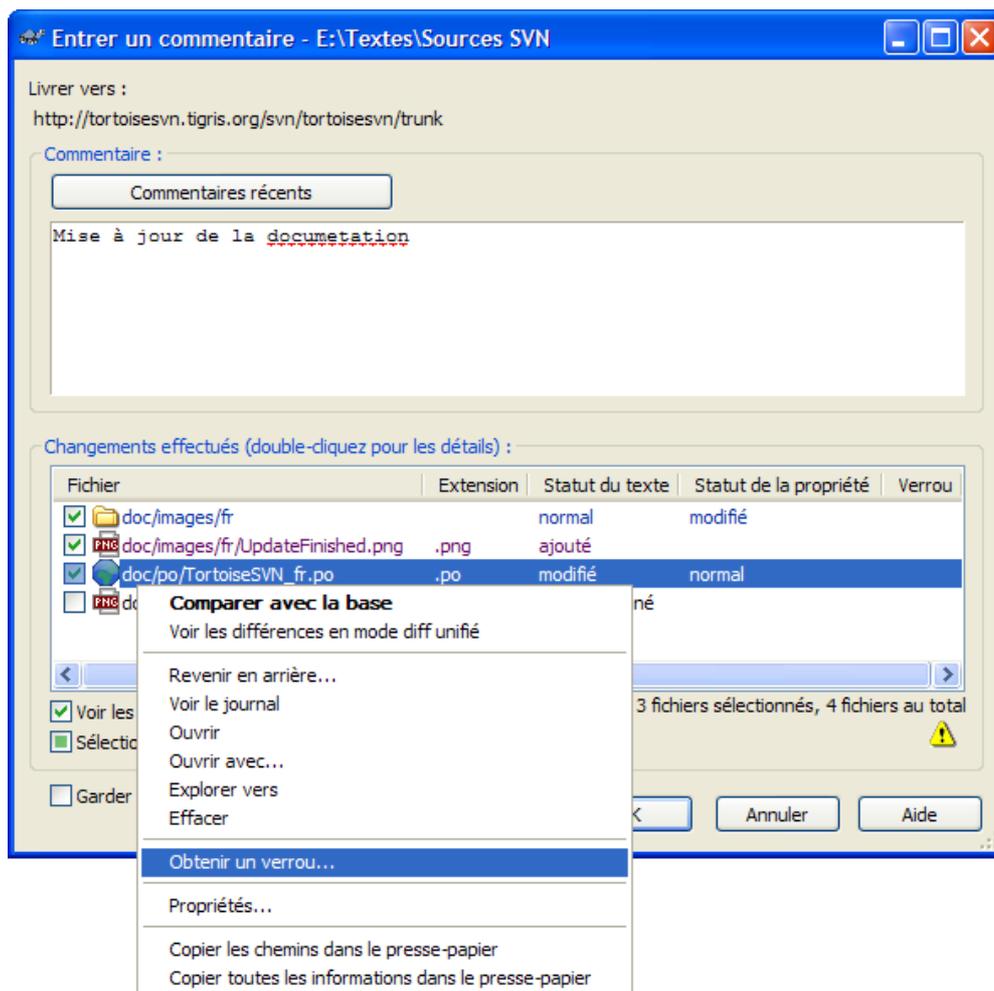


Figure 4.8. La boîte de dialogue Livrer

La boîte de dialogue Livrer vous montrera chaque fichier changé, y compris les fichiers ajoutés, supprimés et non versionnés. Si vous ne voulez pas qu'un fichier modifié soit livré, décochez juste ce fichier. Si vous voulez inclure un fichier non versionné, cochez juste ce fichier pour l'ajouter à la livraison.

Les éléments qui ont été commutés vers un chemin de référentiel différent sont aussi indiqués en utilisant un marqueur (⊗). Vous pouvez avoir commuté quelque chose en travaillant sur une branche et avoir oublié de rebasculer sur le tronc. C'est votre signal d'alarme !



## Livrer des fichiers ou des dossiers ?

Quand vous livrez des fichiers, la boîte de dialogue Livrer montre seulement les fichiers que vous avez choisis. Quand vous livrez un dossier, la boîte de dialogue Livrer choisira les fichiers modifiés automatiquement. Si vous oubliez un nouveau fichier que vous avez créé, livrer le dossier le trouvera de toute façon. La livraison d'un dossier ne veut *pas* dire que chaque fichier est marqué comme changé ; il rend juste votre vie plus facile en faisant plus de travail pour vous.

Si vous avez modifié des fichiers qui ont été inclus à partir d'un référentiel différent en utilisant `svn:externals`, ces changements ne peuvent pas être inclus dans la même livraison atomique. Un symbole d'avertissement au-dessous de la liste de fichier vous indique si cela se produit et l'info-bulle explique que ces fichiers externes doivent être livrés séparément.



## Beaucoup de fichiers non versionnés dans la boîte de dialogue Livrer

Si vous pensez que la boîte de dialogue Livrer de TSVN vous montre trop de fichiers non versionnés (générés par la compilation ou sauvegarde de votre éditeur par exemple), il y a plusieurs solutions. Vous pouvez :

- ajouter le fichier (ou une extension de caractère de remplacement) à la liste de fichiers pour exclure sur la page de configuration. Cela affectera toutes vos copies de travail.
- ajouter le fichier à la liste `svn:ignore` en utilisant TortoiseSVN → Ajouter à la liste des ignorés Cela affectera seulement le répertoire sur lequel vous mettez la propriété `svn:ignore`. En utilisant la boîte de dialogue de propriétés SVN, vous pouvez changer la propriété `svn:ignore` pour un répertoire.

Lire [Section 4.13](#), « Ignorer des fichiers et des répertoires » pour plus d'informations.

Double-cliquer sur un fichier modifié dans la boîte de dialogue Livrer lancera l'outil externe de différenciation pour afficher vos changements. Le menu contextuel vous donnera plus d'options, comme affiché sur la capture d'écran. Vous pouvez aussi glissez les fichiers vers une autre application comme un éditeur de texte ou un IDE à partir d'ici.

Vous pouvez sélectionner ou désélectionner des éléments en cliquant sur la case à cocher située à leur gauche. Pour les répertoires, vous pouvez utiliser **Shift**-sélectionner pour le faire récursivement.

Les colonnes affichées dans le panneau du bas sont personnalisables. Si vous faites un click droit sur n'importe quel en-tête de colonne vous verrez un menu contextuel permettant de choisir quelles colonnes afficher. Vous pouvez aussi changer la largeur des colonnes en faisant glisser le bord des entêtes. Ces personnalisations sont gardées, donc vous verrez les mêmes en-tête la prochaine fois.

Par défaut, lorsque que vous livrez une version, tous les verrous que vous gardez sur des fichiers sont automatiquement retirés à la fin de la livraison. Si vous souhaitez conserver ces verrous, cochez la case **Garder les verrous**. L'état par défaut de cette case, est récupéré dans le fichier de configuration de Subversion, à la clé `no_unlock`. Lisez [Section 4.30.1](#), « Configuration générale » pour plus d'informations sur l'édition du fichier de configuration de Subversion.



## Glisser-déposer

Vous pouvez glisser des fichiers dans la boîte de dialogue Livrer d'ailleurs, tant que les copies de travail sont extraites du même référentiel. Par exemple, vous pouvez avoir une copie de travail énorme avec plusieurs fenêtres d'explorateur ouvertes pour regarder les dossiers éloignés de la hiérarchie. Si vous voulez éviter de livrer le dossier de niveau supérieur (avec un long parcours du dossier pour vérifier les changements) vous pouvez ouvrir la boîte de dialogue de livraison pour un répertoire et y glisser des éléments des autres fenêtres pour les inclure dans la même livraison atomique.

Vous pouvez faire glisser les fichiers non versionnés de votre copie de travail dans la fenêtre de livraison, ils seront alors automatiquement ajoutés au référentiel.



## Réparation des renommages externes

Parfois, les fichiers sont renommés en dehors de Subversion, et son donc affichés comme fichiers manquants et non versionnés. Pour éviter de perdre l'historique vous devez indiquer à Subversion la connexion. Sélectionnez le nom manquant (nom précédent) et le nouveau et utilisez Menu Contextuel → Réparer le Déplacement pour associer les deux fichiers au renommage.

### 4.4.2. Listes de changements

La fenêtre de livraison supporte la fonctionnalité liste des modifications de Subversion et s'en sert pour grouper les fichiers. Pour en savoir plus sur cette fonctionnalité vois [Section 4.8, « Listes de changements »](#).

### 4.4.3. Exclure des éléments de la livraison

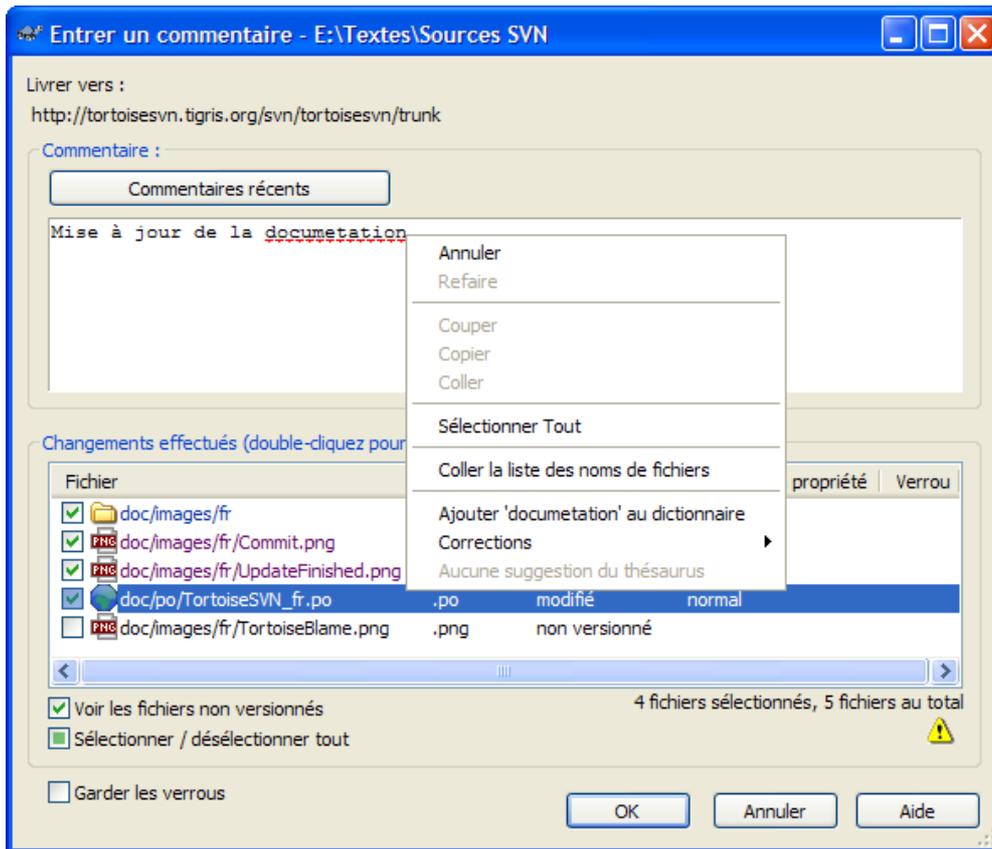
Parfois des fichiers sont fréquemment modifiés mais vous ne voulez pas vraiment en faire la livraison. Cela peut montrer une faille dans votre environnement de travail - Pourquoi ces fichiers sont-ils versionnés ? Ne devriez vous pas utiliser des fichiers templates ? Cependant, c'est parfois inévitable. Une cause classique est que votre fichier de projet contient la date et l'heure de la dernière compilation. Le fichier de projet doit être versionné dans la mesure où il contient toutes les propriétés de compilation, mais pas nécessairement livré vu que seul ce timestamp a changé.

Pour améliorer le traitement de cas bizarres, nous avons créé une liste de modification appelée `ignorer à la livraison`. Tout fichier ajouté dans cette liste sera automatiquement décoché dans la fenêtre de livraison. Vous pouvez cependant toujours livrer les modifications en cochant manuellement les fichiers à livrer dans la fenêtre de livraison.

### 4.4.4. Commentaires de livraison

Assurez-vous d'entrer un commentaire qui décrit les changements que vous livrez. Cela vous aidera à voir ce qui est arrivé et quand, lorsque vous naviguerez dans les commentaires du projet à une date ultérieure. Le message peut être aussi long ou aussi court que vous le souhaitez ; beaucoup de projets ont des directives pour ce qui devrait être inclus, la langue à utiliser et parfois même un format strict.

Vous pouvez appliquer un formatage simple à vos commentaires en utilisant une convention similaire à celle utilisée dans les emails. Pour appliquer un style au `texte`, utilisez `*texte*` pour le gras, `_texte_` pour souligner, et `^texte^` pour l'italique.



**Figure 4.9. Le vérificateur d'orthographe de la boîte de dialogue Livrer**

TortoiseSVN inclut un vérificateur d'orthographe pour vous aider à obtenir des commentaires corrects. Il mettra en évidence les mots mal orthographiés. Utilisez le menu contextuel pour avoir accès aux corrections suggérées. Bien sûr, il ne connaît pas *tous* les termes techniques comme vous, donc parfois les mots correctement orthographiés s'afficheront en tant qu'erreurs. Mais ne vous inquiétez pas. Vous pouvez simplement les ajouter à votre dictionnaire personnel en utilisant le menu contextuel.

La fenêtre de commentaire inclut aussi une fonctionnalité d'auto-achèvement des noms de fichier et des fonctions. Elle utilise des expressions régulières pour extraire les noms des fonctions et des classes à partir des fichiers (texte) que vous livrez, ainsi que des noms de fichier eux-mêmes. Si vous tapez un mot qui correspond à quoi que ce soit dans la liste (après avoir tapé 3 caractères ou avoir appuyé sur la touche **Ctrl+Space**), une liste déroulante apparaît vous permettant de compléter le nom. Les expressions régulières fournies avec TortoiseSVN sont contenues dans le répertoire bin de l'installation de TortoiseSVN. Vous pouvez aussi définir vos propres regexes et les stocker dans %APPDATA%\TortoiseSVN\autolist.txt. Bien sûr cette liste privée ne sera pas écrasée quand vous mettrez à jour votre installation de TortoiseSVN. Si vous n'êtes pas à l'aise avec l'utilisation des expressions régulières, jetez un coup d'oeil à l'introduction [http://en.wikipedia.org/wiki/Regular\\_expression](http://en.wikipedia.org/wiki/Regular_expression) et à la documentation en ligne et au tutoriel : [http://en.wikipedia.org/wiki/Regular\\_expression](http://en.wikipedia.org/wiki/Regular_expression).

Vous pouvez réutiliser les messages de livraison des livraisons précédentes. Cliquez juste sur **Messages récents** pour en voir la liste. Le nombre de messages à garder en mémoire peut être modifié dans la fenêtre de propriété de TortoiseSVN.

Vous pouvez supprimer tous les messages de livraison depuis la page **Données Sauvegardées** de la fenêtre de Configuration de TortoiseSVN, vous pouvez également supprimer les messages précisément depuis la boîte de dialogue **Messages Récents** en utilisant la touche **Suppr.**

Si vous souhaitez inclure les chemins mis à jour dans votre message, vous pouvez utiliser la commande **Menu Contextuel** → **Coller la liste des noms de fichier** dans le champs texte.

Une autre façon d'insérer des chemins dans les messages informatifs est simplement de glisser/déposer les fichiers depuis la liste des fichiers dans le champs d'édition du message.



### Propriétés de dossier spéciales

Il y a plusieurs propriétés de dossier spéciales qui peuvent être utilisées pour aider donner plus de contrôle sur le formatage des commentaires de livraison et la langue utilisée par le module de vérificateur d'orthographe. Lisez [Section 4.17, « Configuration des projets »](#) pour plus d'informations.

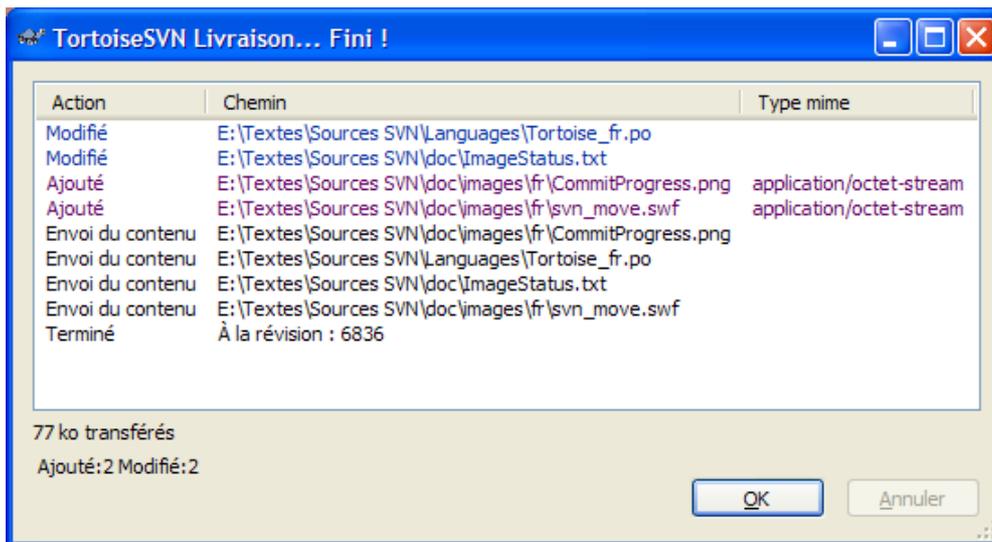


### Intégration aux outils de Bug tracking

Si vous avez activé le système de bugtracking, vous pouvez indiquer un ou plusieurs bugs dans le champ ID Bug / No Incident:. Les ID doivent être séparés par des virgules. Autrement, si vous utilisez un système de bugtracking utilisant les expressions régulières, ajoutez simplement les références des bugs dans le commentaire. Pour en savoir plus : [Section 4.28, « Intégration avec des systèmes de gestion de bug / gestion d'incidents »](#).

## 4.4.5. Progression de la Livraison

Après avoir appuyé sur OK, une boîte de dialogue apparaît affichant la progression de la livraison.



**Figure 4.10. La boîte de dialogue de progression montrant une livraison en cours**

La boîte de dialogue de progression utilise un code couleur pour mettre en évidence les diverses actions de la livraison

Bleu

Livraison d'une modification.

Pourpre

Livraison d'un nouvel ajout.

Rouge foncé

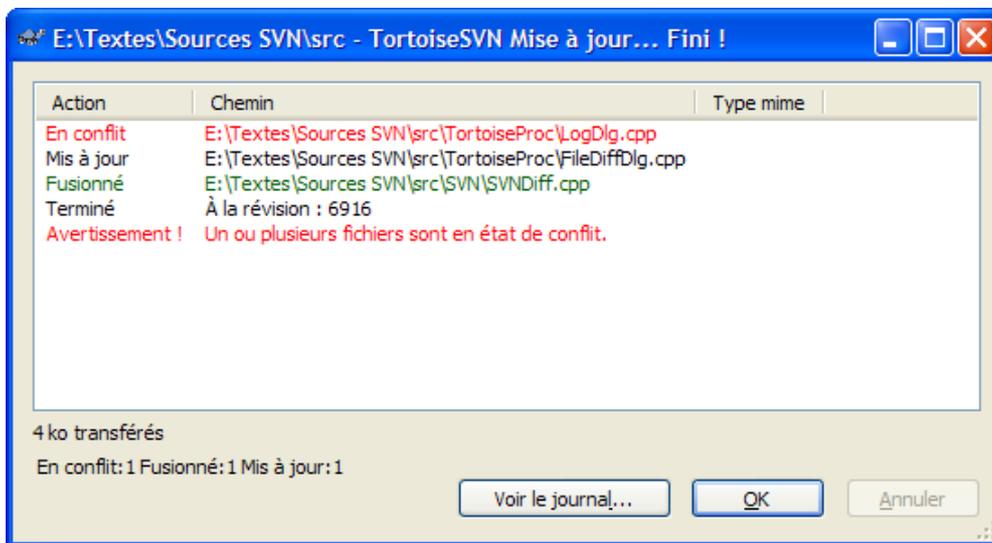
Livraison d'une suppression ou d'un remplacement.

Noir

Tous les autres éléments.

C'est la combinaison de couleur par défaut, mais vous pouvez personnaliser ces couleurs en utilisant la boîte de dialogue de configuration. Lisez [Section 4.30.1.4, « Configuration des couleurs de TortoiseSVN »](#) pour plus d'informations.

## 4.5. Mettre à jour votre copie de travail avec les changements des autres



**Figure 4.11. la boîte de dialogue de progression montrant une mise à jour terminée**

Périodiquement, vous devriez vous assurer que les modifications des autres sont répercutés dans votre copie de travail. Le processus d'obtention des changements du serveur vers votre copie locale s'appelle la mise à jour. La mise à jour peut être faite sur des fichiers seuls, un jeu de fichiers sélectionnés, ou récursivement sur des hiérarchies de répertoire entières. Pour mettre à jour, sélectionnez les fichiers et/ ou les répertoires de votre choix, faites un clic droit et choisissez TortoiseSVN → Mettre à jour dans le menu contextuel de l'explorateur. Une fenêtre apparaîtra montrant la progression de la mise à jour. Les changements faits par les autres seront fusionnés avec vos fichiers, en gardant les changements que vous pourriez avoir faits aux mêmes fichiers. Le référentiel *n'est pas* affecté par une mise à jour.

La boîte de dialogue de progression utilise un code couleur pour mettre en évidence les diverses actions de la mise à jour

Pourpre

Nouvel élément ajouté à votre CdT.

Rouge foncé

Élément redondant supprimé de votre CdT, ou élément manquant remplacé dans votre CdT.

Vert

Changements du référentiel fusionnés avec vos changements locaux avec succès.

Rouge clair

Changements du référentiel fusionnés avec des changements locaux, aboutissant à des conflits que vous devez résoudre.

Noir

Élément inchangé dans votre CdT mis à jour par une version plus récente du référentiel.

C'est la combinaison de couleur par défaut, mais vous pouvez personnaliser ces couleurs en utilisant la boîte de dialogue de configuration. Lisez [Section 4.30.1.4, « Configuration des couleurs de TortoiseSVN »](#) pour plus d'informations.

Si vous avez des *conflits* pendant une mise à jour (cela peut arriver si d'autres ont changé les mêmes lignes dans le même fichier que vous et que ces changements ne correspondent pas) alors la boîte de dialogue montre ces conflits en rouge. Vous pouvez double-cliquer sur ces lignes afin démarrer l'outil externe de fusion pour résoudre les conflits.

Quand la mise à jour est terminée, la boîte de dialogue de progression affiche sous la liste des fichiers un résumé du nombre d'éléments mis à jour, ajoutés, supprimés, en conflit, etc. Ce résumé d'information peut être copié dans le presse-papiers en utilisant **Ctrl+C**.

La commande de mise à jour standard n'a aucune option et met juste à jour votre copie de travail à la révision de tête du référentiel, ce qui est le cas d'utilisation le plus courant. Si vous voulez plus de contrôle sur le processus de mise à jour, vous devriez utiliser TortoiseSVN → **Mettre à jour à la révision...** à la place. Cela vous permet de mettre à jour votre copie de travail à une révision spécifique, et non à la plus récente. Supposons que votre copie de travail soit à la révision 100, mais vous voulez qu'elle reflète l'état qu'elle avait à la révision 50 - alors mettez simplement à jour à la révision 50. Dans la même boîte de dialogue, vous pouvez aussi choisir la *profondeur* à laquelle mettre à jour le répertoire courant. Les termes utilisés sont décrits dans [Section 4.3.1, « Profondeur d'extraction »](#). La profondeur par défaut est **Copie de Travail**, qui préserve le paramètre de profondeur courant. Vous pouvez également choisir d'ignorer les références externes dans la mise à jour (i.e. les projets référencés en utilisant `svn:externals`).



## Attention

Si vous mettez à jour un fichier ou un dossier à une révision spécifique, vous ne devriez pas le modifier. Vous obtiendriez des messages d'erreurs de *péremption* quand vous essayerez de les livrer ! Si vous voulez annuler les changements d'un fichier et recommencer d'une révision précédente, vous pouvez revenir à une révision précédente à partir de la boîte de dialogue de journal de révision. Jetez un coup d'œil à [Section B.4, « Annuler des révisions dans le référentiel »](#) pour avoir plus d'instructions et des méthodes alternatives.

**Mettre à jour à la révision** peut être utile de temps à autre pour voir à quoi ressemblait votre projet à un certain moment. Mais en général, mettre à jour seulement certains fichiers à une révision précédente n'est pas une bonne idée car votre copie de travail est alors dans un état incohérent. Si le fichier que vous mettez à jour a été renommé, vous pouvez même le voir disparaître de votre copie de travail parce qu'aucun fichier de ce nom n'a existé dans la révision précédente. Remarquez également que l'élément aura une icône de recouvrement normale (verte), ce qui le rend complètement indistinct des autres fichiers qui sont à jour.

Si vous voulez simplement une copie locale d'une version ancienne d'un fichier, il est mieux d'utiliser la commande **Menu Contextuel** → **Sauver la révision dans...** depuis la fenêtre de commentaire de ce fichier.



## Plusieurs fichiers / répertoires

Si vous choisissez plusieurs fichiers et plusieurs dossiers dans l'explorateur et sélectionnez ensuite **Mettre à jour**, tous ces fichiers/dossiers sont mis à jour un par un. TortoiseSVN s'assure que tous les fichiers/dossiers qui sont du même référentiel sont mis à jour à la même révision exacte ! Même si une autre livraison se produit entre ces mises à jour.



## Le fichier local existe déjà

Parfois quand vous essayez de mettre à jour, la mise à jour échoue avec un message disant qu'il y a déjà un fichier local du même nom. Cela arrive typiquement quand Subversion essaye d'extraire un fichier nouvellement versionné et constate qu'un fichier non versionné du même nom existe déjà dans votre dossier de travail. Subversion n'écrasera jamais un fichier non versionné - il pourrait contenir quelque chose sur lequel vous travaillez, qui

a par hasard le même nom de fichier qu'un autre développeur a utilisé pour son fichier nouvellement livré.

Si vous obtenez ce message d'erreur, la solution est simplement de renommer le fichier local non versionné. Une fois la mise à jour terminée, vous pouvez vérifier si le fichier renommé est toujours nécessaire.

Si vous continuez à obtenir des messages d'erreur, utilisez TortoiseSVN → Vérifier les modifications pour lister tous les fichiers à problème à la place. De cette façon vous pouvez tous les traiter d'un coup.

## 4.6. Résoudre des conflits

De temps en temps, vous aurez un *conflit* au moment de mettre à jour/fusionner vos fichiers avec le référentiel ou lorsque vous migrerez votre copie de travail vers une autre URL. Il y a deux sortes de conflits:

### Conflit de fichier

Un conflit sur un fichier arrive si deux (ou plus) développeurs changent les mêmes lignes d'un fichier.

### Conflits dans l'arborescence

Un conflit dans l'arborescence arrive quand un développeur déplace/renomme/supprime un fichier ou un dossier, qu'un autre développeur a aussi déplacé/renommé/supprimé voire juste modifié.

### 4.6.1. Conflit de fichiers

A file conflict occurs when two or more developers have changed the same few lines of a file. As Subversion knows nothing of your project, it leaves resolving the conflicts to the developers. Whenever a conflict is reported, you should open the file in question, and search for lines starting with the string <<<<<<. The conflicting area is marked like this:

```
<<<<<< filename
  your changes
=====
  code merged from repository
>>>>>> revision
```

Also, for every conflicted file Subversion places three additional files in your directory:

#### nom\_du\_fichier.ext.mine

C'est votre fichier comme il a existé dans votre copie de travail avant que vous mettiez à jour votre copie de travail - c'est-à-dire sans marqueurs de conflit. Ce fichier a vos derniers changements et rien d'autre.

#### nom\_du\_fichier.ext.OLDREV

C'est le fichier qui était la révision de BASE avant que vous mettiez à jour votre copie de travail. C'est-à-dire le fichier que vous avez extrait avant de faire votre dernière édition.

#### nom\_du\_fichier.ext.rNEWREV

C'est le fichier que votre client de Subversion venait de recevoir du serveur quand vous avez mis à jour votre copie de travail. Ce fichier correspond à la révision de tête du référentiel.

Vous pouvez soit lancer un outil externe de fusion / un éditeur de conflit avec TortoiseSVN → Éditer les conflits soit utiliser un autre éditeur pour résoudre le conflit manuellement. Vous devrez décider à quoi devrait ressembler le code, faites les changements nécessaires et sauvegardez le fichier.

Exécutez ensuite la commande TortoiseSVN → Résolu... et livrez vos modifications au référentiel. Veuillez noter que la commande de résolution ne résout pas vraiment le conflit. Il supprime juste les fichiers `nom_du_fichier.ext.mine` et `nom_du_fichier.ext.r*`, pour vous permettre de livrer vos changements.

Si vous avez des conflits avec des fichiers binaires, Subversion n'essaye pas de fusionner les fichiers lui-même. Le fichier local reste inchangé (exactement comme la dernière fois que vous l'avez modifié) et vous avez des fichiers `nom_du_fichier.ext.r*`. Si vous voulez renoncer à vos changements et garder la version du référentiel, utilisez simplement la commande Revenir en arrière. Si vous voulez garder votre version et écraser la version du référentiel, utilisez la commande Résolu..., livrez ensuite votre version.

Vous pouvez utiliser la commande Résolu... pour plusieurs fichiers si vous faites un clic droit sur le dossier parent et sélectionnez TortoiseSVN → Résolu... Cela affichera une boîte de dialogue listant tous les fichiers en conflit dans ce dossier et vous pouvez choisir lesquels marquer comme résolus.

## 4.6.2. Conflits dans l'arborescence

Un conflit dans l'arborescence est causé quand un développeur déplace/renomme/supprime un fichier ou un répertoire, qu'un autre développeur a également déplacé/renommé/supprimé ou juste modifié. Plusieurs situations peuvent générer un conflit d'arborescence, et chacune a une solution différente.

Dans Subversion, lorsque qu'un fichier est supprimé, il l'est aussi localement, de ce fait, même s'il génère un conflit d'arborescence, aucune action de résolution de conflit de ne peut être effectuée grâce au menu contextuel du click droit. Utilisez la fenêtre Vérifier les Modifications à la place pour pouvoir accéder aux options Edition des conflits.

TortoiseSVN peut aider à trouver les endroits où fusionner les modifications, mais cela ne résoudra pas tous les conflits. Souvenez vous qu'après une mise à jour la base de travail contiendra la révision des éléments tels qu'ils étaient au moment de la mise à jour. Si vous annulez une modification après avoir fait une mise à jour l'élément reviendra à la version du référentiel, pas à l'état dans lequel était l'élément avant que vous fassiez vos modifications.

### 4.6.2.1. Suppression locale, édition de la mise à jour

1. Le développeur A modifie `Foo.c` et en fait la livraison
2. Le développeur B a renommé simultanément `Foo.c` en `Bar.c` dans sa copie de travail, ou simplement supprimé `Foo.c` ou son répertoire parent.

Une mise à jour de la copie de travail du développeur B mène à un conflit dans l'arborescence:

- `Foo.c` a été supprimé de la copie de travail, mais est marqué comme étant en conflit d'arborescence.
- Si le conflit vient d'un renommage plus que d'une suppression alors `Bar.c` est marqué comme ajouté, mais ne contient pas les modifications du développeur A.

Le développeur B doit maintenant choisir s'il veut conserver les modifications du développeur A. Dans le cas d'un renommage de fichier, il peut fusionner les modifications de `Foo.c` dans le fichier renommé `Bar.c`. Pour des suppressions de fichier ou de répertoire il peut choisir de garder l'élément avec les modifications du développeur A et annuler la suppression. Ou, en ne faisant que marquer le conflit comme étant résolu, il annule les modifications du développeur A.

La fenêtre d'édition de conflits offre la possibilité de fusionner les changements si elle peut trouver le fichier original de celui renommé en `Bar.c`. Dépendant d'où la mise à jour a été invoqué, il n'est pas toujours possible de trouver le fichier source.

### 4.6.2.2. Modification locale, suppression dans la mise à jour

1. Le développeur A renomme `Foo.c` en `Bar.c` et envoie les modifications au dépôt.
2. Le développeur B modifie `Foo.c` dans sa copie de travail.

Ou dans le cas d'un déplacement de répertoire ...

1. Le développeur A déplace le répertoire parent FooFolder vers BarFolder et soumet celui-ci au dépôt.
2. Le développeur B modifie Foo.c dans sa copie de travail.

Une mise à jour de la copie de travail du développeur B mène à un conflit dans l'arborescence. Pour un simple conflit de fichier :

- Bar.c est ajouté dans la copie de travail avec le statut 'normal'.
- Foo.c est marqué comme ayant été ajouté (avec historique) et étant en conflit dans l'arborescence.

Pour un conflit de répertoire:

- Bar.c est ajouté dans la copie de travail avec le statut 'normal'.
- Foo.c est marqué comme ayant été ajouté (avec historique) et étant en conflit dans l'arborescence.

Foo.c est marqué comme modifié.

Le développeur B doit à présent décider de garder la réorganisation du développeur A et de fusionner ses changements dans le fichier correspondant dans la nouvelle architecture, ou simplement annuler les modifications du développeur A et garder le fichier local.

Pour fusionner ses modifications locales avec la réorganisation, le développeur B doit d'abord trouver quel est le nouveau nom du fichier Foo.c dans le référentiel. Cela peut être fait en utilisant la fenêtre de commentaires. A présent les modifications doivent être réintégrées à la main dans le mesure où il n'y a aucun outil existant pour simplifier ce processus. Dès que les modifications ont été répercutées, le chemin conflictuel est redondant et peut donc être supprimé. Dans ce cas utilisez le bouton **Supprimer** dans la fenêtre de résolution de conflits pour faire le nettoyage et marquer le conflit comme résolu.

Si le développeur B décide que les modifications du développeur A étaient mauvaises alors elle doit choisir le bouton **Garder** dans la fenêtre de résolution des conflits. Cette action a pour effet de marquer le conflit du fichier/répertoire comme résolu, mais les modifications du développeur A doivent être retirées à la main. De nouveau la fenêtre de commentaires aide à trouver ce qui a été déplacé.

#### 4.6.2.3. Suppression locale, suppression dans la mise à jour

1. Le développeur A renomme Foo.c en Bar.c et envoie les modifications au dépôt
2. Le développeur B renomme Foo.c en Bix.c

Une mise à jour de la copie de travail du développeur B mène à un conflit dans l'arborescence:

- Bix.c est marqué comme ajouté avec l'historique.
- Bar.c est ajouté dans la copie de travail avec le statut 'normal'.
- Foo.c est marqué comme étant supprimé et étant en conflit dans l'arborescence.

Pour résoudre ce conflit, le développeur B doit trouver à quel fichier renommé/déplacé correspond la version conflictueuse de Foo.c dans le référentiel. Ce peut être fait en utilisant la fenêtre de log.

Le développeur B doit alors décider quel nouveau nom de Foo.c garder - celui du développeur A ou celui issu du renommage qu'il a lui même effectué.

Dès que le développeur B a résolu manuellement le conflit, l'arborescence des conflits doit être marquée comme résolue grâce au bouton approprié dans la fenêtre d'édition des conflits.

#### 4.6.2.4. Copie locale incomplète, modification dans la mise à jour

1. Le développeur A travaillant sur le tronc modifie Foo.c et en fait la livraison au dépôt.

2. Le développeur B travaillant sur une branche renommée `Foo.c` en `Bar.c` et en fait la livraison au dépôt

Une fusion des modifications du tronc du développeur A dans la branche de copie de travail du développeur B conduit à un conflit dans l'arborescence:

- `Bar.c` est déjà dans la copie de travail avec le statut 'normal'.
- `Foo.c` est marqué comme manquant dans l'arborescence des conflits.

Pour résoudre ce conflit, le développeur B doit marquer le fichier comme résolu dans la fenêtre de résolution de conflits, lequel sera retiré de la liste des conflits. Le développeur B doit alors décider s'il faut copier le fichier manquant `Foo.c` depuis le référentiel, ou fusionner `Foo.c` avec le fichier renommé `Bar.c` ou simplement s'il faut ignorer les changements en marquant le conflit comme étant résolu et ne rien faire d'autre.

Notez que si vous copiez le fichier manquant depuis le référentiel et que vous le marquez comme résolu, votre copie sera de nouveau retirée. Vous devez résoudre d'abord le conflit.

#### 4.6.2.5. Modification locale, suppression dans la fusion

1. Le développeur A travaillant sur le tronc renommé `Foo.c` en `Bar.c` et l'envoie au dépôt
2. Le développeur B travaillant sur une branche modifie le fichier `Foo.c` et l'envoie au dépôt.

Il y a des cas équivalents pour les déplacements de répertoire, mais ce n'est pas encore détecté dans Subversion 1.6 ...

1. Le développeur A travaillant sur le tronc déplace le dossier parent `FooFolder` vers `BarFolder` et l'envoie au dépôt.
2. Le développeur B travaillant sur une branche modifie `Foo.c` dans sa copie de travail.

Une fusion des modifications du tronc du développeur A dans la branche de copie de travail du développeur B conduit à un conflit dans l'arborescence:

- `Bar.c` est marqué comme étant ajouté.
- `Foo.c` est marqué comme ayant été modifié et étant en conflit dans l'arborescence.

Le développeur B doit à présent décider de garder la réorganisation du développeur A et de fusionner ses changements dans le fichier correspondant dans la nouvelle architecture, ou simplement annuler les modifications du développeur A et garder le fichier local.

Pour fusionner ses modifications locales avec les autres remaniements, le développeur B doit trouver à quel fichier renommé/déplacé correspond la version conflictueuse de `Foo.c` dans le dépôt. Cela peut être fait en utilisant la fenêtre de commentaires du fichier source. L'éditeur de conflit ne montre que les commentaires relatifs à la copie de travail puisqu'il ne sait pas quel chemin a été utilisé lors de la fusion, donc vous devrez le découvrir vous-même. Les modifications doivent alors être fusionnées à la main puisqu'il n'existe actuellement aucun moyen d'automatiser ou de simplifier ce processus. Une fois que les changements ont été effectués, le chemin conflictueux est superflu et peut être supprimé. Dans ce cas, utilisez le bouton **Supprimer** dans la fenêtre d'édition de conflits pour marquer le conflit comme étant résolu.

Si le développeur B décide que les modifications du développeur A étaient mauvaises elle doit alors choisir le bouton **Garder** dans la fenêtre de résolution de conflit. Cette action marque le conflit du fichier/répertoire comme étant résolu, mais le développeur A doit reporter ses modifications à la main. La fenêtre de commentaires du fichier source de la fusion aide à détecter ce qui a été déplacé.

#### 4.6.2.6. Suppression locale, suppression dans la fusion

1. Le développeur A travaillant sur le tronc renommé `Foo.c` en `Bar.c` et l'envoie au dépôt

2. Le développeur B travaillant sur une branche renommée Foo.c en Bix.c et envoie les modifications au dépôt

Une fusion des modifications du tronc du développeur A dans la branche de copie de travail du développeur B conduit à un conflit dans l'arborescence:

- Bix.c est marqué comme étant dans un état normal (non modifié).
- Bar.c est marqué comme ajouté avec son historique.
- Foo.c est marqué comme manquant et générant une erreur dans l'arborescence.

Pour résoudre ce conflit, le développeur B doit trouver quel fichier renommé/déplacé correspond à la version conflictueuse de Foo.c dans le référentiel. Ce peut être fait en utilisant la fenêtre de commentaires pour les sources fusionnées. L'éditeur de conflit ne montre que les commentaires relatifs à la copie de travail puisqu'il ne sait pas quel chemin a été utilisé lors de la fusion, donc vous devrez le découvrir vous même.

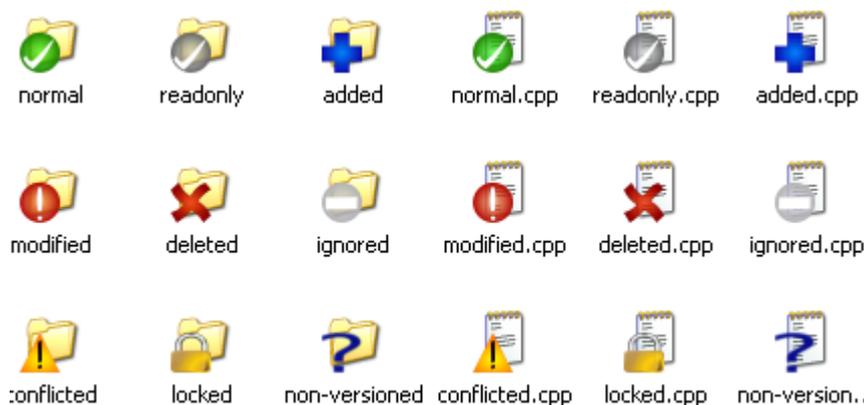
Le développeur B doit alors décider quel nouveau nom de Foo.c garder - celui du développeur A ou celui issu du renommage qu'il a lui même effectué.

Dès que le développeur B a résolu manuellement le conflit, l'arborescence des conflits doit être marquée comme résolue grâce au bouton approprié dans la fenêtre d'édition des conflits.

## 4.7. Obtenir des information sur le statut

Pendant que vous travaillez sur votre copie de travail, vous avez souvent besoin de savoir quels fichiers vous avez changé/ajouté/supprimé ou renommé, ou même quels fichiers ont été changé et livré par les autres.

### 4.7.1. Recouvrement d'icônes



**Figure 4.12. L'Explorateur montrant le recouvrement d'icônes**

Maintenant que vous avez extrait une copie de travail à partir d'un référentiel Subversion, vous pouvez voir que vos fichiers ont d'autres icônes dans l'explorateur Windows. C'est une des raisons pour laquelle TortoiseSVN est si populaire. TortoiseSVN ajoute une icône de recouvrement à chaque icône de fichier chevauchant l'icône de fichier original. Selon le statut Subversion du fichier, l'icône de recouvrement est différente.



Une copie de travail fraîchement extraite a une coche verte comme icône de recouvrement. Cela signifie que le statut Subversion est normal.



Dès que vous commencez à éditer un fichier, le statut change en `modifié` et l'icône de recouvrement devient alors un point d'exclamation rouge. De cette façon, vous pouvez facilement voir quels fichiers ont été modifiés depuis votre dernière mise à jour de la copie de travail et ont donc besoin d'être livrés.



Si un `conflit` se produit lors d'une mise à jour alors l'icône de recouvrement devient un point d'exclamation jaune.



Si vous avez mis la propriété `svn:needs-lock` sur un fichier, Subversion met ce fichier en lecture seule jusqu'à ce que vous verrouilliez ce fichier. Ces fichiers ont cette icône de recouvrement pour indiquer que vous devez d'abord les verrouiller pour pouvoir les éditer.



Si vous avez verrouillé un fichier et que le statut de Subversion est `normal`, cette icône de recouvrement vous rappelle que vous devriez relâcher le verrou si vous ne l'utilisez pas pour permettre aux autres utilisateurs de livrer leurs changements.



Cette icône vous montre que quelques fichiers ou dossiers à l'intérieur du dossier ont été marqués comme *supprimés* du contrôle de version ou qu'un fichier est manquant.



Le signe plus vous indique qu'un fichier ou un dossier a été marqué comme étant *ajouté* au contrôle de version.



Le signe "sens interdit" vous indique qu'un fichier ou un dossier a été marqué comme étant à *ignorer* par le contrôle de version. Cette indication est optionnelle.



Cette icône est associée aux éléments qui ne sont pas sous contrôle de version, et qui ne sont pas à ignorer. Cette icône de recouvrement est facultative.

En fait, vous pouvez constater que ces icônes ne sont pas toutes utilisées sur votre système. C'est dû au fait que le nombre de recouvrements permis par Windows est très limité et si vous utilisez aussi TortoiseCVS, alors il n'y a pas assez d'emplacements de recouvrement disponibles. TortoiseSVN essaie d'être un « Bon Citoyen (TM) » et limite son utilisation des recouvrements pour laisser de la place aux autres applications.

Depuis que différents clients Tortoise sont disponibles (TortoiseCVS, TortoiseHG, ...) ces limites deviennent un réel problème. Pour contourner ce souci, le projet TortoiseSVN met à présent à votre disposition une série d'icônes dans une DLL, qui peuvent être utilisées par tous les clients Tortoise. Vérifiez avec votre fournisseur pour savoir si cette fonctionnalité a été intégrée :-)

Pour avoir une description des icônes de recouvrement correspondant aux statuts de Subversion et d'autres détails techniques, lisez [Section F.1, « Recouvrement d'icônes »](#).

## 4.7.2. Les colonnes de TortoiseSVN dans l'explorateur Windows

Les mêmes informations, disponibles avec le recouvrement d'icônes, (et plus encore) peuvent être affichées avec des colonnes supplémentaires dans l'affichage en détail de l'explorateur Windows.

Faites simplement un clic droit sur un des titres d'une colonne, choisissez Autres... dans le menu contextuel. Une boîte de dialogue apparaîtra dans laquelle vous pourrez spécifier les colonnes à afficher dans la « Vue détaillée » ainsi que leur ordre. Faites défiler pour que s'affichent les entrées commençant par SVN. Cochez celles vous voudriez afficher et fermez la boîte de dialogue en appuyant sur OK. Les colonnes seront ajoutées à droite de celles déjà affichées. Vous pouvez les réordonner par glisser-déposer, ou les redimensionner, pour qu'elles correspondent à vos besoins.



### Important

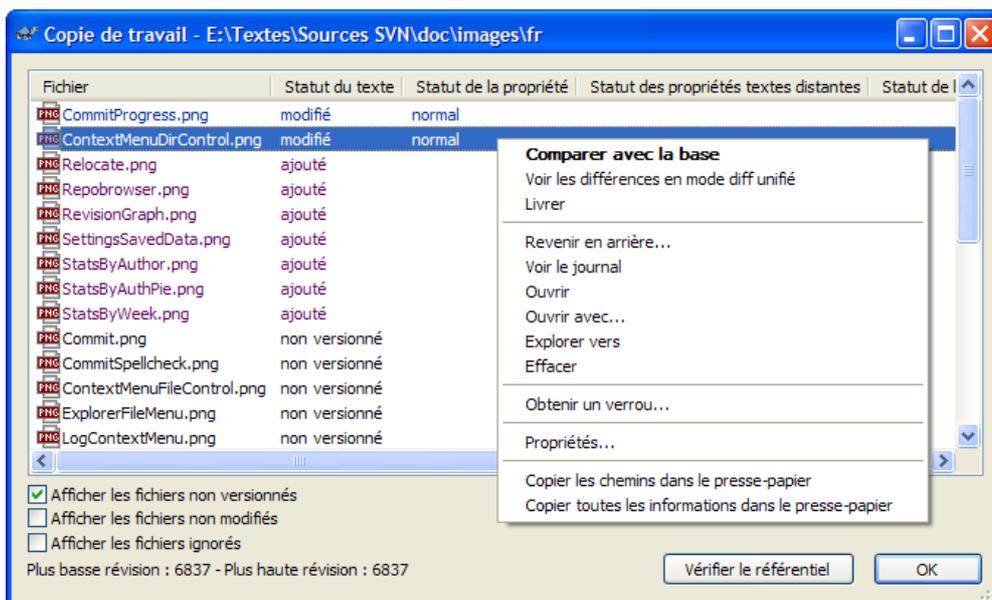
Les colonnes supplémentaires dans l'Explorateur Windows ne sont pas disponibles sous Vista, Microsoft n'autorisant plus ce type de colonnes pour *tous* les fichiers, mais plus que pour certains types de fichiers.



### Astuce

Si vous voulez que la disposition actuelle soit affichée dans toutes vos copies de travail, vous pouvez en faire la vue par défaut.

## 4.7.3. Statut local et distant



**Figure 4.13. Vérifier les modifications**

Il est souvent très utile de savoir quels fichiers vous avez changé et aussi quels fichiers ont été changés et livrés par les autres. C'est là où la commande TortoiseSVN → Vérifier les modifications... devient pratique. Cette boîte de dialogue vous montrera les fichiers ayant été modifié dans votre copie de travail, ainsi que les fichiers non versionnés que vous pouvez avoir.

Si vous cliquez sur Vérifier le référentiel alors vous pouvez aussi chercher les changements dans le référentiel. De cette manière vous pouvez vérifier avant une mise à jour s'il y a un conflit potentiel. Vous pouvez aussi mettre à jour certains fichiers du référentiel sans mettre à jour le dossier entier. Par défaut, le bouton Vérifier le référentiel ne récupère que le statut du référentiel à la profondeur de la copie de travail. Si vous voulez voir tous les fichiers et répertoires du référentiel, même ceux qui n'ont pas été extraits, vous devez cliquer sur le bouton Vérifier le référentiel en appuyant sur la touche **Shift**.

La boîte de dialogue utilise un code couleur pour mettre le statut en évidence .

Bleu

Éléments modifiés localement

Pourpre

Éléments ajoutés. Les éléments qui ont été ajoutés avec un historique ont un signe + dans la colonne Statut du texte et une info-bulle montre d'où l'article a été copié.

Rouge foncé

Éléments supprimés ou manquants.

Vert

Éléments modifiés localement et dans le référentiel. Les changements seront fusionnés lors de la mise à jour. Cela *peut* produire des conflits à la mise à jour.

Rouge clair

Éléments modifiés localement et supprimés dans le référentiel, ou modifiés dans le référentiel et supprimés localement. Cela *va* produire des conflits à la mise à jour.

Noir

Éléments inchangés et non versionnés.

C'est la combinaison de couleur par défaut, mais vous pouvez personnaliser ces couleurs en utilisant la boîte de dialogue de configuration. Lisez [Section 4.30.1.4, « Configuration des couleurs de TortoiseSVN »](#) pour plus d'informations.

Les éléments qui ont été commutés vers un chemin de référentiel différent sont aussi indiqués en utilisant un marqueur ( $\$$ ). Vous pouvez avoir commuté quelque chose en travaillant sur une branche et avoir oublié de rebasculer sur le tronc. C'est votre signal d'alarme !

À partir du menu contextuel de la boîte de dialogue vous pouvez afficher une comparaison des changements. Vérifiez les changements locaux que *vous* avez fait en utilisant **Menu contextuel** → **Comparer avec la Base**. Vérifiez les changements du référentiel faits par les autres en utilisant **Menu contextuel** → **Voir les différences en mode diff unifié**.

Vous pouvez aussi annuler des changements dans des fichiers individuels. Si vous avez supprimé un fichier accidentellement, il apparaîtra comme *Manquant* et vous pourrez utiliser *Revenir en arrière* pour le récupérer.

Les fichiers non versionnés et les fichiers ignorés peuvent être envoyés dans la corbeille à partir d'ici en utilisant **Menu contextuel** → **Supprimer**. Si vous voulez supprimer définitivement les fichiers (sans passer par la corbeille) maintenez la touche **Maj** pendant que vous cliquez sur **Supprimer**.

Si vous voulez examiner un fichier en détail, vous pouvez le glisser d'ici vers une autre application comme un éditeur de texte ou un IDE.

Les colonnes affichées sont personnalisables. Si vous faites un clic droit sur n'importe quel en-tête de colonne un menu contextuel vous permettant de choisir les colonnes à afficher apparaîtra. Vous pouvez aussi changer la largeur de colonne en faisant glisser le bord des entêtes. Ces personnalisations sont préservées, donc vous verrez les mêmes en-tête la prochaine fois.

Si vous travaillez sur plusieurs tâches distinctes en même temps, vous pouvez regrouper les fichiers dans des listes de modification. Lisez [Section 4.4.2, « Listes de changements »](#) pour plus d'informations.

En bas de la boîte de dialogue vous pouvez apercevoir un sommaire de l'éventail des révisions en mémoire utilisées dans votre copie active. Celles-ci sont du type *Livrer*, et non de type *mise à jour*; elles représentent l'éventail de révisions où ces fichiers ont été livrés dernièrement, et non les révisions vers lesquelles elles ont été mises à jour. Notez que l'éventail de révisions présentées s'applique uniquement

aux éléments affichés, et non à la copie active entière. Si vous désirez voir cette information pour la copie active dans son intégralité vous devez cocher la case **Montrer les éléments non-modifiés**.



### Astuce

Si vous voulez une vue plate de votre copie de travail, c'est-à-dire qui montre tous les fichiers et les dossiers à chaque niveau de la hiérarchie des dossiers, alors la boîte de dialogue **Vérifier les modifications** est la façon la plus facile d'y arriver. Cochez seulement la case **Afficher les fichiers non modifiés** pour afficher tous les fichiers de votre copie de travail.



### Réparation des renommages externes

Parfois, les fichiers sont renommés en dehors de Subversion, et sont donc affichés comme fichiers manquants et non versionnés. Pour éviter de perdre l'historique vous devez indiquer à Subversion la connexion. Sélectionnez le nom manquant (nom précédent) et le nouveau et utilisez **Menu Contextuel** → **Réparer le Déplacement** pour associer les deux fichiers au renommage.

## 4.7.4. Voir les différences

Souvent vous voulez regarder à l'intérieur de vos fichiers, pour regarder ce que vous avez changé. Vous pouvez accomplir cela en sélectionnant un fichier qui a changé et en choisissant **Voir les différences** à partir du menu contextuel de TortoiseSVN. Cela démarre le visualisateur externe de différence, qui comparera alors le fichier actuel avec la copie primitive (la révision BASE), qui a été stockée après la dernière extraction ou la dernière mise à jour.



### Astuce

Même lorsque vous ne vous trouvez pas dans une copie de travail ou quand vous avez de multiples versions du fichier ici et là, vous pouvez toujours afficher les différences :

Sélectionnez les deux fichiers que vous voulez comparer dans l'explorateur (en utilisant par exemple **Ctrl** et la souris) et sélectionnez **Voir les différences** à partir du menu de contextuel de TortoiseSVN. Le fichier cliqué en dernier (celui avec le focus, c'est-à-dire le rectangle pointillé) sera considéré comme le plus récent.

## 4.8. Listes de changements

Dans un monde idéal, vous ne travaillez jamais que sur une seule chose à la fois, et votre copie active ne contient qu'un seul jeu de changements logiques. OK, retour à la réalité. Il arrive souvent que vous ayez à travailler sur plusieurs tâches à la fois, sans rapport les unes avec les autres, et quand vous regardez dans la boîte de dialogue **Livrer**, toutes les modifications sont mélangées ensemble. L'élément *changelist* vous aide à regrouper les fichiers, ce qui vous aide à voir plus facilement ce que vous faites. Bien sûr, ceci ne peut fonctionner que si les modifications ne se chevauchent pas. Si deux tâches différentes affectent le même fichier, il devient impossible de séparer les modifications.



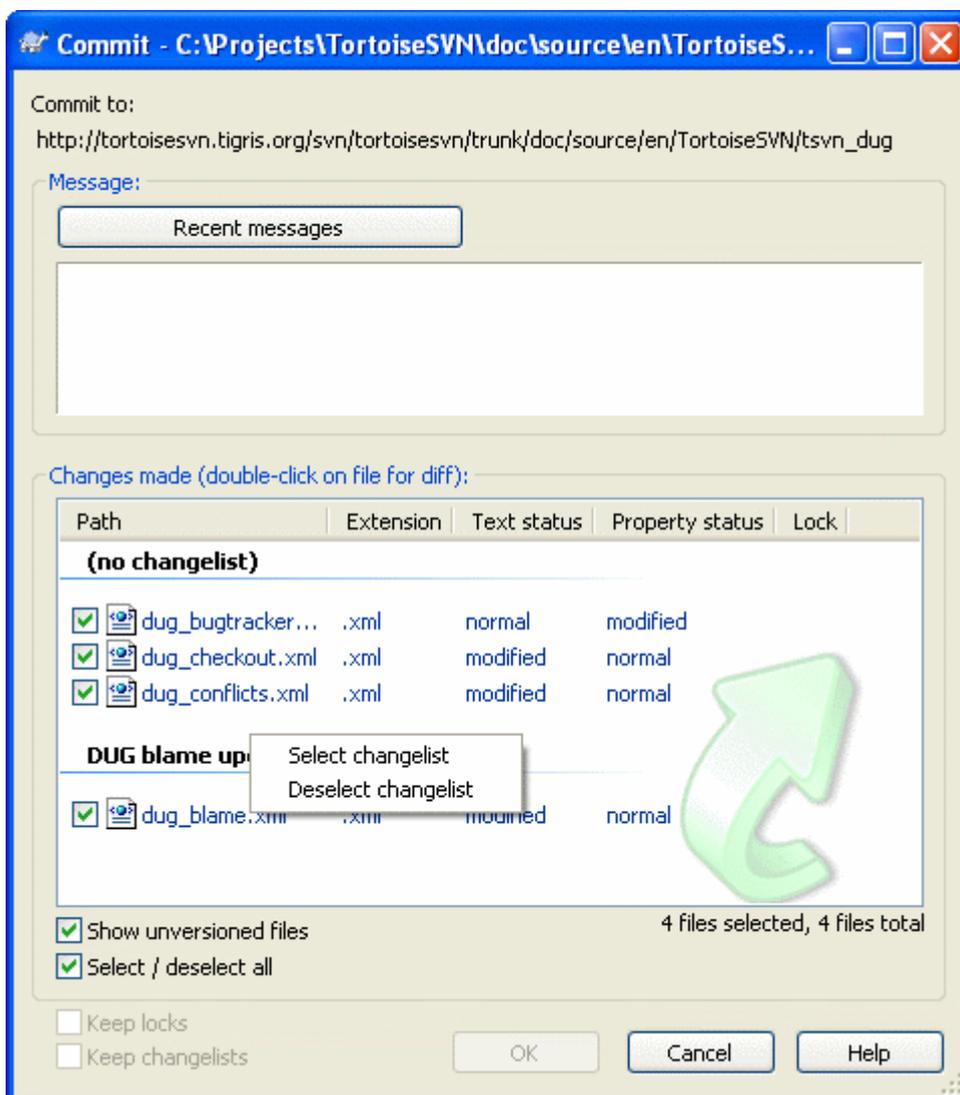
### Important

La fonctionnalité de TortoiseSVN donnant la liste des changements n'est disponible qu'à partir de Windows XP, dans la mesure où elle dépend d'une fonctionnalité du shell qui n'est pas disponible dans Windows 2000. Désolé mais Win2K est vraiment vieux maintenant, donc ne vous plaignez pas.

Vous pouvez voir la liste des modifications à différents endroits, mais les plus importants sont la fenêtre de livraison et la fenêtre de vérification des modifications. Commençons par la fenêtre de vérification des modifications après avoir travaillé sur différentes fonctionnalités dans beaucoup de fichiers. Lorsque vous ouvrez la fenêtre pour la première fois, tous les fichiers modifiés sont listés. Supposons maintenant que vous vouliez tout organiser et grouper ces fichiers par fonctionnalité.

Sélectionnez un ou plusieurs fichiers et utilisez MenuContextuel → Déplacer vers la liste des modifications pour ajouter un élément à une liste de modifications. Initialement il n'y aura pas de liste de modifications, donc la première fois que vous ferez ceci vous devrez créer une nouvelle liste de modifications. Donnez-lui un nom qui décrit ce pour quoi vous l'utilisez, et cliquez sur OK. La boîte de dialogue va maintenant vous montrer des groupes d'éléments.

Dès que vous avez créé une liste de modification vous pouvez y glisser/déposer des éléments depuis une autre liste de modifications, ou depuis Windows Explorer. L'avantage de le faire depuis Explorer est de permettre d'ajouter des éléments avant qu'ils ne soient modifiés. Vous pourriez faire de même depuis la fenêtre de vérification des modifications, mais seulement en affichant tous les fichiers non modifiés.



**Figure 4.14. Fenêtre de livraison avec les listes de modification**

Dans la fenêtre de livraison vous pouvez voir ces mêmes fichiers, regroupés par liste de modifications. En dehors de donner directement une indication sur les regroupements, vous pouvez également vous servir des entêtes des groupes pour sélectionner les fichiers à livrer.

Sous XP, un menu contextuel apparaît lorsque vous faites un clic droit sur l'entête d'un groupe, vous donnant le choix de cocher ou de décocher toutes les entrées. Sous Vista le menu contextuel n'est pas nécessaire. Cliquez sur l'entête du groupe pour tout sélectionner, puis cochez une des entrées sélectionnées pour toutes les cocher.

TortoiseSVN se réserve une liste de modification, appelée `ignore-on-commit`. Elle est utilisée pour marquer les fichiers versionnés que vous souhaitez rarement livrer même si vous les avez modifiés. Cette fonctionnalité est décrite dans [Section 4.4.3, « Exclude des éléments de la livraison »](#).

Lorsque vous livrez des fichiers faisant partie d'une liste de modifications alors vous vous attendez naturellement à ce qu'ils n'en fassent plus partie ensuite. Donc par défaut, les fichiers sont retirés automatiquement de la liste des modifications après avoir été livrés. Si ce comportement ne vous convient pas, utilisez la case à cocher **Garder les listes de modifications** en bas de la fenêtre de livraison.



### Astuce

Les listes de modifications sont une fonctionnalité du client local. Créer et supprimer ces listes n'influencera en rien ni le référentiel, ni les copies de travail des autres. C'est juste un moyen pratique d'organiser vos fichiers.

## 4.9. La boîte de dialogue du Journal de révision

A chaque modification que vous faites et que vous livrez, vous devriez fournir un commentaire décrivant cette modification. De cette façon, vous pouvez par la suite comprendre quelles modifications vous avez faites et pourquoi, et vous disposez d'un journal détaillé pour votre processus de développement.

La boîte de dialogue du Journal de révision récupère tous ces commentaires de révision et vous les présente. L'affichage est divisé en 3 panneaux.

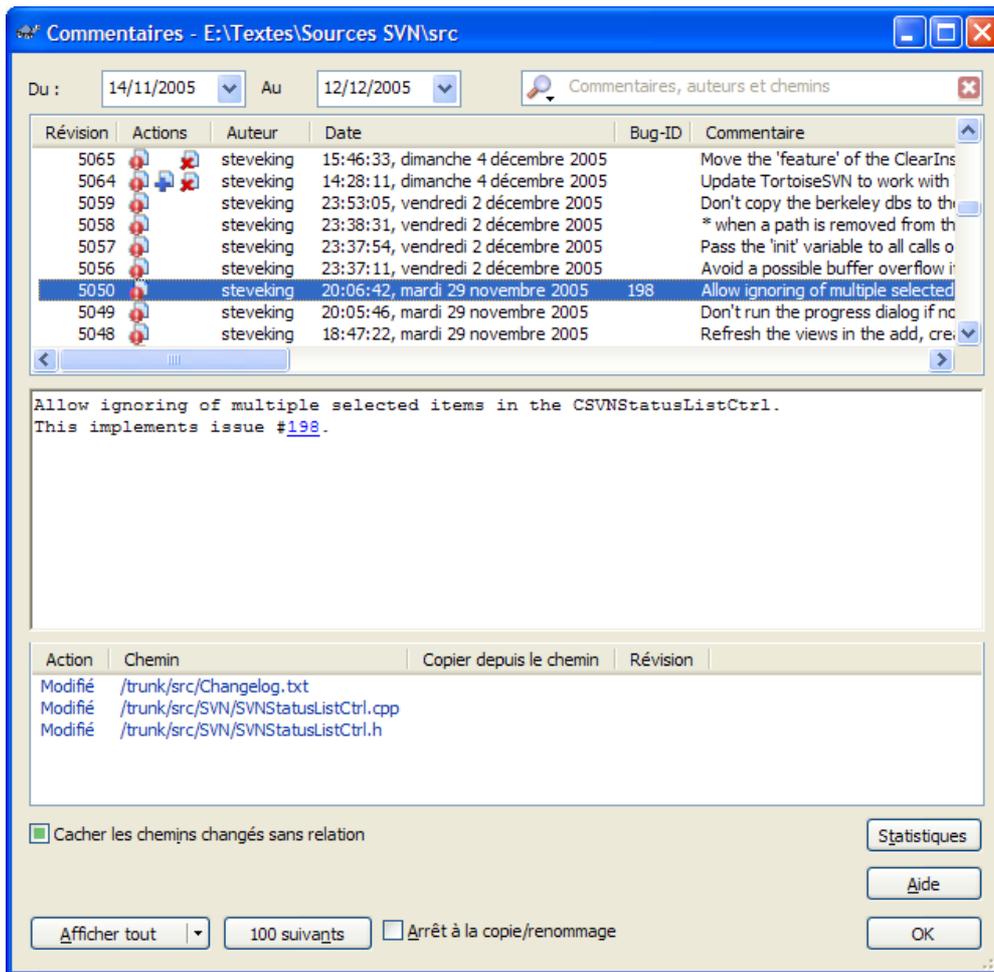
- Le panneau supérieur présente une liste de révisions où des modifications au fichier/dossier ont été livrées. Ce résumé inclut la date et l'heure, la personne qui a livré la révision et le début du commentaire.

Les lignes affichées en bleu indiquent que quelque chose a été copié vers cette ligne de développement (peut-être d'une branche).

- Le panneau du milieu montre le commentaire en entier pour la révision choisie.
- Le panneau du bas montre une liste de tous les fichiers et de tous les dossiers qui ont été modifiés lors de la révision sélectionnée.

Mais elle fait beaucoup plus que cela : elle fournit des commandes de menu contextuel que vous pouvez utiliser pour obtenir encore plus d'informations sur l'historique du projet.

### 4.9.1. Appeler la boîte de dialogue du Journal de révision



**Figure 4.15. La boîte de dialogue du Journal de révision**

Il existe plusieurs endroits à partir desquels vous pouvez afficher la boîte de dialogue de Journal :

- À partir du sous-menu contextuel de TortoiseSVN
- À partir de la page de propriété
- À partir de la boîte de dialogue de progression après qu'une mise à jour a fini. Alors la boîte de dialogue de Journal ne montre que les révisions qui ont été modifiées depuis votre dernière mise à jour

Si le référentiel n'est pas joignable vous verrez une fenêtre Travailler hors ligne ?, décrite dans [Section 4.9.10, « Mode hors ligne »](#).

### 4.9.2. La boîte de dialogue du Journal de révision

Le panneau supérieur a une colonne **Actions** contenant des icônes résumant ce qui a été fait dans cette révision. Il y a quatre icônes différents, chacun est affiché dans sa propre colonne.



Si une révision modifie un élément, l'icône *modifié* est affichée dans la première colonne.



Si un élément a été ajouté lors d'une révision, l'icône *ajouté* est affichée dans la seconde colonne.

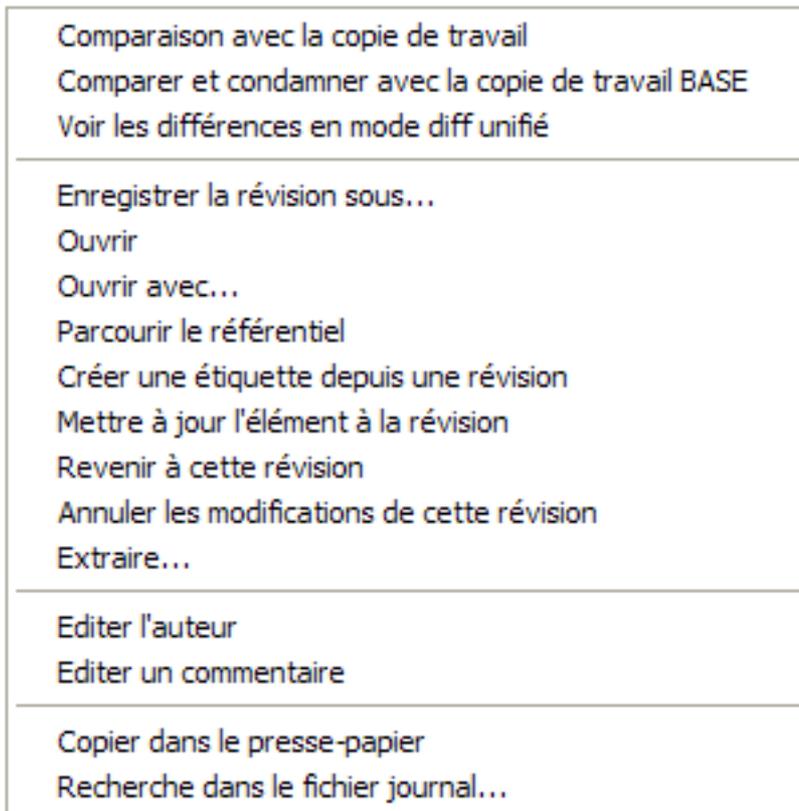


Si un élément a été supprimé dans une révision, l'icône *deleted* s'affiche dans la troisième colonne.



Si un élément a été remplacé, l'icône *remplacé* est affichée dans la quatrième colonne.

### 4.9.3. Obtenir des informations supplémentaires



**Figure 4.16. Le panneau supérieur de la boîte de dialogue du Journal de révision avec le menu contextuel**

De plus amples informations sont disponibles via un menu contextuel dans la panneau supérieur de la boîte de dialogue du Journal de révision. Certains éléments de ce menu sont spécifiques aux fichiers, d'autres aux dossiers.

#### Comparer avec la copie de travail

Comparer la révision sélectionnée avec votre copie de travail. L'outil de différenciation par défaut est TortoiseMerge qui est fourni avec TortoiseSVN. Si la boîte de dialogue de journal concerne un dossier, cela vous montrera une liste de fichiers modifiés et vous permettra de passer en revue les modifications apportées à chaque fichier individuellement.

#### Comparer et annoter avec la BASE de travail

Annoter la révision sélectionnée et le fichier dans votre base de travail et comparer les résultats en utilisant un outil de différenciation. Lire [Section 4.23.2, « Annoter les différences »](#) pour plus de détails (uniquement pour les fichiers).

#### Voir les modifications comme des différences unifiées

Voir les modifications faites dans la révision sélectionnée en tant que fichier de différences unifiées (format de patch GNU). Cela montre seulement les différences avec quelques lignes de contexte.

Cela est plus difficile à lire qu'une comparaison visuelle de fichiers, mais cela vous présentera tous les changements dans un format compact.

#### Comparer avec la révision précédente

Compare la révision sélectionnée avec la révision précédente. Cette fonctionnalité est similaire à la comparaison avec votre copie de travail. Pour les dossiers, cette option montrera d'abord la boîte de dialogue des fichiers modifiés, vous permettant de sélectionner les fichiers à comparer.

#### Comparer avec la révision précédente et annoter

Montre la boîte de dialogue des fichiers modifiés vous permettant de sélectionner les fichiers. Annote la révision sélectionnée, la révision précédente et compare les résultats en utilisant un outil de comparaison. (valable pour les répertoires uniquement).

#### Sauvegarder la révision dans...

Sauvegarder la révision sélectionnée dans un fichier pour avoir une version antérieure de ce fichier (valable uniquement pour les fichiers).

#### Ouvrir / Ouvrir avec...

Ouvrir le fichier sélectionné, avec l'application par défaut pour ce type de fichier, ou avec le programme de votre choix (valable uniquement pour les fichiers).

#### Annoter...

Annoter le fichier jusqu'à la révision sélectionnée (uniquement pour les fichiers).

#### Parcourir le référentiel

Ouvrir l'explorateur de référentiel pour examiner le fichier ou le dossier sélectionné dans le référentiel tel qu'il était à la révision sélectionnée.

#### Créer une branche/une étiquette depuis la révision

Créer une branche ou une étiquette à partir d'une révision choisie. C'est utile par exemple si vous avez oublié de créer une étiquette et avez déjà livré des modifications qui n'étaient pas supposées être intégrées à cette révision.

#### Mettre à jour l'élément à la révision

Mettre à jour votre copie de travail à la révision sélectionnée. Utile si vous voulez que votre copie de travail soit le reflet d'un état précis du passé ou s'il y a eu depuis d'autres livraisons dans le référentiel et que vous voulez mettre à jour votre copie de travail par étapes successives. Il est préférable de mettre à jour un répertoire entier de votre copie de travail, et pas seulement un fichier, autrement votre copie de travail pourrait être incohérente.

Si vous voulez annuler définitivement une modification, utilisez **Revenir à cette révision**.

#### Revenir à cette révision

Revenir à une révision antérieure. Si vous avez fait plusieurs modifications et décidez ensuite que vous voulez vraiment remettre les choses telles qu'elles étaient à la révision N, c'est la commande dont vous avez besoin. Les modifications sont annulées dans votre copie de travail : cette opération n'affecte donc *pas* le référentiel tant que vous n'avez pas livré ces changements. Notez que cela annulera *toutes* les modifications faites après la révision sélectionnée, remplaçant le fichier/dossier avec cette révision antérieure.

Si votre copie de travail n'a pas été modifiée, après avoir fait cette action elle sera marquée comme étant modifiée. Si vous avez déjà des modifications locales, cette commande fusionnera les modifications *annulées* dans votre copie de travail.

En interne, Subversion annule la fusion de toutes les modifications effectuées après la révision sélectionnée, revenant ainsi à l'état précédant ces livraisons.

Après avoir effectué cette action, si vous voulez *annuler l'annulation* et retrouver votre copie active dans son état précédent non-modifié, nous vous conseillons d'utiliser TortoiseSVN → Revert à

partir de Windows Explorer, ce qui aura pour effet de rejeter les modifications locales effectuées par cette action de dé-fusion.

Si vous voulez juste avoir un aperçu d'un fichier ou d'un répertoire dans une révision antérieure, utilisez **Revenir à cette révision** ou **Sauvegarder la révision sous...**

#### Annuler les modification pour revenir à cette révision

Annuler les modifications effectuées à la révision sélectionnée. Les modifications sont annulées dans votre copie de travail donc cette opération n'affecte *pas* du tout le référentiel ! Notez que cela annulera les modifications de cette révision seulement. Cela ne remplace pas votre copie de travail par le fichier complet de la révision précédente. C'est très utile pour annuler une modification antérieure quand d'autres modifications sans rapport ont été faites depuis.

Si votre copie de travail n'a pas été modifiée, après avoir fait cette action elle sera marquée comme étant modifiée. Si vous avez déjà des modifications locales, cette commande fusionnera les modifications *annulées* dans votre copie de travail.

En interne, Subversion annule la fusion de cette révision particulière, annulant l'effet d'une Livraison antérieure.

Vous pouvez *annuler une annulation* comme décrit ci dessus dans **Revenir à cette révision**.

#### Fusionner la révision avec...

Fusionner la (les) révision(s) sélectionnée(s) en une copie active séparée. Une boîte de dialogue de sélection de dossiers vous permet de choisir la copie active qui sera la cible de la fusion, mais attention : il n'y a par la suite aucune demande de confirmation ni aucune possibilité de faire une fusion de test. Il est judicieux de faire la fusion dans une copie active intacte afin de pouvoir annuler les modifications si cela ne fonctionne pas! Cette fonctionnalité se révèle très utile si vous désirez fusionner des révisions choisies d'une branche à une autre.

#### Extraire...

Fait une extraction propre de la révision sélectionnée. Cette fonctionnalité déclenche l'affichage d'une fenêtre permettant de confirmer l'URL, le numéro de la révision, et de sélectionner l'emplacement de l'extraction.

#### Exporter...

Exporter le fichier/répertoire sélectionné à la révision sélectionnée. Une fenêtre vous demande alors de confirmer l'URL, la révision, et de sélectionner le répertoire où faire exportation.

#### Modifier l'auteur/le commentaire de révision

Éditer le commentaire de révision ou l'auteur attaché à une précédente livraison. Lisez [Section 4.9.7, « Changer le commentaire et l'auteur »](#) pour découvrir comment cela fonctionne.

#### Montrer les propriétés de la révision

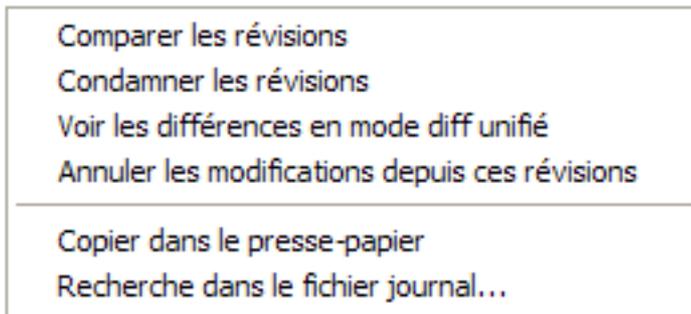
Voir ou éditer les propriétés d'une révision, outre le commentaire de révision et l'auteur. Voir [Section 4.9.7, « Changer le commentaire et l'auteur »](#).

#### Ajouter au presse-papier

Copier les commentaires de révisions sélectionnées dans le presse-papier. Pour chaque révision, la révision, l'auteur, la date, le commentaire et la liste des éléments modifiés seront alors copiés.

#### Rechercher les commentaires de révision...

Chercher les commentaires de révision correspondant au texte que vous saisissez. La recherche s'effectue dans les commentaires de révision que vous avez saisis ainsi que dans les résumés d'action créés par Subversion (présenté dans le panneau du bas). La recherche n'est pas sensible à la casse.



**Figure 4.17. Menu contextuel du panneau supérieur avec 2 révisions sélectionnées**

Si vous sélectionnez deux révisions en même temps (en utilisant le modificateur habituel **Ctrl**), le menu contextuel change et vous propose moins d'options :

**Comparer des révisions**

Comparer les deux révisions choisies en utilisant un outil de différenciation visuel. L'outil de différenciation par défaut est TortoiseMerge qui est fourni avec TortoiseSVN.

Si vous choisissez cette option pour un dossier, une nouvelle boîte de dialogue apparaît, qui liste les fichiers modifiés et vous offre des options de comparaison avancées. Pour en savoir plus sur la comparaison de révisions, reportez-vous [Section 4.10.3, « Comparer des répertoires »](#).

**Annoter les révisions**

Annoter les deux révisions et comparer les résultats en utilisant un outil de différenciation visuel. Lisez [Section 4.23.2, « Annoter les différences »](#) pour plus de détails.

**Voir les différences comme un diff unifié**

Voir les différences entre les deux révisions sélectionnées en tant que fichier de différences unifiées. Cela fonctionne pour les fichiers et les dossiers.

**Ajouter au presse-papier**

Copier les commentaires de révision dans le presse-papiers comme décrit ci-dessus.

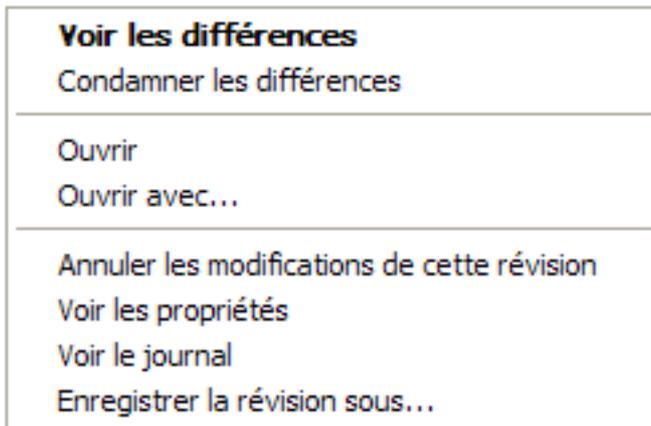
**Rechercher les commentaires de révision...**

Chercher les commentaires de révision comme décrit ci-dessus.

Si vous sélectionnez deux ou plusieurs révisions (en utilisant les modificateurs habituels **Ctrl** ou **Shift**), le menu contextuel inclura une entrée pour Annuler toutes les modifications qui ont été faites dans ces révisions. C'est la façon la plus facile d'annuler un groupe de révisions en une fois.

Vous pouvez également choisir de fusionner les révisions sélectionnées dans une autre copie de travail, comme décrit ci-dessus.

Si toutes les révisions sélectionnées ont le même auteur, vous pouvez éditer le champ auteur en une seule opération.



**Figure 4.18. Le panneau inférieur de la boîte de dialogue du Journal avec le menu contextuel**

Le panneau du bas de la boîte de dialogue Journal a aussi un menu contextuel qui vous permet de :

Voir les modifications

Show changes made in the selected revision for the selected file. This context menu is only available for files shown as *modified*.

Annoter les modifications

Annoter la révision sélectionnée et la révision précédente pour le fichier choisi et afficher les différences en utilisant un outil de différenciation visuel. Lisez [Section 4.23.2, « Annoter les différences »](#) pour plus de détails.

Voir comme un diff unifié

Montrer les modifications en tant que fichier de différences unifiées. Ce menu contextuel est seulement disponible pour les fichiers affichés comme *Modifiés*.

Ouvrir / Ouvrir avec...

Ouvrir le fichier sélectionné, avec le visualisateur par défaut pour ce type de fichier ou avec le programme de votre choix.

Annoter...

Ouvrir la fenêtre d'annotation, vous permettant d'annoter la révision sélectionnée.

Annuler les modification pour revenir à cette révision

Annuler les changements effectués au fichier sélectionné dans cette révision.

Voir les propriétés

Voir les propriétés Subversion pour l'élément sélectionné.

Voir le journal

Afficher le journal de révision pour le seul fichier choisi.

Récupérer les commentaires de fusion

Montrer les commentaires de révision pour le seul fichier sélectionné, y compris les modifications fusionnées. Plus de détail dans [Section 4.9.6, « Fonctionnalités de Suivi des Fusions »](#).

Sauvegarder la révision dans...

Sauvegarder la révision sélectionnée dans un fichier pour que vous ayez une version antérieure de ce fichier.



### Astuce

Vous avez peut être remarqué que nous faisons parfois référence à des modifications et parfois à des différences. Quelle différence y a-t-il ?

Subversion utilise des numéros de révision pour deux raisons. Une révision représente l'état d'un référentiel à un moment donné, mais elle peut aussi matérialiser la liste des modifications ayant mené à cette révision, i.e. « Fait dans r1234 » veut dire que l'implémentation de la fonctionnalité X a été faite via les modifications livrées dans la r1234. Pour préciser quel sens est utilisé, nous utilisons deux termes différents.

Si vous sélectionnez deux révisions N et M, le menu contextuel vous permettra de voir les *différences* entre ces deux révisions. Dans Subversion, cette opération correspond à la commande `diff -r M:N`.

Si vous sélectionnez une seule révision N, le menu contextuel vous proposera d'afficher les *modifications* faites dans cette révision. Dans le jargon de Subversion cela se traduit par `diff -r N-1:N` ou `diff -c N`.

Le panneau du bas de la boîte de dialogue du Journal de révision présente les fichiers modifiés dans toutes les révisions sélectionnées. Le menu contextuel propose donc toujours d'afficher les *modifications*.

#### 4.9.4. Obtenir plus de commentaires

Le boîte de dialogue du Journal ne montre pas toujours toutes les modifications faites pour plusieurs raisons :

- Pour un grand référentiel, il peut y avoir des centaines ou même des milliers de modifications et les parcourir toutes pourrait prendre longtemps. Normalement vous n'êtes intéressé que par les modifications les plus récentes. Par défaut, le nombre de commentaires parcourus est limité à 100, mais vous pouvez changer cette valeur dans TortoiseSVN → Configuration ([Section 4.30, « Configuration de TortoiseSVN »](#)),
- Quand la case **Arrêt à la copie/renommage** est cochée, Voir le journal s'arrêtera au moment où le fichier ou le dossier sélectionné ont été copiés d'ailleurs dans le référentiel. Cela peut être utile en regardant les branches (ou les étiquettes) puisque cela s'arrête à la racine de cette branche et donne une indication rapide des modifications faites dans cette branche seulement.

Normalement vous voudrez laisser cette option décochée. TortoiseSVN se souvient de l'état de la case à cocher, donc il respectera votre préférence.

Quand la boîte de dialogue Voir le journal est appelée depuis la boîte de dialogue Fusionner, la case est toujours cochée par défaut. La raison en est que la fusion concerne le plus souvent des modifications sur des branches et que revenir au-delà de la racine de la branche n'a alors aucun sens.

Notez que Subversion implémente actuellement le renommage comme une paire copie/suppression, donc renommer un fichier ou un dossier causera aussi l'arrêt de l'affichage du journal si cette option est cochée.

Si vous voulez voir plus de commentaires de révision, cliquez sur le bouton 100 suivants pour récupérer les 100 commentaires suivants. Vous pouvez répéter cela autant de fois que nécessaire.

À côté de ce bouton il y a un bouton multi-fonctions qui se souvient de la dernière option pour laquelle vous l'avez utilisé. Cliquez sur la flèche pour voir les autres options offertes.

Utilisez **Afficher la plage ...** si vous voulez consulter une plage spécifique de révisions. Une boîte de dialogue vous demandera alors d'entrer les révisions de début et de fin.

Utilisez **Afficher tout** si vous voulez voir *tous* les commentaires depuis HEAD jusqu'à la révision 1.

#### 4.9.5. Révision de la Copie de Travail Courante

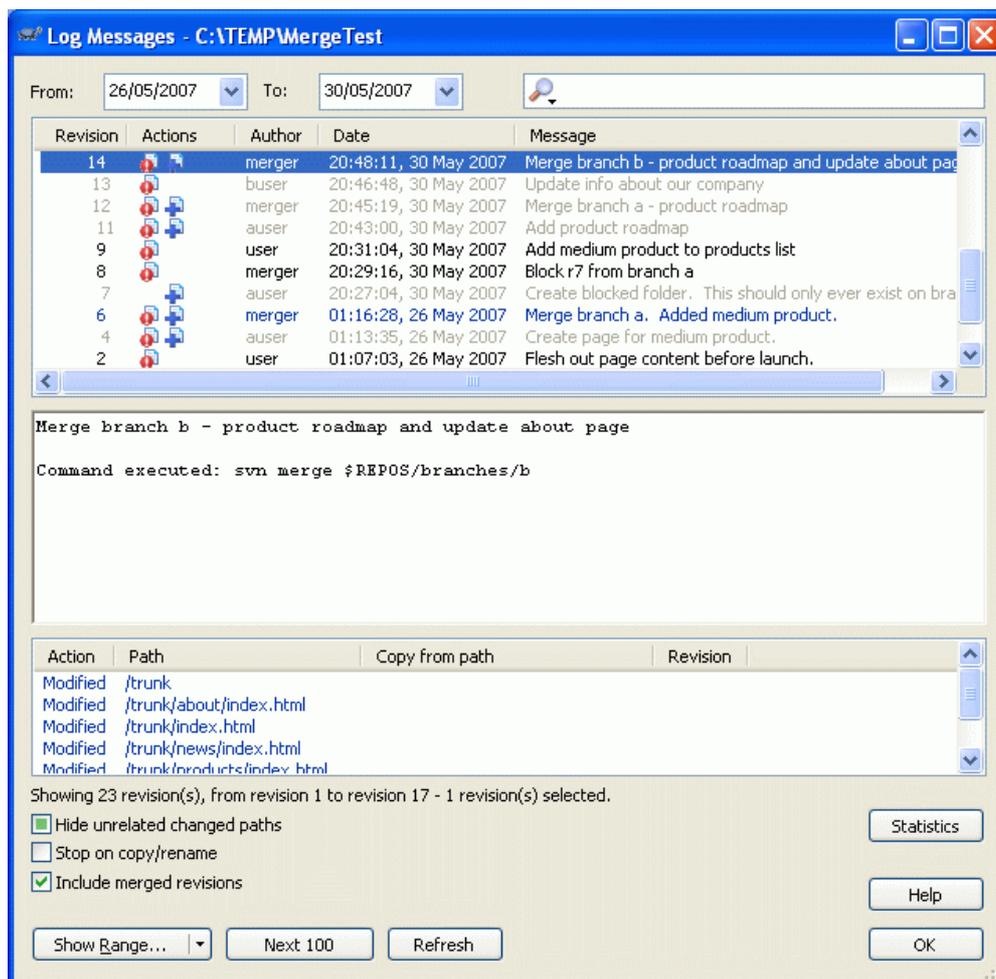
La fenêtre de commentaires affichant les commentaires depuis HEAD et non de la révision de la copie de travail, il arrive souvent que des commentaires de révisions non encore importées dans votre copie de

travail soient affichés. Pour clarifier, le message de livraison correspondant à la révision de votre copie de travail est affiché en gras.

When you show the log for a folder the revision highlighted is the highest revision found anywhere within that folder, which requires a crawl of the working copy. This can be a slow operation for large working copies, and the log messages are not displayed until the crawl completes. If you want to disable or limit this feature you need to set a registry key HKCU\Software\TortoiseSVN\RecursiveLogRev as described in [Section 4.30.10, « Réglages dans le registre »](#).

#### 4.9.6. Fonctionnalités de Suivi des Fusions

Le logiciel Subversion 1.5 et ultérieur



**Figure 4.19. La Fenêtre du Journal de révision Montrant les Révisions Fusionnées**

Si vous voulez voir quelles révisions ont été fusionnées dans un commit donné, utilisez la case à cocher **Inclure les révisions fusionnées**. Les commentaires de révision seront à nouveau affichés, mais ils incluront également les commentaires des révisions fusionnées. Les révisions fusionnées sont affichées en gris car elles représentent des modifications effectuées dans une autre branche de l'arborescence.

Bien sûr, fusionner n'est jamais une mince affaire ! Pendant le développement des fonctionnalités dans une branche il y aura sûrement des fusions à faire avec la tête pour que la branche reste synchronisée avec la version de tête. Donc l'historique des fusions de la branche inclura une autre couche d'historique des fusions. Ces différentes couches sont affichées dans la fenêtre du journal avec une indentation différente.

### 4.9.7. Changer le commentaire et l'auteur

Les propriétés des révisions sont complètement différentes des propriétés de Subversion de chaque élément. Les Revprops sont des éléments descriptifs qui sont associés à un numéro de révision spécifique dans le référentiel, comme les commentaires de révision, date de livraison et nom de la personne ayant livré (auteur).

Parfois vous pourriez vouloir changer un commentaire que vous avez saisi, peut-être parce qu'il y a une faute d'orthographe dedans ou parce que vous voulez améliorer le message ou le changer pour d'autres raisons. Ou vous voulez changer l'auteur de la livraison parce que vous avez oublié de mettre en place l'authentification ou...

Subversion vous laisse changer les propriétés d'une révision aussi souvent que vous le souhaitez. Mais puisque de tels changements ne peuvent pas être annulés (ces changements ne sont pas versionnés) cette fonctionnalité est désactivée par défaut. Pour faire que cela fonctionne, vous devez mettre en place un hook pre-revprop-change. Veuillez vous référer au chapitre sur *Hook Scripts* [<http://svnbook.red-bean.com/en/1.1/ch05s02.html#svn-ch-5-sect-2.1>] dans le manuel de Subversion pour plus de détails sur la façon de le faire. Lisez [Section 3.3, « Scripts de hook côté serveur »](#) pour plus de détails sur l'implémentation de hooks sur une machine Windows.

Une fois que vous avez mis en place votre serveur avec les hooks requis, vous pouvez changer l'auteur et le commentaire (ou toute autre revprop) de n'importe quelle révision, en utilisant le menu contextuel du panneau supérieur de la boîte de dialogue du Journal. Vous pouvez également éditer un commentaire de révision par l'intermédiaire du menu contextuel du panneau du milieu.



#### Avertissement

Parce que les propriétés de révision de Subversion ne sont pas versionnées, modifier une telle propriété (par exemple, la propriété du commentaire de livraison `svn:log`) écrasera la valeur précédente de cette propriété *pour toujours*.

### 4.9.8. Filtrer les commentaires

Si vous voulez limiter les commentaires pour afficher seulement ceux qui vous intéressent, plutôt que de faire défiler une liste de centaines de commentaires, vous pouvez utiliser les commandes de filtre en haut de la boîte de dialogue du Journal. Les dates de début et de fin vous permettent de limiter les résultats à une plage de dates connue. La boîte de recherche vous permet de n'afficher que les messages qui contiennent une expression particulière.

Click on the search icon to select which information you want to search in, and to choose *regex* mode. Normally you will only need a simple text search, but if you need to more flexible search terms, you can use regular expressions. If you hover the mouse over the box, a tooltip will give hints on how to use the regex functions. You can also find online documentation and a tutorial at <http://www.regular-expressions.info/>. The filter works by checking whether your filter string matches the log entries, and then only those entries which *match* the filter string are shown.

Pour faire en sorte que tous les commentaires *ne vérifiant pas* le filtre s'affichent, commencez l'expression régulière par un point d'exclamation (!). Par exemple, la chaîne `!username` ne montrera que les entrées n'ayant pas été livrées par `username`.

Notez que ces filtres agissent sur les commentaires déjà récupérés. Ils ne provoquent pas de téléchargement de commentaires du référentiel.

You can also filter the path names in the bottom pane using the **Hide unrelated changed paths** checkbox. Related paths are those which contain the path used to display the log. If you fetch the log for a folder, that means anything in that folder or below it. For a file it means just that one file. The checkbox is tristate: you can show all paths, grey out the unrelated ones, or hide the unrelated paths completely.

Parfois, vos habitudes de travail nécessiteront que les commentaires de révision aient un format particulier, ce qui signifie que le texte résumant les modifications ne sera pas visible dans le panneau supérieur. La propriété `svn:logsummary` peut être utilisée pour afficher dans le panneau supérieur une partie du commentaire de révision. Lisez [Section 4.17.2, « Propriétés du projet TortoiseSVN »](#) pour savoir comment utiliser cette propriété.



## Pas de Formattage des Commentaires depuis l'Explorateur de Référentiel

Le formattage dépendant des propriétés d'accès de subversion, vous ne verrez le résultat que si vous utilisez une copie de travail exportée. Afficher les propriétés à distance étant une opération lente, vous ne pourrez pas l'effectuer depuis l'explorateur de référentiel.

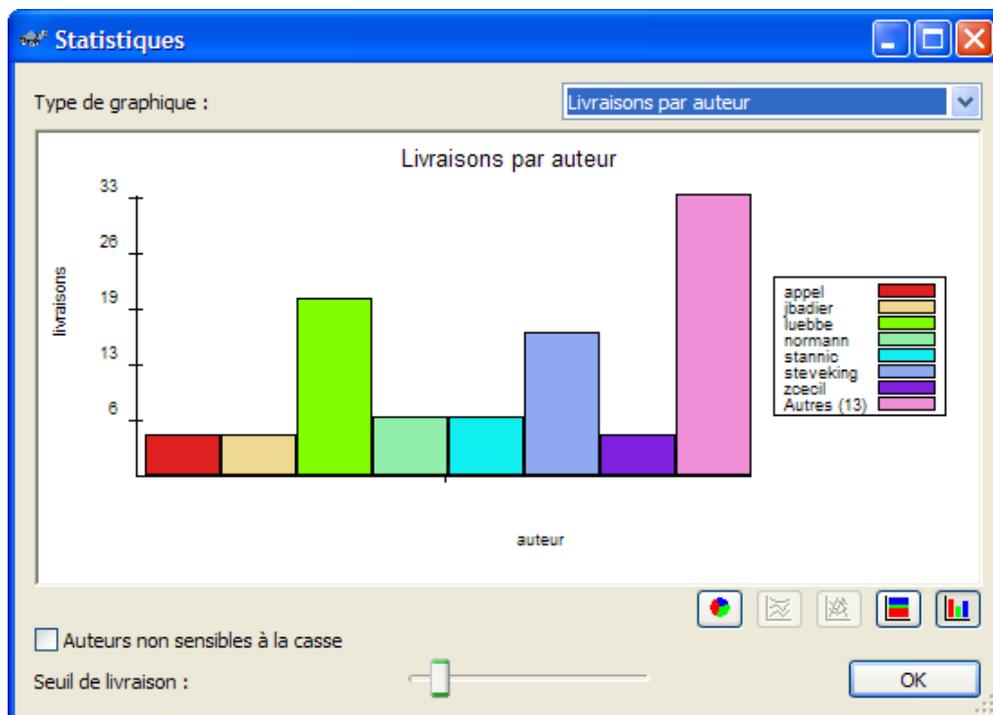
### 4.9.9. Informations statistiques

Le bouton **Statistiques** fait apparaître une boîte montrant des informations intéressantes sur les révisions affichées dans la boîte de dialogue de Journal. Elle montre combien d'auteurs ont travaillé, combien de livraisons ils ont fait, la progression par semaine et beaucoup plus. Maintenant vous pouvez voir d'un coup d'oeil qui a travaillé le plus dur et qui se relâche ;-)

#### 4.9.9.1. Page des statistiques

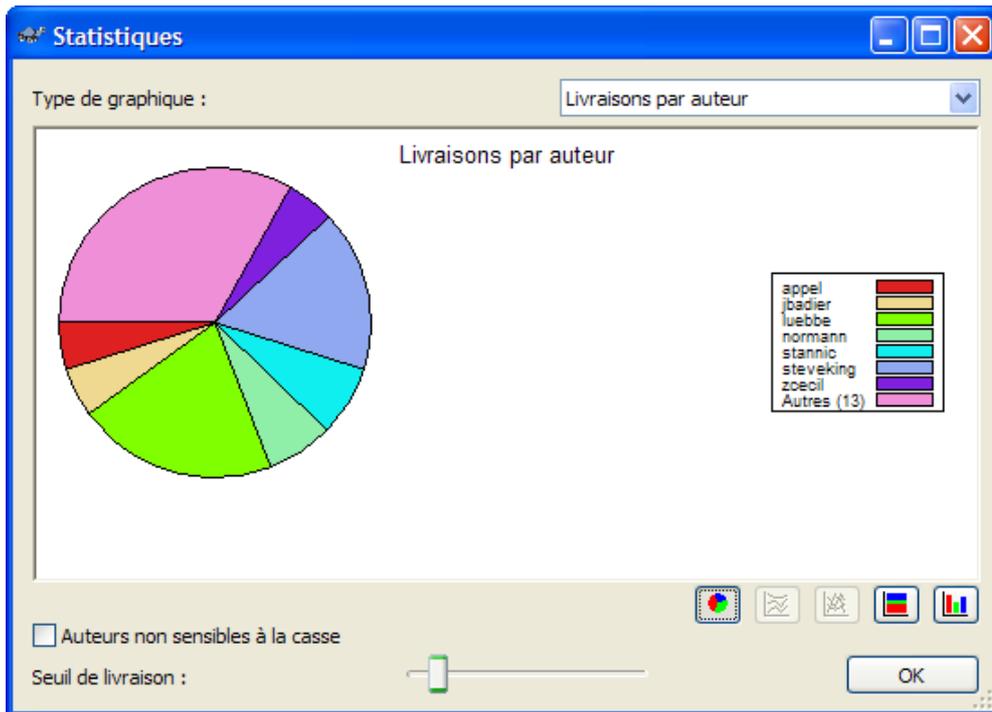
Cette page vous donne tous les nombres auxquels vous pouvez penser, en particulier la période et le nombre de révisions couvertes et quelques valeurs min/max/moyennes.

#### 4.9.9.2. Page de livraisons par auteur



**Figure 4.20. Histogramme de livraisons par auteur**

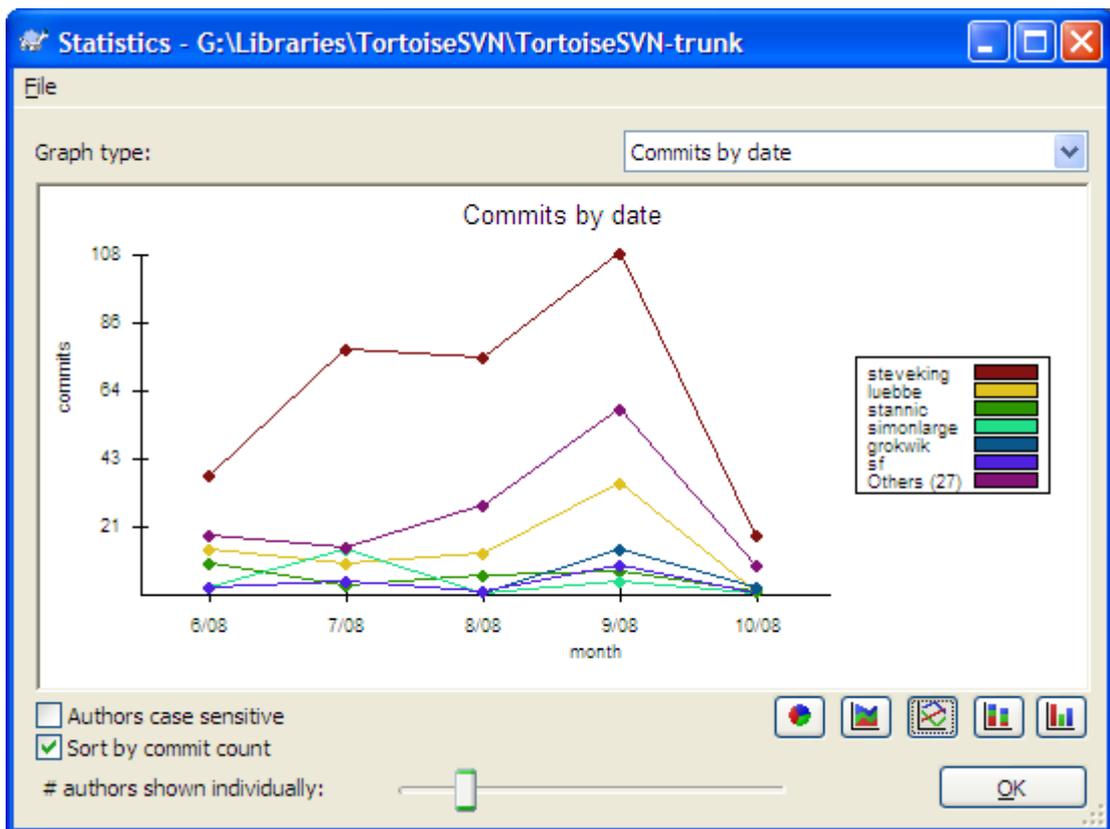
Ce graphique vous montre quels auteurs ont été actifs sur le projet par un simple histogramme, un histogramme empilé ou un camembert.



**Figure 4.21. Camembert de livraisons par auteur**

Quand il y a quelques auteurs majeurs et beaucoup de contributeurs mineurs, le nombre de segments minuscules peut rendre le graphique plus difficile à lire. Le slider en bas vous permet de définir un seuil (en pourcentage du total des livraisons) en dessous duquel l'activité est groupée dans une catégorie *Autres*.

### 4.9.9.3. Page de livraisons par date



**Figure 4.22. Graphique de livraisons par date**

Cette page vous donne une représentation graphique de l'activité du projet en termes de nombre de livraisons *et* d'auteurs. Cela donne une certaine idée de quand un projet est actif et de qui travaillait à quel moment.

Quand il y a plusieurs auteurs, vous obtiendrez beaucoup de lignes sur le graphique. Il y a deux vues disponibles ici : *normale*, où l'activité de chaque auteur est relative à la ligne de base et *empilée*, où l'activité de chaque auteur est relative à la ligne d'en dessous. La dernière option évite les lignes qui se traversent, ce qui peut rendre le graphique plus facile à lire, mais la production d'un auteur moins facile à voir.

Par défaut l'analyse est sensible à la casse donc les utilisateurs PeterEgan et PeteRegan sont considérés comme des auteurs différents. Cependant, dans de nombreux cas les noms des utilisateurs ne sont pas sensibles à la casse et sont parfois saisis de façon variable, donc vous pourriez vouloir que DavidMorgan et davidmorgan soient considérés comme la même personne. Utilisez la case à cocher Auteurs non sensibles à la casse pour contrôler la manière dont cela est traité.

Notez que les statistiques couvrent la même période que la boîte de dialogue du Journal. Si elle ne montre qu'une révision alors les statistiques ne vous diront pas grand chose.

### 4.9.10. Mode hors ligne

Si le serveur est inaccessible, et que vous avez activé la mise en cache des commentaires de révision, vous pouvez utiliser la boîte de dialogue du Journal et le graphique des révisions en mode hors ligne. Ces outils utiliseront alors les données mises en cache, ce qui vous permet de continuer à travailler même si les informations sont incomplètes ou ne sont pas à jour.

Vous avez ici trois choix :

#### Mode hors ligne

Fini l'opération courante en mode hors ligne, mais réessaie avec le référentiel à la prochaine demande d'informations.

#### Constamment hors ligne

Reste en mode hors ligne jusqu'à la prochaine demande explicite de vérification. Voir [Section 4.9.11](#), « [Refraîchissement de l'affichage](#) ».

#### Annuler

Si vous ne voulez pas continuer l'opération en cours avec le risque d'avoir des données bancales, annulez.

La case à cocher Définir par défaut empêche cette fenêtre de réapparaître et retient toujours l'option suivante choisie. Vous pouvez toujours modifier (ou supprimer) cette valeur par défaut plus tard dans le menu TortoiseSVN → Paramètres.

### 4.9.11. Refraîchissement de l'affichage

Si vous voulez vérifier si de nouveaux commentaires sont arrivés sur le serveur, vous pouvez simplement rafraîchir l'interface en appuyant sur **F5**. Si vous utilisez un cache pour les commentaires (ce qui est le comportement par défaut), cette vérification se fera et seuls les nouveaux commentaires seront rappatriés. Si le cache des commentaires était en mode hors ligne, il tentera de se reconnecter.

Si vous utilisez le cache du journal et si vous pensez que le contenu du message ou son auteur ont pu changer, vous pouvez utiliser **Shift+F5** ou **Ctrl-F5** pour récupérer à nouveau les messages affichés à partir du serveur et mettre à jour le cache journal. Notez que ceci affecte uniquement les messages affichés à cet instant et que ceci n'invalide pas le cache complet pour ce référentiel.

## 4.10. Voir les différences

Une des exigences les plus courantes dans le développement de projet est de voir ce qui a changé. Vous pourriez vouloir regarder les différences entre deux révisions du même fichier, ou les différences entre deux fichiers séparés. TortoiseSVN fournit un outil intégré appelé TortoiseMerge pour voir les différences dans les fichiers texte. Pour voir les différences dans les fichiers image, TortoiseSVN a aussi un outil appelé TortoiseIDiff. Bien sûr, vous pouvez utiliser votre propre programme de comparaison favori si vous le souhaitez.

### 4.10.1. Différences de fichier

#### Changements locaux

Si vous voulez voir quels changements *vous* avez fait dans votre copie de travail, utilisez juste le menu contextuel de l'explorateur et choisissez TortoiseSVN → Voir les différences.

#### Comparaison avec une autre branche/étiquette

Si vous voulez voir ce qui a changé sur le tronc (si vous travaillez sur une branche) ou sur une branche spécifique (si vous travaillez sur le tronc), vous pouvez utiliser le menu contextuel de l'explorateur. Maintenez juste la touche **Maj** tandis que vous faites un clic droit sur le fichier. Sélectionnez alors TortoiseSVN → Diff avec l'URL. Dans la boîte de dialogue suivante, spécifiez l'URL dans le référentiel avec laquelle vous voulez comparer votre fichier local.

Vous pouvez aussi utiliser l'explorateur de référentiel et sélectionner deux arborescences à comparer, deux étiquettes peut-être, ou une branche/étiquette et le tronc. Le menu contextuel vous permet là de les comparer en utilisant Comparer les révisions. Plus d'informations dans [Section 4.10.3](#), « [Comparer des répertoires](#) ».

#### Comparaison avec une révision précédente

Si vous voulez voir les différences entre une révision particulière et votre copie de travail, utilisez la boîte de dialogue de Journal de révision, sélectionnez la révision qui vous intéresse, puis choisissez Comparaison avec la copie de travail dans le menu contextuel.

Si vous voulez voir la différence entre la dernière révision et votre copie active, en supposant que la copie active n'ait pas été modifiée, faites simplement un clic droit sur le fichier. Ensuite sélectionnez TortoiseSVN → Différences avec la version précédente. Ceci calculera les différences entre la révision avant la date de dernière Livraison (comme sauvegardé dans votre copie active) et la BASE active. Ceci vous montre la dernière modification effectuée sur ce fichier et qui l'a amené à l'état dans lequel vous le voyez maintenant dans votre copie active. Ceci ne va pas montrer les modifications ultérieures à votre copie active.

#### Comparaison entre deux révisions précédentes

Si vous voulez voir les différences entre deux révisions déjà livrées, utilisez la boîte de dialogue de Journal de révision et sélectionnez les deux révisions que vous voulez comparer (en utilisant le modificateur habituel **Ctrl**). Puis choisissez Comparer les révisions à partir du menu contextuel.

Si vous avez fait ceci depuis le journal sur un répertoire, une boîte de dialogue Comparer les révisions apparaît, affichant la liste des fichiers dans ce dossier. Plus d'informations dans [Section 4.10.3, « Comparer des répertoires »](#).

#### Tous les changements faits dans une livraison

Si vous voulez voir les changements faits à tous les fichiers dans une révision particulière en une vue, vous pouvez utiliser la sortie en mode Diff-Unifié (format de patch GNU). Cela montre seulement les différences replacées dans leur contexte. Il est plus difficile à lire qu'une comparaison de fichier visuelle, mais montrera tous les changements ensemble. À partir de la boîte de dialogue du Journal de Révision sélectionnez la révision qui vous intéresse, puis choisissez Voir les différences en mode diff unifié à partir du menu contextuel.

#### Comparaison entre fichiers

Si vous voulez voir les différences entre deux fichiers différents, vous pouvez le faire directement dans l'explorateur en sélectionnant les deux fichiers (en utilisant le modificateur habituel **Ctrl**). Alors à partir du menu contextuel de l'explorateur, sélectionnez TortoiseSVN → Voir les différences.

#### Différence entre fichiers/répertoires de la CdT et une URL

Si vous voulez voir les différences entre un fichier de votre copie de travail, et un fichier dans n'importe quel référentiel Subversion, vous pouvez le faire directement dans l'explorateur en sélectionnant le fichier puis en maintenant enfoncé la touche **Shift** en faisant un clic droit pour obtenir le menu contextuel. Sélectionnez TortoiseSVN → Diff avec l'URL. Vous pouvez faire la même chose pour un répertoire de la copie de travail. TortoiseMerge vous montre ces différences de la même manière qu'un patch - une liste des fichiers modifiés que vous pouvez voir un par un.

#### Comparaison avec les informations d'annotation

Si vous voulez ne pas voir que les différences mais aussi l'auteur, la révision et la date des modifications effectuées, vous pouvez combiner les rapports de différenciation et d'annotation au sein de la boîte de dialogue du journal de révision. Lisez [Section 4.23.2, « Annoter les différences »](#) pour plus de détails.

#### Comparaison entre répertoires

Les outils intégrés fournis avec TortoiseSVN ne supportent pas l'observation de différences entre les hiérarchies de répertoire. Mais si vous avez un outil externe qui supporte vraiment cette fonctionnalité, vous pouvez l'utiliser à la place. Dans [Section 4.10.5, « Outils de différenciation/fusion externes »](#) nous vous parlerons de quelques outils que nous avons utilisés.

Si vous avez fourni un outil de diff alternatif, vous pouvez utiliser la touche **Shift** quand vous sélectionnez la commande Diff pour utiliser l'outil alternatif. Lisez [Section 4.30.5, « Réglages des programmes externes »](#) pour savoir comment configurer les autres outils de diff.

## 4.10.2. Options de fins de ligne et d'espacement

Parfois dans la vie d'un projet il se peut que vous modifiiez les fins de ligne de CRLF en LF, ou que vous changiez l'alinéa d'une section. Malheureusement ceci va marquer un grand nombre de lignes comme étant modifiées, bien qu'il n'y ait pas de variation dans la signification du code. Ces options ici vont aider à gérer ces modifications lorsqu'il s'agira de comparer et d'appliquer des différences. Vous pourrez

voir ces réglages dans les boîtes de dialogue Fusionner et Annoter, tout comme dans les paramètres de FusionTortoise.

Ignorer les caractères de fins de ligne exclue les modifications seulement dues au style des caractères de fin de ligne.

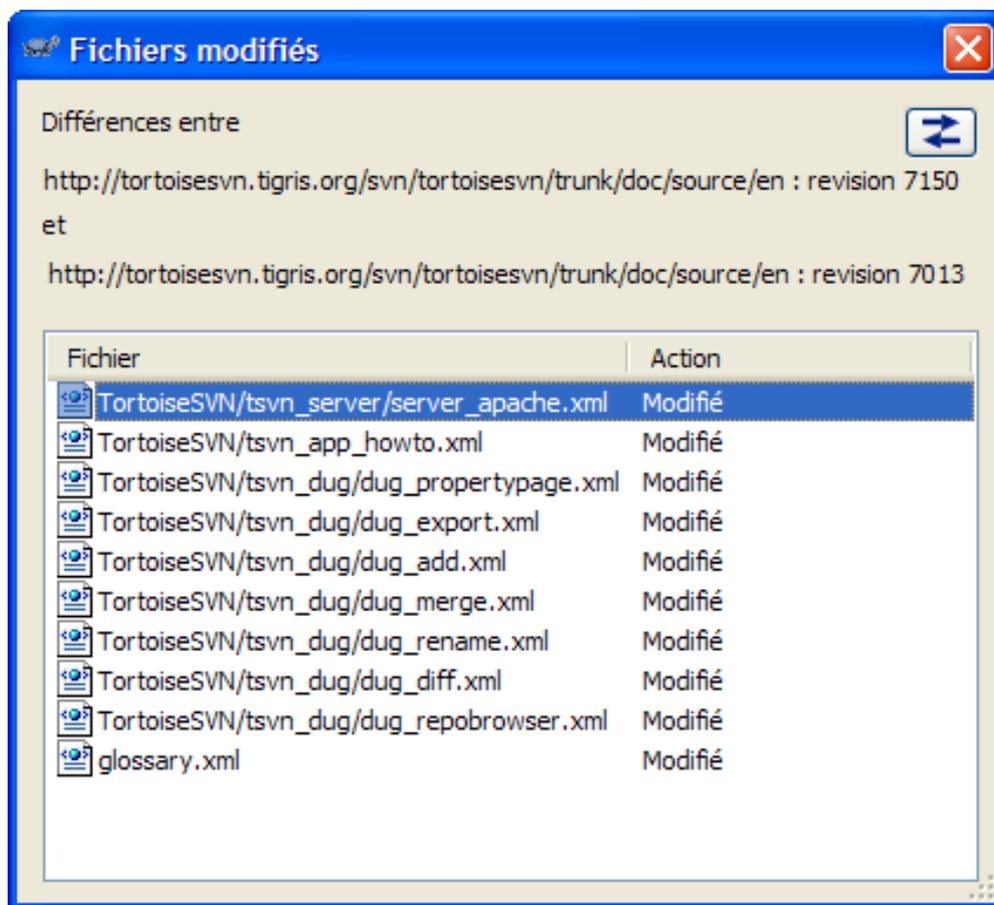
Compare les caractères d'espacement inclue les modifications d'indentation et d'espaces à la liste des lignes ajoutées/supprimées.

Ignore les modifications des caractères d'espacement exclue les modifications étant seulement dues au style ou à la quantité de caractères d'espacement, i.e. changement dans l'indentation ou remplacement des tabulations par des espaces. Ajouter des espaces là où il n'y en avait pas, ou tous les retirer est toujours affiché comme étant une modification.

Ignorer tous les caractères d'espacement exclue toutes les modifications dues aux caractères d'espacement.

Naturellement, chaque ligne modifiée est toujours incluse dans le diff.

### 4.10.3. Comparer des répertoires



**Figure 4.23. La boîte de dialogue Comparer les révisions**

Quand vous sélectionnez deux arborescences dans l'explorateur de référentiel ou quand vous sélectionnez deux révisions d'un répertoire dans la boîte de dialogue du journal, vous pouvez Menu contextuel → Comparer les révisions.

Cette boîte de dialogue affiche une liste des fichiers modifiés et vous permet de les comparer ou de les annoter individuellement en utilisant le menu contextuel.

Vous pouvez exporter un *arbre des modifications*, qui est utile si vous avez besoin d'envoyer la structure de l'arbre de votre projet à un tiers, contenant uniquement les fichiers qui ont été modifiés. Cette opération fonctionne sur les fichiers sélectionnés uniquement, ainsi vous devez sélectionner les fichiers qui vous intéressent - ce qui signifie souvent tous les fichiers - puis ensuite sélectionner Menu Contextuel → Exporter la sélection vers.... Vous serez invités à choisir un emplacement où sauvegarder l'arbre des modifications.

Vous pouvez également exporter la *liste* des fichiers modifiés dans un fichier texte en utilisant le menu contextuel → Enregistrer la liste de fichiers sélectionnés vers....

Si vous voulez exporter la listes des fichiers *et* aussi les actions (modifié, ajouté, supprimé), vous pouvez le faire en utilisant le menu contextuel → Copier la sélection dans le presse-papier.

Le bouton en haut vous permet de modifier le sens de la comparaison. Vous pouvez afficher les changements nécessaires pour arriver de A vers B, ou si vous préférez, de B vers A.

Les boutons avec les numéros de révision peuvent être utilisés pour passer à un éventail de révisions différent. Lorsque vous changez d'éventail, la liste des éléments qui diffèrent entre les deux révisions sera mise à jour automatiquement.

Si la liste des noms de fichier est très longue, vous pouvez utiliser le champ de recherche pour réduire sa taille. Remarquez qu'une simple recherche texte est utilisée, donc si vous voulez restreindre la liste aux fichiers C vous devrez rechercher `.c` et non `*.c`.

#### 4.10.4. Comparaison des images en utilisant TortoiseIDiff

Il y a beaucoup d'outils disponibles pour comparer des fichiers texte, incluant notre propre TortoiseMerge, mais souvent, nous voulons également voir les modifications effectuées sur un fichier image. C'est pourquoi nous avons créé TortoiseIDiff.

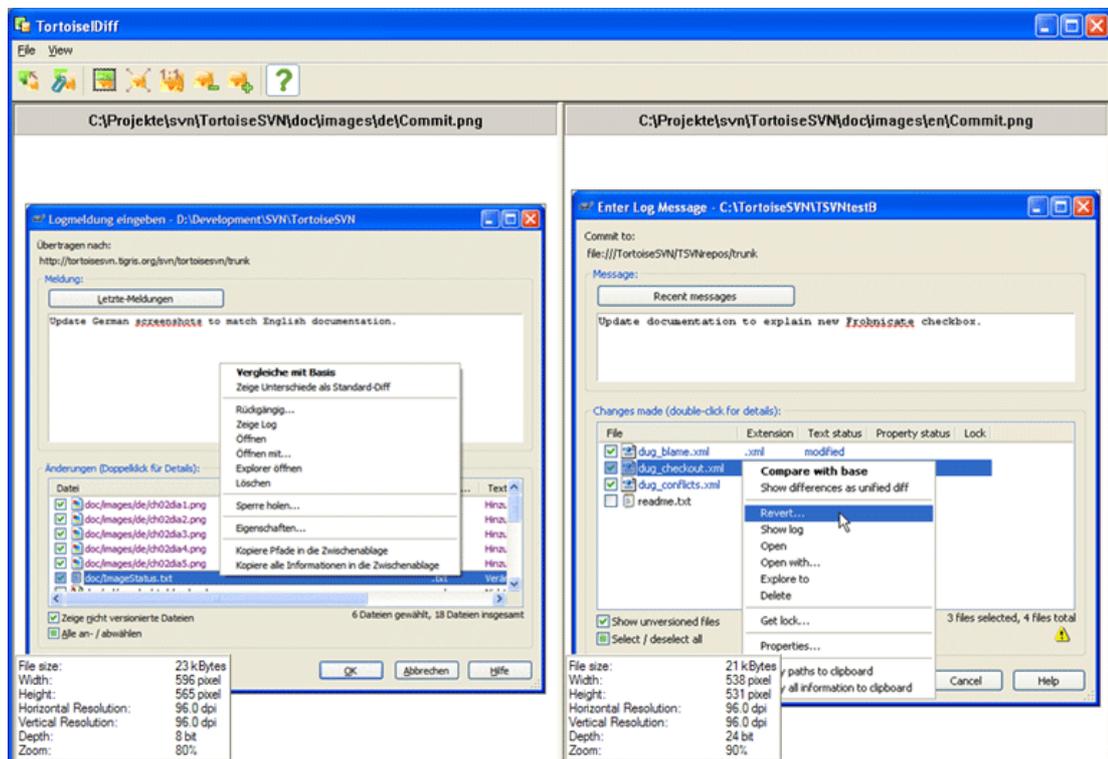


Figure 4.24. Le visualiseur de différences d'images

TortoiseSVN → Voir les différences pour n'importe quel format d'image commun démarrera TortoiseIDiff pour afficher les différences d'image. Par défaut les images sont affichées côte à côte mais vous pouvez utiliser le menu Affichage ou la barre d'outils pour basculer vers une vue haut-bas à la place, ou si vous préférez, vous pouvez superposer les images et utiliser une visionneuse.

Naturellement vous pouvez aussi faire un zoom avant, arrière tout en vous déplaçant sur l'image. Vous pouvez également effectuer un mouvement de l'image à gauche simplement en la faisant glisser. Si vous sélectionnez l'option **Lier les images ensemble**, alors les contrôles de déplacement (barre de défilement, molette) des deux images sont reliés.

Dans l'image, une boîte d'informations affiche les détails concernant le fichier image, comme la taille en pixels, la résolution et la profondeur des couleurs. Si cette boîte se trouve dans le chemin, utilisez **Affichage → Informations de l'image** pour la masquer. Vous pouvez obtenir la même information dans une info-bulle lorsque vous passez la souris sur la barre de titre de l'image.

Lorsque les images sont recouvertes, l'intensité relative des images (transparence) est contrôlée grâce à un ascenseur situé sur la gauche. Vous pouvez directement cliquer n'importe où sur l'ascenseur, ou faire glisser le curseur pour modifier interactivement sa valeur. **Ctrl+Shift-Roulette** pour modifier la transparence.

Le bouton au dessus de l'ascenseur permet de passer de 0% à 100% de transparence, et si vous double-cliquez dessus, la transparence change automatiquement chaque seconde jusqu'à ce que vous cliquiez de nouveau. Ce qui peut être utile lorsque vous cherchez beaucoup de petites modifications.

Parfois vous voulez voir une différence plutôt qu'un mélange. Vous pourriez avoir les fichiers d'images de deux révisions d'un circuit imprimé et dont vous souhaitez voir quelles pistes ont changées. Si vous désactivez le mode alpha blend, la différence sera présentée comme un *XOR* de la valeur des pixels de couleur. Les zones inchangées seront de couleur blanche et les changements seront colorés.

#### 4.10.5. Outils de différenciation/fusion externes

Si les outils que nous fournissons ne font pas ce dont vous avez besoin, essayez un des nombreux programmes open-source ou commerciaux disponibles. Chacun a ses préférences et cette liste n'est en aucun cas exhaustive, mais en voici quelques-uns intéressants :

##### WinMerge

*WinMerge* [<http://winmerge.sourceforge.net/>] est un grand outil de comparaison open-source qui peut aussi manipuler les répertoires.

##### Perforce Merge

Perforce est un RCS commercial, mais vous pouvez télécharger l'outil de différenciation/fusion gratuitement . Obtenez plus d'informations sur *Perforce* [<http://www.perforce.com/perforce/products/merge.html>].

##### KDiff3

KDiff3 est un outil de comparaison gratuit qui peut aussi manipuler des répertoires. Vous pouvez le télécharger *ici* [<http://kdiff3.sf.net/>].

##### ExamDiff

ExamDiff Standard est un logiciel gratuit. Il peut manipuler les fichiers, mais pas les répertoires. ExamDiff Pro est shareware et ajoute un certain nombre de goodies incluant la comparaison de répertoires et des possibilités d'édition. Dans les deux, les versions 3.2 et supérieures peuvent gérer l'unicode. Vous pouvez les télécharger chez *PrestoSoft* [<http://www.prestosoft.com/>].

##### Beyond Compare

Semblable à ExamDiff Pro, c'est un outil de différenciation shareware excellent qui peut manipuler les comparaisons de répertoires et l'unicode. Téléchargez-le chez *Scooter Software* [<http://www.scootersoftware.com/>].

#### Araxis Merge

Araxis Merge est un outil commercial utile pour comparer et fusionner tant les fichiers que les dossiers. Il fait la comparaison à trois voies pour les fusions et a des liens de synchronisation à utiliser si vous avez changé l'ordre des fonctions. [Araxis](http://www.araxis.com/merge/index.html) [http://www.araxis.com/merge/index.html].

#### SciTE

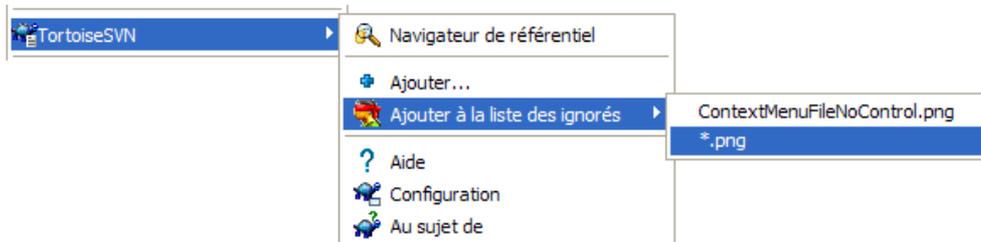
Cet éditeur de texte inclut la coloration syntaxique pour les diffs unifié, les rendant beaucoup plus faciles à lire. Téléchargez-le chez [Scintilla](http://www.scintilla.org/SciTEDownload.html) [http://www.scintilla.org/SciTEDownload.html].

#### Notepad2

Notepad2 est conçu comme un remplaçant pour le programme standard de Bloc-notes Windows et est basé sur le contrôle d'édition open-source de Scintilla. En plus d'être bon pour voir les diffs unifiés, il est beaucoup mieux que le bloc-notes Windows pour la plupart des travaux. Téléchargez-le gratuitement [ici](http://www.flos-freeware.ch/notepad2.html) [http://www.flos-freeware.ch/notepad2.html].

Lisez [Section 4.30.5, « Réglages des programmes externes »](#) pour des informations sur la façon de configurer TortoiseSVN pour utiliser ces outils.

## 4.11. Ajouter de nouveaux fichiers et répertoires



**Figure 4.25. Menu contextuel de l'explorateur pour les fichiers non versionnés**

Si vous avez créé de nouveaux fichiers et/ou de nouveaux répertoires pendant votre processus de développement alors vous devez aussi les ajouter au contrôle de source. Sélectionnez les fichiers et/ou les répertoires et utilisez TortoiseSVN → Ajouter.

Après que vous ayez ajouté les fichiers/répertoires au contrôle de source, le fichier apparaît avec une icône de recouvrement ajouté qui veut dire que vous devez d'abord livrer votre copie de travail pour rendre ces fichiers/répertoires disponibles aux autres développeurs. L'ajout d'un fichier/répertoire n'affecte *pas* le référentiel !



### Plusieurs ajouts

Vous pouvez aussi utiliser la commande Ajouter sur des dossier déjà versionnés. Dans ce cas, la boîte de dialogue ajouter vous montrera tous les fichiers non versionnés à l'intérieur de ce dossier versionné. C'est utile si vous avez beaucoup de nouveaux fichiers et que vous avez besoin de les ajouter en une fois.

Pour ajouter des fichiers de l'extérieur de votre copie de travail, vous pouvez utiliser le glisser-déplacer :

1. sélectionnez les fichiers que vous voulez ajouter
2. glissez-déposer avec le bouton droit ces fichiers dans le nouvel emplacement à l'intérieur de la copie de travail
3. relâchez le bouton droit de la souris
4. sélectionnez Menu contextuel → SVN Ajouter les fichiers à cette CdT. Les fichiers seront alors copiés dans la copie de travail et ajoutés au contrôle de version.

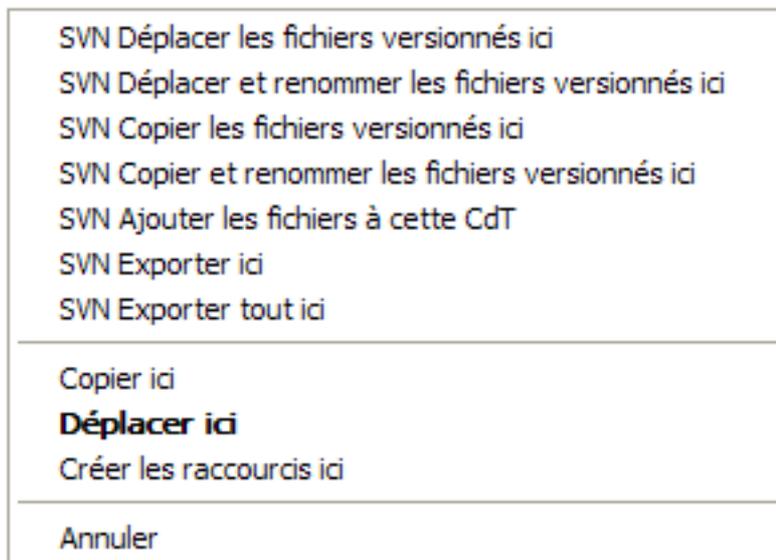
Vous pouvez également ajouter des fichiers dans la copie de travail en les faisant simplement glisser dans la fenêtre de livraison.

Si vous ajoutez un élément par erreur, vous pouvez annuler l'opération avant de livrer en utilisant la commande TortoiseSVN → Annuler l'ajout....

## 4.12. Copier/Déplacer/Renommer des Fichiers et des Dossiers

Il arrive souvent que les fichiers dont vous avez besoin soient dans une autre projet du référentiel, et que vous vouliez simplement en récupérer une copie. Vous pourriez simplement les copier puis les ajouter comme décrit ci dessus, mais vous perdriez ainsi tout l'historique. Et si plus tard vous corrigez un bogue dans les fichiers originaux, vous ne pouvez que fusionner le correctif.

Le moyen le plus simple de copier des fichiers et/ou des répertoires au sein de la copie de travail est d'utiliser le menu contextuel clique droit. Lorsque vous glissez/déposez un fichier/répertoire en utilisant le bouton droit de la souris d'une copie de travail vers une autre, voire à l'intérieur d'un même répertoire, un menu contextuel s'affiche quand vous relâchez le bouton de la souris.



**Figure 4.26. Menu pour un répertoire sous contrôle de version lors d'un glisser-déplacer avec le bouton droit**

Vous pouvez à présent copier du contenu déjà versionné à un autre endroit, et éventuellement le renommer d'une même action.

Vous pouvez également copier ou déplacer des fichiers versionnés à l'intérieur de la copie de travail, ou entre deux copies de travail, en utilisant le maintenant familier copier/coller. Utilisez Copier ou Couper pour copier un ou plusieurs éléments versionnés dans le presse papier. Si celui ci contient du contenu versionné, vous pouvez alors utiliser TortoiseSVN → Coller (note: PAS l'action Windows standard Coller) pour copier ou déplacer ces éléments au nouvel endroit dans la copie de travail.

Vous pouvez copier des fichiers ou des répertoires depuis votre copie de travail vers un autre endroit du référentiel en utilisant TortoiseSVN → Branche/Tag. Lisez [Section 4.19.1, « Créer une branche ou une étiquette »](#) pour plus d'informations.

Vous pouvez localiser l'ancienne version d'un fichier ou d'un répertoire dans la fenêtre de commentaires et la copier directement à un autre endroit du référentiel en utilisant Menu Contextuel → Créer

une branche/étiquette depuis la révision. Lisez [Section 4.9.3, « Obtenir des informations supplémentaires »](#) pour plus d'informations.

Vous pouvez également utiliser l'explorateur de référentiel pour localiser du contenu, et le copier directement dans votre copie de travail, ou entre deux endroits du référentiel. Lisez [Section 4.24, « l'explorateur de référentiel »](#) pour plus d'informations.

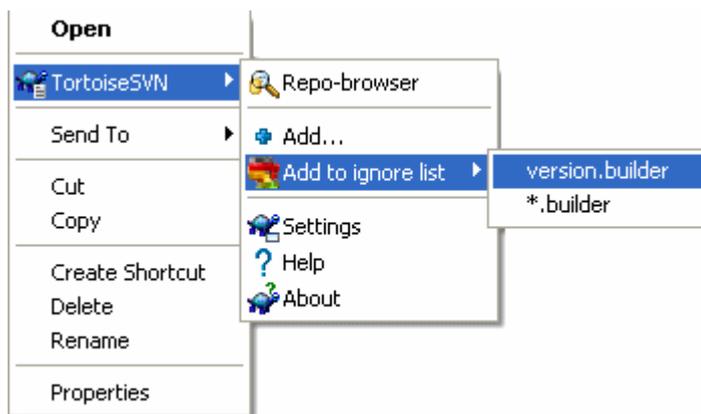


### Impossible de copier entre référentiels

Whilst you can copy and files and folders *within* a repository, you *cannot* copy or move from one repository to another while preserving history using TortoiseSVN. Not even if the repositories live on the same server. All you can do is copy the content in its current state and add it as new content to the second repository.

Si vous n'êtes pas sûr de savoir si deux URL pointant sur le même serveur font référence à un même référentiel, utilisez l'explorateur de référentiel pour ouvrir une des deux et localiser sa racine. Si vous voyez les deux chemins dans la même fenêtre d'explorateur de référentiel alors ils sont dans le même référentiel.

## 4.13. Ignorer des fichiers et des répertoires



**Figure 4.27. Menu contextuel de l'explorateur pour les fichiers non versionnés**

Dans la plupart des projets vous aurez des fichiers et des dossiers qui n'auront pas à être sous contrôle de version. Par exemple, les fichiers de compilation, `*.obj`, `*.lst`, peut-être un dossier destiné à recevoir l'exécutable. Chaque fois que vous livrez des changements, TSVN vous montre les fichiers non versionnés, encombrant la liste des fichiers dans la fenêtre de livraison. Vous pouvez bien sûr désactiver cet affichage, mais vous pourriez alors oublier d'ajouter un nouveau fichier source.

La meilleure façon d'éviter ces problèmes est d'ajouter les fichiers à la liste des ignorés du projet. De cette manière, ils ne s'afficheront jamais dans la fenêtre de livraison, mais les vrais fichiers source non versionnés le seront toujours.

Si vous faites un clic droit sur un seul fichier non versionné et sélectionnez la commande TortoiseSVN → Ajouter à la liste des ignorés à partir du menu contextuel, un sous-menu apparaît vous permettant de ne choisir que ce fichier, ou tous les fichiers avec la même extension. Si vous sélectionnez plusieurs fichiers, il n'y a pas de sous-menu et vous pouvez seulement ajouter ces fichiers/dossiers spécifiques.

Si vous voulez supprimer un ou plusieurs éléments de la liste des ignorés, faites un clic droit sur ces éléments et sélectionnez TortoiseSVN → Retirer de la liste des ignorés Vous pouvez aussi directement accéder à la propriété `svn:ignore` d'un dossier. Cela vous permet de spécifier des modèles

plus généraux en utilisant des jokers, décrit dans la section ci-dessous. Lisez [Section 4.17, « Configuration des projets »](#) pour plus d'informations sur la définition directe des propriétés. Notez qu'il faut une règle de filtrage par ligne. Les séparer d'un espace ne fonctionne pas.



## La liste des ignorés globale

Une autre façon d'ignorer des fichiers est de les ajouter à la *liste des ignorés globale*. La grande différence ici, c'est que la liste des ignorés globale est une propriété client. Elle s'applique à *tous* les projets Subversion, mais sur le PC client uniquement. En général, c'est mieux d'utiliser la propriété `svn:ignore` où c'est possible, parce qu'elle peut être appliquée à des secteurs spécifiques du projet et elle fonctionne pour tous ceux qui extraient le projet. Lisez [Section 4.30.1, « Configuration générale »](#) pour plus d'informations.



## Ignorer les éléments versionnés

Les fichiers et les répertoires versionnés ne peuvent jamais être ignorés - c'est une fonctionnalité de Subversion. Si vous avez versionné un fichier par erreur, lisez [Section B.8, « Ignorer les fichiers déjà versionnés »](#) pour savoir comment le « déversionner ».

### 4.13.1. L'utilisation des pattern matching dans la liste des fichier à ignorer

Les modèles d'exclusion de Subversion se servent de l'expansion des jokers (globbing) dans les noms de fichier, une technique à l'origine utilisée sous Unix pour spécifier des fichiers utilisant des méta-caractères comme caractères de remplacement. Les caractères suivants ont une signification spéciale :

- \*  
Correspond à n'importe quelle chaîne de caractères (de 0 à n caractères), y compris la chaîne vide (aucun caractère).
- ?  
Correspond à n'importe quel caractère.
- [...]  
Correspond à n'importe lequel des caractères inclus dans les crochets. Dans les crochets, une paire de caractères séparés par « - » correspond à n'importe quel caractère lexicalement entre les deux. Par exemple [AGm-p] correspond à A, G, m, n, o ou p.

La correspondance des modèles est sensible à la casse, ce qui peut causer des problèmes sous Windows. Vous pouvez forcer la non sensibilité à la casse en dur en appareillant des caractères, par exemple pour ignorer `*.tmp` indépendamment de la casse, vous pourriez utiliser un modèle comme `*.[Tt][Mm][Pp]`.

Si vous voulez une définition officielle pour l'expansion de jokers (globbing), vous pouvez la trouver dans les spécifications IEEE pour le langage de commande d'interpréteur de commandes [Pattern Matching Notation](#) [[http://www.opengroup.org/onlinepubs/009695399/utilities/xcu\\_chap02.html#tag\\_02\\_13](http://www.opengroup.org/onlinepubs/009695399/utilities/xcu_chap02.html#tag_02_13)].



## Pas de Chemins dans la Liste des Fichiers Ignorés

Vous ne devez pas inclure de chemin d'accès dans vos patterns. Le pattern matching a pour vocation d'être utilisé à la place des noms de fichiers ou de répertoire. Si vous souhaitez ignorer tous les répertoires CVS, ajoutez juste CVS à la liste des ignorés. Il n'y a pas besoin de spécifier `CVS */CVS` comme vous le faisiez dans les versions antérieures. Si vous souhaitez ignorer tous les répertoires tmp lorsqu'ils sont dans un répertoire prog mais pas quand ils sont dans un répertoire doc vous devez utiliser la propriété `svn:ignore` à la place. Il n'y a pas de manière sûre d'avoir ce type de règle de filtrage.

## 4.14. Supprimer, déplacer et renommer

À la différence de CVS, Subversion permet de renommer et de déplacer des fichiers et des dossiers. Il y a donc des entrées de menu pour supprimer et renommer dans le sous-menu de TortoiseSVN.

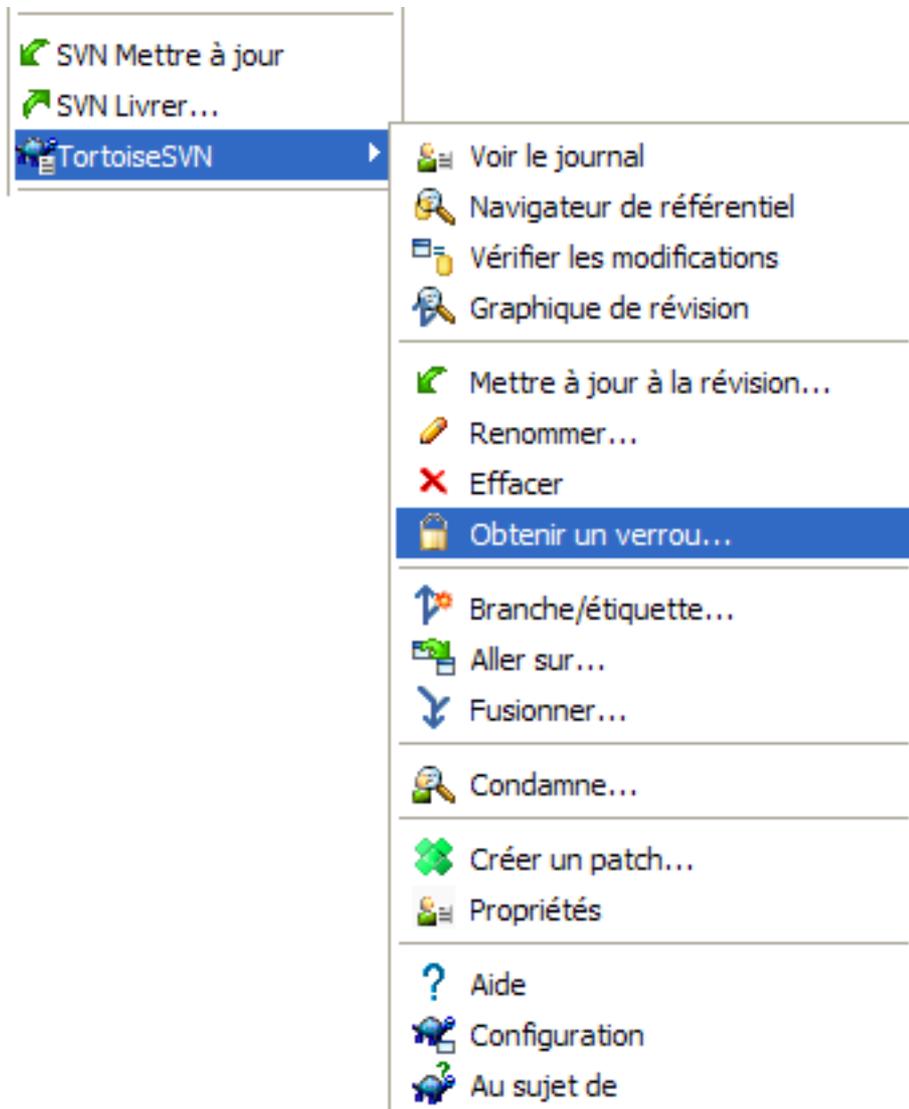


Figure 4.28. Menu contextuel de l'explorateur pour les fichiers non versionnés

### 4.14.1. Supprimer des fichiers et des dossiers

Utiliser TortoiseSVN → Supprimer pour effacer des fichiers ou répertoires dans subversion.

Lorsque vous utilisez la commande TortoiseSVN → Supprimer sur un fichier, il est supprimé immédiatement de votre copie de travail tout en étant marqué pour suppression dans le référentiel jusqu'au prochain envoi au dépôt. Il s'affiche sur le dossier parent du fichier un icône « supprimé ». Jusqu'à ce que vous envoyez les modifications au dépôt, vous pouvez toujours récupérer le fichier comme à l'origine si vous appelez TortoiseSVN → Revenir en arrière sur le dossier parent.

Lorsque vous utilisez la commande TortoiseSVN → Supprimer sur un répertoire, il est conservé dans votre copie de travail, mais l'icône change pour indiquer qu'il est marqué pour suppression. Jusqu'à ce que

vous envoyez les modifications au référentiel, vous pouvez revenir en arrière et récupérer le dossier en faisant **TortoiseSVN → Revenir en arrière** sur le dossier lui-même. Cette différence de comportement entre les fichiers et dossiers est une caractéristique de Subversion, pas de TortoiseSVN.

Si vous voulez supprimer un élément dans le référentiel, mais le conserver localement comme un fichier/répertoire sans version, utilisez **Menu Contextuel Avancé → Supprimer (conserver localement)**. Vous devez tenir la touche **Maj** enfoncée tout effectuant un clic droit avec la souris sur l'élément désiré dans le volet de l'explorateur (volet droit) afin de voir dans le menu contextuel avancé.

Si un *fichier* est supprimé via l'explorateur au lieu d'utiliser le menu contextuel TortoiseSVN, la boîte de dialogue Livrer affiche ces fichiers et vous laisse aussi les supprimer du contrôle de version avant la livraison. Cependant, si vous mettez à jour votre copie de travail, Subversion découvrira le fichier manquant et le remplacera par la dernière version du référentiel. Si vous devez supprimer un fichier sous contrôle de version, utilisez toujours **TortoiseSVN → Supprimer** pour que Subversion n'ait pas à deviner ce que vous voulez vraiment faire.

Si un *dossier* est supprimé via l'explorateur au lieu d'utiliser le menu contextuel TortoiseSVN, votre copie de travail sera cassée et vous serez incapables de livrer. Si vous mettez à jour votre copie de travail, Subversion remplacera le dossier manquant avec la dernière version du référentiel et vous pourrez alors le supprimer correctement en utilisant **TortoiseSVN → Supprimer**.



### Récupérer un fichier ou un répertoire supprimé

Si vous avez supprimé un fichier ou un dossier et avez déjà livré cette opération de suppression au référentiel, alors un normal **TortoiseSVN → Revenir en arrière** normal ne peut plus le ramener. Mais le fichier ou le dossier n'est pas perdu pour autant. Si vous connaissez la révision où le fichier ou le dossier a été supprimé (sinon, utilisez la boîte de dialogue Journal pour le savoir) ouvrez l'explorateur de référentiel et allez à cette révision. Choisissez alors le fichier ou le dossier que vous avez supprimé, faites un clic-droit et choisissez **Menu contextuel → Copier vers...** et comme cible pour cette opération de copie, choisissez le chemin de votre copie de travail.

## 4.14.2. Déplacer des fichiers et des dossiers

Si vous voulez faire un simple renommage d'un fichier ou d'un dossier, utilisez **Menu contextuel → Renommer...** Entrez le nouveau nom de l'objet et vous avez terminé.

Si vous voulez déplacer des fichiers à l'intérieur d'une copie de travail, peut-être vers un différent sous-répertoire, utilisez une nouvelle fois le glisser-déplacer :

1. sélectionnez les fichiers ou les répertoires que vous voulez déplacer
2. glissez-déposer avec le bouton droit ces fichiers dans le nouvel emplacement à l'intérieur de la copie de travail
3. relâchez le bouton droit de la souris
4. Dans le menu qui apparaît, sélectionnez **Menu contextuel → SVN Déplacer les fichiers dans Subversion ici**



### Livrez le répertoire parent

Puisque les renommages et les déplacements sont gérés comme une suppression suivie d'un ajout vous devez livrer le dossier parent du fichier renommé/déplacé pour que la partie

supprimée du renommage/déplacement apparaisse dans la boîte de dialogue Livrer. Si vous ne livrez pas la partie supprimée du renommage/déplacement, il restera dans le référentiel et une mise à jour par vos collègues ne supprimera pas le vieux fichier. C'est-à-dire qu'ils auront *les deux*, les vieilles et les nouvelles copies.

Vous *devez* livrer un renommage de dossier avant de changer l'un des fichiers de ce dossier, autrement votre copie de travail peut être vraiment salie.

Vous pouvez aussi utiliser le navigateur du référentiel pour déplacer des éléments.



### **Ne faites pas SVN Déplacer sur les externes**

Vous ne devriez *pas* utiliser les commandes Déplacer ou Renommer de TortoiseSVN sur un dossier qui a été créé en utilisant `svn:externals`. Cette action causerait la suppression de l'élément externe de son référentiel parent, en dérangeant probablement beaucoup d'autres personnes. Si vous devez déplacer un dossier externe, vous devez utiliser un déplacement ordinaire, ajuster ensuite les propriétés `svn:externals` des répertoires parents de la source et de la destination.

#### **4.14.3. Modifier la casse dans le nom d'un fichier.**

Effectuer des modifications à un nom de fichier est plus délicat avec Subversion sous Windows, parce que pendant un court laps de temps lors du renommage, les deux noms de fichiers existent. Comme Windows a un cas de système de fichiers non sensible à la casse, cela ne fonctionne pas en utilisant la commande habituelle Renommer.

Heureusement, il y a (au moins) deux façons de renommer un fichier sans perdre son historique de journal. C'est important de le renommer au sein de Subversion. Le renommer seulement dans l'explorateur corrompra votre copie de travail!

Solution A) (recommandée)

1. Livrez les changements dans votre copie de travail.
2. Renommez le fichier MAJUScules en majusCULES directement dans le référentiel en utilisant l'explorateur de référentiel.
3. Mettez à jour votre copie de travail.

Solution B)

1. Renommer MAJUScules en MAJUScules\_ avec la commande renommer du sous-menu TortoiseSVN.
2. Livrez les changements.
3. Renommer de MAJUScules\_ en majusCULES.
4. Livrez les changements.

#### **4.14.4. Gestion des conflits de nom de fichier.**

Dans le cas où vous avez deux fichiers dans le référentiel avec le même nom mais qui ne diffèrent qu'avec la casse (par exemple TEST.TXT et test.txt), vous ne pouvez plus mettre à jour ou extraire le répertoire parent sous Windows. Bien que Subversion supporte la casse sur les noms de fichiers, avec Windows cela n'est pas possible.

Cela arrive parfois, lorsque deux personnes envoi les modifications au dépôt, à partir de copies de travail séparées, les fichiers qui se trouvent avoir le même nom, mais avec une casse différence. Cela peut

également être le cas lorsque les fichiers sont envoyés au dépôt depuis un système gérant par défaut la casse de fichiers, comme Linux.

Dans ce cas, vous devez décider lequel des deux vous voulez conserver et supprimer (ou renommer) l'autre du référentiel.



### Éviter que deux fichiers aient le même nom

Il existe un script hook pour le serveur disponible à : <http://svn.collab.net/repos/svn/trunk/contrib/hook-scripts/> qui préviendra les livraisons qui résultent en conflits de casse.

#### 4.14.5. Réparer les renommages de fichier

Parfois, votre environnement de développement va renommer des fichiers pour vous dans le cadre d'une restructuration des sources, et bien sûr il ne le dit pas à Subversion. Si vous essayez d'envoyer au dépôt vos modifications, Subversion peut voir l'ancien nom de fichier comme manquant et le nouveau pas encore versionné. Vous pourriez simplement ajouter le nouveau nom de fichier dans le référentiel, mais on perdrait alors l'historique, Subversion ne sait pas que les fichiers sont liés.

Une meilleure solution est d'informer Subversion que ce changement est en fait un renommage, et vous pouvez le faire par les boîtes de dialogues **Livrer** et **Vérifier les modifications**. Il suffit de sélectionner à la fois l'ancien nom (manquant) et le nouveau nom (sans version) et d'utiliser **Menu contextuel** → **Réparation Déplacer** pour lier les deux fichiers en tant qu'un renommé.

#### 4.14.6. Supprimer les fichiers non versionnés

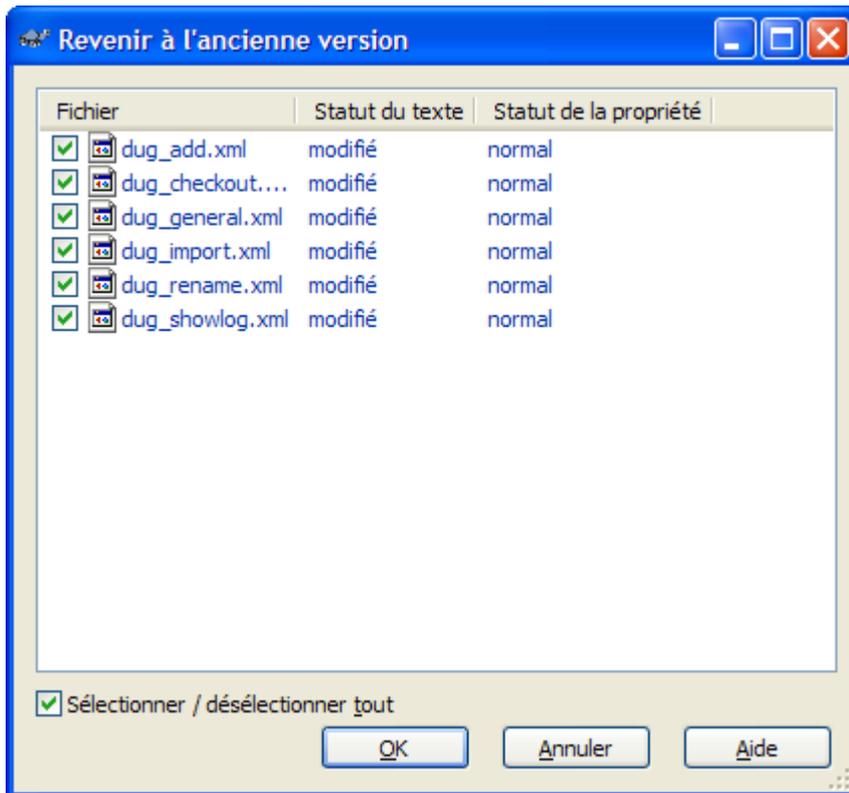
Généralement vous devrez placer votre liste de fichiers ignorés tel que tous les fichiers générés sont ignorés dans Subversion. Mais que faire si vous souhaitez effacer tous les éléments ignorés pour produire une génération propre? Habituellement, vous devez définir cela dans votre makefile, mais si vous débogez le makefile ou changez le système de construction, il est utile d'avoir un moyen de nettoyer la plate-forme.

TortoiseSVN fournit juste une telle option à l'aide **Menu contextuel Étendu** → **Supprimer les éléments non versionnés...** Vous devez tenir enfoncé la touche **Maj** tout en effectuant un clic droit sur un dossier dans le volet de l'explorateur (volet droit) afin de le voir apparaître dans le menu contextuel étendu. Cela a pour effet d'ouvrir une boîte de dialogue qui répertorie tous les fichiers non versionnés n'importe où dans votre copie de travail. Vous pouvez ensuite sélectionner ou désélectionner les éléments à enlever.

Lorsque ces éléments sont supprimés, la corbeille est utilisée, donc si vous faites une erreur ici et supprimez un fichier qui aurait dû être versionné, vous pouvez toujours le récupérer.

### 4.15. Annuler les changements

Si vous voulez défaire tous les changements que vous avez fait dans un fichier depuis la dernière mise à jour, vous devez sélectionner le fichier, faites un clic droit pour faire apparaître le menu contextuel et sélectionnez ensuite la commande **TortoiseSVN** → **Revenir en arrière** Une boîte de dialogue apparaîtra vous montrant les fichiers que vous avez changés et que vous pouvez restaurer. Choisissez ceux que vous voulez restaurer et cliquez sur **OK**.



**Figure 4.29. La boîte de dialogue Revenir en arrière**

Si vous souhaitez annuler une suppression ou un renommage, vous devez utiliser "Revenir en arrière" sur le dossier parent car l'élément supprimé n'existe pas pour vous permettre de faire un clic droit dessus.

Si vous souhaitez annuler l'ajout d'un élément, cela apparaît dans le menu contextuel comme TortoiseSVN → Annuler l'ajout.... C'est vraiment un retour arrière, mais le nom a été changé pour le rendre plus évident.

Les colonnes dans cette boîte de dialogue peuvent être personnalisées de la même manière que les colonnes dans la boîte de dialogue Vérifier les modifications. Lisez [Section 4.7.3, « Statut local et distant »](#) pour plus de détails.



### Annuler les changements qui ont été livrés

Revenir en arrière n'annulera que vos changements locaux. Cela n'annulera *pas* les changements déjà livrés. Si vous voulez annuler tous les changements livrés dans une révision spécifique, lisez [Section 4.9, « La boîte de dialogue du Journal de révision »](#) pour plus d'informations.



### Le retour en arrière est Lent

Lorsque vous faites un retour arrière sur les modifications et que vous trouvez mais que l'opération prend beaucoup plus longtemps que prévu. C'est parce que la version modifiée du fichier est envoyé à la corbeille, vous pouvez donc récupérer vos modifications si vous faite un retour arrière par erreur. Toutefois, si votre corbeille est pleine, Windows prend beaucoup de temps pour trouver un endroit pour mettre le fichier. La solution est simple : soit vider la corbeille ou désactiver l'option Utiliser la corbeille lors d'un retour en arrière dans les paramètres de configuration de TortoiseSVN.

## 4.16. Nettoyer

Si une commande Subversion ne peut pas s'achever avec succès, peut-être en raison de problèmes serveur, votre copie de travail peut être laissée dans un état incohérent. Dans ce cas, vous devez utiliser TortoiseSVN → Nettoyer sur le dossier. C'est une bonne idée de le faire au niveau supérieur de la copie de travail.

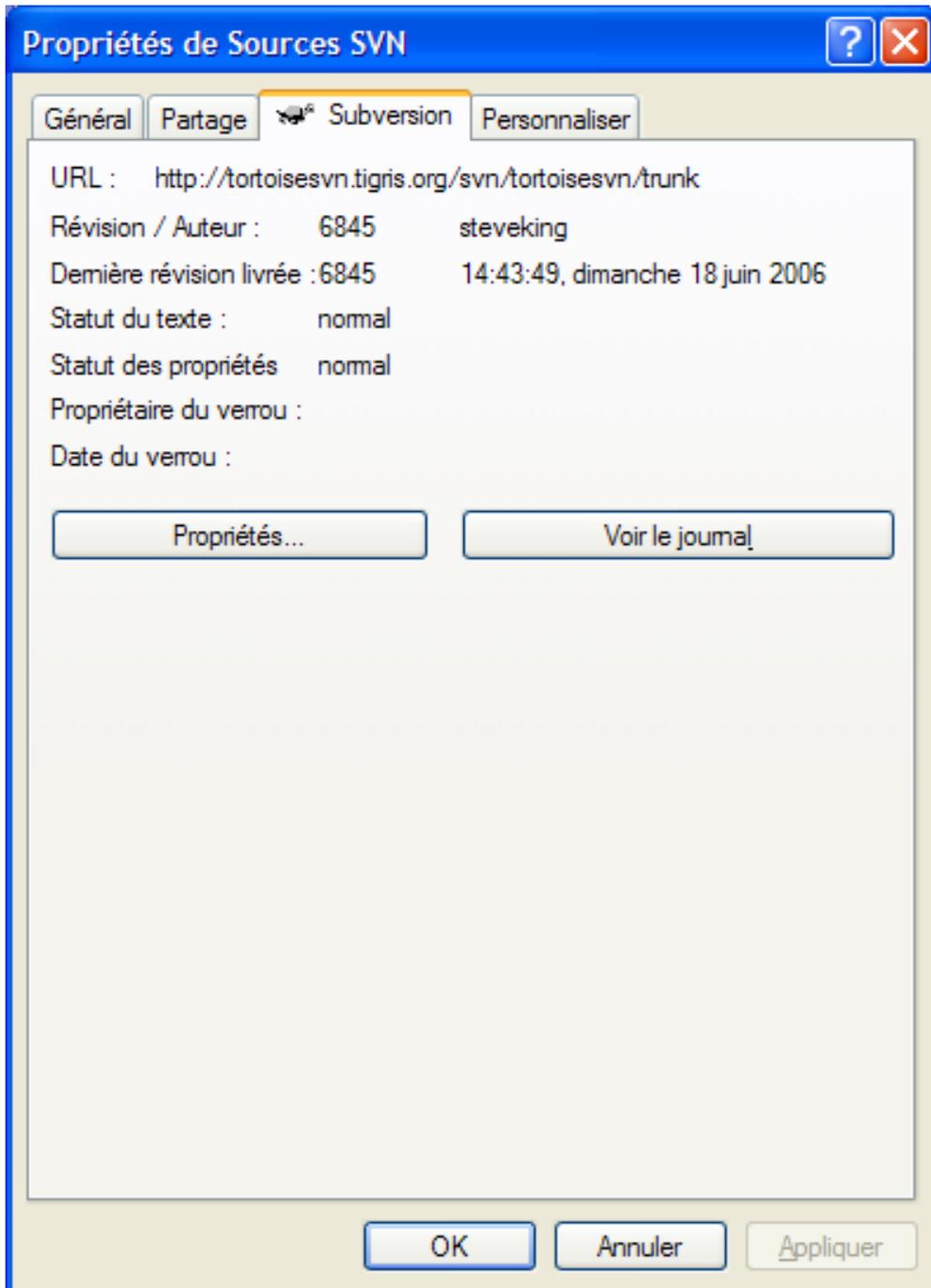
Le nettoyage a un autre effet secondaire utile. Si une date de fichier change mais pas son contenu, Subversion ne peut dire s'il a vraiment changé qu'en faisant une comparaison octet-par-octet avec la copie primitive. Si vous avez beaucoup de fichiers dans cet état, cela rend la récupération du statut très lente, ce qui fera que beaucoup de boîtes de dialogue seront lentes à répondre. Exécuter un nettoyage sur votre copie de travail réparera ces horodatages « cassés » et rendra aux contrôles de statut leur pleine vitesse.



### Utiliser les horodatages de livraison

Des sorties précédentes de Subversion ont été affectées par un bug qui a causé des incohérences dans l'horodatage quand vous extrayez avec l'option Utiliser les horodatages de livraison cochée. Utilisez la commande Nettoyer pour accélérer ces copies de travail.

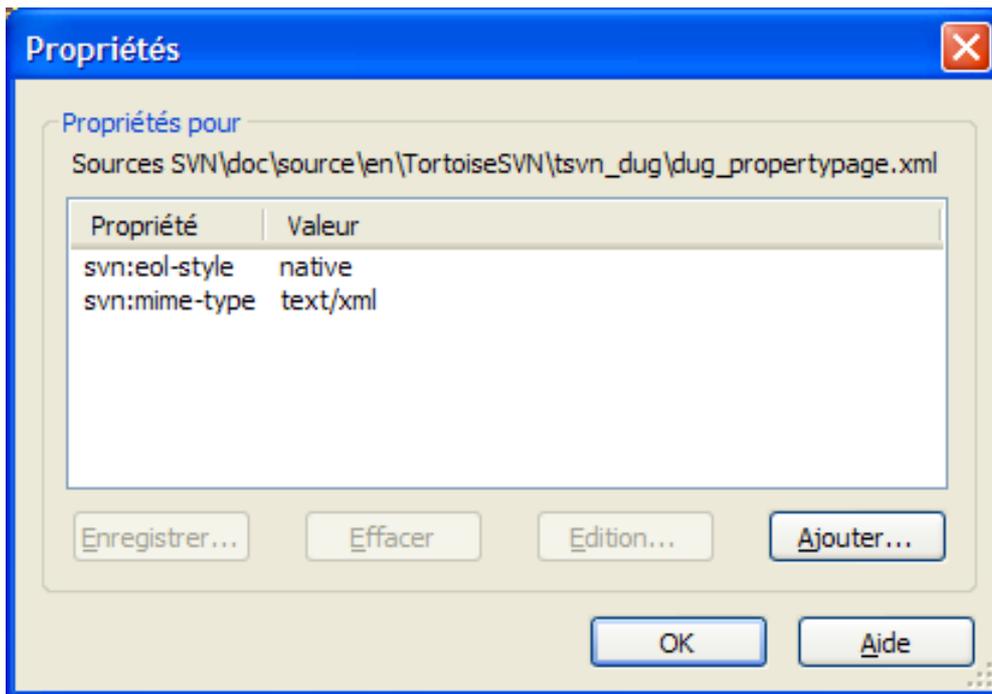
## 4.17. Configuration des projets



**Figure 4.30. Page de propriétés de l'explorateur, onglet Subversion**

Parfois vous voulez avoir des informations plus détaillées sur un fichier/répertoire qu'un recouvrement d'icône. Vous pouvez obtenir toutes les informations que Subversion fournit dans la boîte de dialogue de propriétés de l'explorateur. Sélectionnez seulement le fichier ou le répertoire et choisissez **Menu Windows** → **Propriétés** dans le menu contextuel (note : c'est l'entrée de menu de propriétés standard que l'explorateur fournit, pas celle du sous-menu TortoiseSVN !). Dans la boîte de dialogue de propriétés, TortoiseSVN a ajouté une nouvelle page de propriété pour les fichiers/dossiers sous le contrôle de Subversion où vous verrez toutes les informations pertinentes concernant le fichier/répertoire sélectionné.

#### 4.17.1. Propriétés Subversion



**Figure 4.31. Page de propriété de subversion**

Vous pouvez lire et définir les propriétés Subversion à partir de la boîte de dialogue de propriétés de Windows, mais aussi depuis TortoiseSVN → Propriétés et au sein des listes de statut de TortoiseSVN, depuis Menu contextuel → Propriétés.

Vous pouvez ajouter vos propres propriétés, ou des propriétés avec une signification spéciale pour Subversion. Celles-ci commencent par `svn:`. `svn:externals` est une des propriétés; regardez comment manipuler les propriétés externes dans [Section 4.18, « Éléments externes »](#).

#### 4.17.1.1. svn:keywords

Subversion supporte le mécanisme de mots-clés comme pour CVS qui peuvent être utilisés pour intégrer le nom de fichier et la révision des informations dans le fichier lui-même. Les mots-clés actuellement supportés sont:

##### \$Date\$

Dernière date de livraison connue. Cette information est obtenue quand vous mettez à jour votre copie de travail. Elle *ne vérifie pas* le référentiel pour trouver d'éventuels changements récents.

##### \$Revision\$

Révision de la dernière livraison connue.

##### \$Author\$

Auteur qui a fait la dernière livraison connue.

##### \$HeadURL\$

Le chemin complet de ce fichier dans le référentiel.

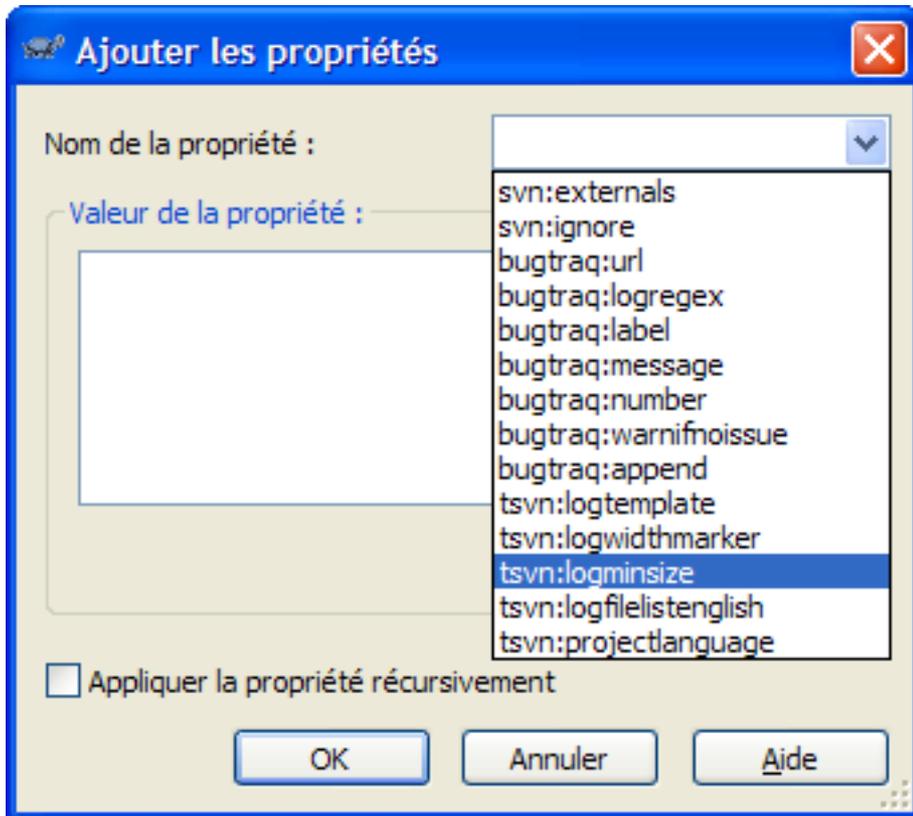
##### \$Id\$

Une combinaison raccourcie des quatre mot clés précédents.

Pour savoir comment utiliser ces mots-clés, regarder le [svn:keywords section](http://svnbook.red-bean.com/en/1.5/svn.advanced.props.special.keywords.html) [http://svnbook.red-bean.com/en/1.5/svn.advanced.props.special.keywords.html] dans le livre Subversion (en Anglais), qui donne une description complète de ces mots clés et la façon de les activer et de les utiliser.

Pour plus d'informations sur les propriétés dans Subversion, regarder les *Propriétés Avancées* [<http://svnbook.red-bean.com/en/1.5/svn.advanced.props.html>].

#### 4.17.1.2. Ajouter et Modifier les propriétés



**Figure 4.32. Ajouter des propriétés**

Pour ajouter une nouvelle propriété, cliquez d'abord sur *Ajouter...* Choisissez le nom de la propriété requise depuis la liste, ou tapez le nom de votre choix, entrez ensuite une valeur dans le champ en dessous. Les propriétés qui prennent des valeurs multiples, comme la liste des ignorés, peuvent être saisies sur plusieurs lignes. Cliquez sur *OK* pour ajouter la propriété à la liste.

Si vous voulez appliquer une propriété à plusieurs éléments à la fois, sélectionnez les fichiers/dossiers dans l'explorateur, puis choisissez *Menu contextuel* → *Propriétés*

Si vous voulez appliquer la propriété à *tous* les fichiers et dossiers au-dessous du dossier actuel dans la hiérarchie, cochez la case *Récursive*.

Certaines propriétés, par exemple `svn:needs-lock`, peuvent seulement être appliquées aux fichiers, donc le nom de la propriété n'apparaît pas dans la liste déroulante pour les dossiers. Vous pouvez toujours appliquer une telle propriété récursivement à tous les fichiers d'une hiérarchie, mais vous devez taper le nom de la propriété vous même.

Si vous voulez éditer une propriété existante, sélectionnez cette propriété dans la liste des propriétés existantes, puis cliquez sur *Editer...*

Si vous voulez supprimer une propriété existante, sélectionnez cette propriété dans la liste des propriétés existantes, puis cliquez sur *Effacer*.

La propriété `svn:externals` peut être utilisée pour intégrer d'autres projets du même référentiel ou d'un référentiel complètement différent. Pour plus d'informations, lisez [Section 4.18, « Eléments externes »](#).

### 4.17.1.3. Exporter et Importer les Propriétés

Souvent, vous vous trouverez appliquant le même ensemble de propriétés de nombreuses fois, par exemple `bugtraq:logregex`. Pour simplifier le processus de copie des propriétés d'un projet à l'autre, vous pouvez utiliser la fonctionnalité d'Export/Import.

A partir du fichier ou du dossier où les propriétés sont déjà définies, utilisez TortoiseSVN → propriétés, sélectionnez les propriétés que vous souhaitez exporter et cliquez sur Export.... Vous serez invité à entrer un nom de fichier où les noms des propriétés et valeurs seront sauvegardés.

Dans le(s) dossier(s) où vous souhaitez appliquer ces propriétés, utilisez TortoiseSVN → propriétés et cliquez sur Importer .... Vous serez invité à entrer un nom de fichier à importer, naviguer alors vers le lieu où vous avez enregistré précédemment le fichier d'exportation et sélectionnez-le. Les propriétés seront ajoutés aux dossiers de manière non-réursive.

Si vous souhaitez ajouter des propriétés à un arbre de manière réursive, suivez les étapes ci-dessus, puis dans la boîte de dialogue de propriété, sélectionnez chaque propriété, cliquez sur Modifier ..., vérifiez la boîte Appliquer la propriété récurivement et cliquez sur OK.

Le format de fichier d'importation est binaire et la propriété de TortoiseSVN. Son seul but est de transférer les propriétés en utilisant l'importation et l'exportation, de sorte qu'il n'est pas nécessaire de modifier ces fichiers.

### 4.17.1.4. Propriétés

TortoiseSVN peut manipuler des valeurs de propriété binaires en utilisant des fichiers. Pour lire une valeur de propriété binaire, Enregistrer... vers un fichier. Pour mettre une valeur binaire, utilisez un éditeur hexadécimal ou un autre outil approprié pour créer un fichier avec le contenu dont vous avez besoin, puis Charger... ce fichier.

Bien que les propriétés binaires ne soient pas très utilisées, elles peuvent être utiles dans certaines applications. Par exemple si vous stockez d'énormes fichiers graphiques ou si l'application utilisée pour charger le fichier est énorme, vous pourriez vouloir stocker un aperçu en tant que propriété pour obtenir une prévisualisation rapide.

### 4.17.1.5. Configuration automatique des propriétés

Vous pouvez configurer Subversion et TortoiseSVN pour définir automatiquement des propriétés sur les fichiers et sur les dossiers lorsqu'ils sont ajoutés au référentiel. Il y a deux manières de faire cela.

Vous pouvez éditer le fichier de configuration Subversion pour activer cette fonctionnalité sur votre client. La partie Général de la fenêtre de configuration de TortoiseSVN a un bouton "Editer" pour afficher le fichier directement. Le fichier de configuration est un simple fichier texte qui contrôle certains des rouages de Subversion. Vous devez modifier deux choses : d'abord dans la section intitulée `miscellany` décommenter la ligne `enable-auto-props = yes`. Ensuite, vous devez modifier la section dessous pour définir quelles propriétés vous souhaitez ajouter à quels types de fichiers. Cette méthode est une caractéristique standard de Subversion et fonctionne avec n'importe quel client Subversion. Cependant, il doit être défini sur chaque client individuellement - il n'existe aucun moyen pour propager ces paramètres à partir du référentiel.

Une autre méthode consiste à définir la propriété `tsvn:autoprops` sur les dossiers, tel que décrit dans la section suivante. Cette méthode fonctionne uniquement pour les clients TortoiseSVN, et elle se propage à toutes les copies de travail mises à jour.

Peu importe la méthode que vous choisissiez, vous devez noter que auto-props n'est appliquée aux fichiers qu'au moment où ils sont ajoutés au référentiel. Auto-props ne changera jamais les propriétés des fichiers qui sont déjà versionnés.

Si vous voulez être absolument sûr que les nouveaux fichiers ont les bonnes propriétés appliquées, vous devez mettre en place un référentiel "hook" pré-livré pour rejeter les livraisons dont les propriétés requises ne sont pas définies.



## Livrer les propriétés

Les propriétés de Subversion sont versionnées. Après avoir changé ou ajouté une propriété, vous devez livrer vos changements.



## Conflits sur les propriétés

S'il y a un conflit en livrant les changements, parce qu'un autre utilisateur a changé la même propriété, Subversion génère un fichier `.prej`. Supprimez ce fichier après avoir résolu le conflit.

### 4.17.2. Propriétés du projet TortoiseSVN

TortoiseSVN a quelques propriétés spéciales de son cru et elles commencent par `tsvn:`.

- `tsvn:logminsize` définit la longueur minimale d'un commentaire pour une livraison. Si vous entrez un message plus court qu'indiqué ici, la livraison est désactivée. Cette fonctionnalité est très utile pour vous rappeler de fournir un message descriptif approprié à chaque livraison. Si cette propriété n'est pas définie, ou si la valeur est zéro, les commentaires vides sont autorisés.

`tsvn:lockmsgminsize` définit la longueur minimale d'un commentaire pour un commentaire de verrou. Si vous entrez un message plus court qu'indiqué ici, le verrouillage est désactivé. Cette fonctionnalité est très utile pour vous rappeler de fournir un message descriptif approprié à chaque verrou. Si cette propriété n'est pas définie, ou si la valeur est zéro, les commentaires de verrou vides sont autorisés.

- `tsvn:logwidthmarker` est utilisé avec les projets qui exigent que les commentaires soient formatés avec une certaine largeur maximale (généralement 80 caractères) avant un saut de ligne. Définir cette propriété à une valeur non nulle fera 2 choses dans la boîte de dialogue d'entrée de commentaire : cela placera un marqueur pour indiquer la largeur maximale et cela désactivera le retour à la ligne automatique dans l'affichage, pour que vous puissiez voir si le texte que vous avez saisi est trop long. Note : cette fonction ne fonctionnera correctement que si vous avez choisi une police à largeur de caractères fixe pour les commentaires.
- `tsvn:logtemplate` est utilisé avec les projets qui ont des règles de formatage des commentaires. La propriété contient une chaîne de caractères multi-ligne qui sera insérée dans le champ de message de la livraison quand vous commencez une livraison. Vous pouvez alors l'éditer pour inclure les informations requises. Note : si vous utilisez aussi `tsvn:logminsize`, assurez-vous de définir une longueur plus longue que le modèle ou vous perdrez le mécanisme de protection.
- Subversion vous permet de définir les « propriétés automatiques (autoprops) » qui seront appliqués aux fichiers nouvellement ajoutés ou importés, basé sur l'extension des fichiers. Cela dépend de chaque "autoprops" configuré par le client dans son fichier de configuration Subversion. `tsvn:autoprops` peut être réglé sur les dossiers et ceux-ci seront fusionnés avec l'autoprops local de l'utilisateur lors de l'importation ou de l'ajout de fichiers. Le format est le même que pour l'autoprops de subversion, par exemple `*.sh = svn:eol-style=native;svn:executable` définit deux propriétés sur les fichiers avec l'extension `.sh`.

S'il y a un conflit en les propriétés autoprops locales et `tsvn:autoprops`, les propriétés du projet ont la priorité car elles lui sont spécifiques.

- Dans la boîte de dialogue Livrer, vous avez une option pour coller la liste des fichiers changés, en incluant le statut de chaque fichier (ajouté, modifié, etc). `tsvn:logfilelistenglish` définit si

le statut de fichier est inséré en Anglais ou dans la langue locale. Si la propriété n'est pas définie, la valeur par défaut est `true`.

- TortoiseSVN peut utiliser des modules de vérification orthographique qui sont aussi utilisés par OpenOffice et Mozilla. Si ceux-ci sont installés, cette propriété déterminera quel vérificateur d'orthographe utiliser, c'est-à-dire en quelle langue les commentaires de votre projet doivent être écrits. `tsvn:projectlanguage` définit le module de langue que le moteur de vérification orthographique doit utiliser quand vous entrez un commentaire. Vous pouvez trouver les valeurs pour votre langue sur cette page : [MSDN : Language Identifiers](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/intl/nls_238z.asp) [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/intl/nls\_238z.asp].

Vous pouvez saisir cette valeur en décimal, ou en hexadécimal si préfixée avec `0x`. Par exemple l'Anglais (US) peut être entré comme `0x0409` ou `1033`.

- La propriété `tsvn:logsummary` est utilisée pour extraire une partie des commentaires étant destinée à être utilisée comme résumé dans la fenêtre des commentaires.

La valeur de la propriété `tsvn:logsummary` doit être une expression régulière d'une ligne contenant un groupe. Tout ce qui correspond à ce groupe sera utilisé comme résumé.

Un exemple : `\[SUMMARY\]:\s+(.*)` gardera tout ce qui est après « [SUMMARY] » dans le commentaire et l'utilisera comme résumé.

- Lorsque vous souhaitez ajouter une nouvelle propriété, vous pouvez soit en choisir une dans la liste déroulante, soit ajouter n'importe quel nom de propriété qui vous convient. Si votre projet utilise certaines propriétés personnalisées, et que vous voulez que ces propriétés apparaissent dans la liste déroulante (pour éviter les fautes de frappe lorsque vous entrez un nom de propriété), vous pouvez créer une liste de vos propriétés personnalisées à l'aide `tsvn:userfileproperties` et `tsvn:userdirproperties`. Appliquez ces propriétés à un dossier. Quand vous éditez les propriétés de n'importe quel élément enfant, vos propriétés personnalisées apparaissent dans la liste des propriétés prédéfinies.

Certaines propriétés `tsvn:` exigent une valeur `true/false`. TortoiseSVN comprend aussi `yes` comme synonyme de `true` et `no` comme synonyme de `false`.

TortoiseSVN peut s'intégrer à quelques outils de gestion de bugs. Cela utilise des propriétés qui commencent par `bugtraq:`. Lisez [Section 4.28, « Intégration avec des systèmes de gestion de bug / gestion d'incidents »](#) pour plus d'informations.

Il peut aussi s'intégrer à certains explorateurs de référentiel en ligne, en utilisant des propriétés qui commencent par `webviewer:`. Lisez [Section 4.29, « Intégration avec des explorateur de référentiel de type web. »](#) pour plus d'informations.



## Fixer les propriétés du projet dans les dossiers

Ces propriétés spéciales doivent être mises sur les *dossiers* pour que le système fonctionne. Quand vous livrez un fichier ou un dossier, les propriétés sont lues à partir de ce dossier. Si les propriétés n'y sont pas trouvées, TortoiseSVN remontera dans l'arborescence des dossiers pour les trouver, jusqu'à ce qu'il arrive à un dossier non versionné, ou jusqu'à ce que la racine de l'arbre (par exemple `C:\`) soit trouvée. Si vous pouvez être sûr que chaque utilisateur extraie seulement du `trunk/` par exemple et non d'un sous-dossier, il suffit de mettre les propriétés sur `trunk/`. Si vous ne pouvez pas en être sûr, vous devriez définir les propriétés récursivement sur chaque sous-dossier. Une propriété définie plus profondément dans la hiérarchie du projet ignore les réglages faits sur les niveaux supérieurs (plus près de `trunk/`).

For project properties *only* you can use the **Recursive** checkbox to set the property to all sub-folders in the hierarchy, without also setting it on all files.

When you add new sub-folders using TortoiseSVN, any project properties present in the parent folder will automatically be added to the new child folder too.



## Attention

Bien que les propriétés du projet TortoiseSVN soient extrêmement utiles, elles fonctionnent uniquement avec TortoiseSVN, et certaines ne fonctionnent que dans les versions les plus récentes de TortoiseSVN. Si les personnes qui travaillent sur votre projet utilisent une variété de clients Subversion, ou peut-être dsiposent d'anciennes versions de TortoiseSVN, vous devriez utiliser des référentiels "hooks" pour renforcer les politiques du projet. Les propriétés du projet ne peuvent que contribuer à mettre en œuvre une politique, elles ne peuvent pas l'imposer.

## 4.18. Eléments externes

Il est parfois utile de construire une copie de travail faite d'un certain nombre d'extractions différentes. Par exemple, vous pouvez vouloir que des fichiers ou des sous-répertoires différents viennent d'emplacements différents dans un référentiel, ou peut-être même de référentiels différents. Si vous voulez que chaque utilisateur ait la même disposition, vous pouvez définir les propriétés `svn:externals` de façon à positionner les ressources spécifiées dans les emplacements cibles souhaités.

### 4.18.1. Répertoires externes

Si vous souhaitez extraire une copie de travail du `/projet1` dans `D:\dev\projet1`. Sélectionnez le dossier `D:\dev\projet1`, faites un click droit et choisissez **Menu Windows** → **Propriétés** dans le menu contextuel. La boîte de dialogue **Propriétés** apparaît. Allez alors dans l'onglet **Subversion**. Il vous permet de modifier les propriétés. Cliquez **Ajouter...** Choisissez la propriété `svn:externals` dans la liste déroulante et écrivez dans la boîte de saisie l'URL du référentiel au format `url dossier` ou si vous voulez spécifier une révision particulière, `-rREV url dossier`. Vous pouvez ajouter plusieurs projets externes, 1 par ligne. Supposons que vous ayez saisi les propriétés suivantes sur `D:\dev\projet1` :

```
http://sounds.red-bean.com/repos sounds
http://graphics.red-bean.com/repos/fast%20graphics "quick_graphs"
-r21 http://svn.red-bean.com/repos/skin-maker skins/toolkit
```

Cliquez ensuite sur **Définir** et livrez vos changements. Quand vous (ou tout autre utilisateur) mettez à jour votre copie de travail, Subversion créera un sous-dossier `D:\dev\projet1\sounds` et extraira le projet des sons, un autre sous-dossier `D:\dev\projet1\quick_graphs` contenant le projet des graphiques et finalement un sous-dossier imbriqué `D:\dev\projet1\skins\toolkit` contenant la révision 21 du projet de fabrication des thèmes.

Notez que les URL doivent être correctement saisies ou elles ne fonctionneront pas. Par exemple vous devez remplacer chaque espace par `%20`, comme dans le second exemple ci-dessus.

Si vous souhaitez que le chemin local contienne des espaces ou d'autres caractères spéciaux, vous pouvez les encadrer avec des doubles cotes, ou vous pouvez utiliser le caractère `\` (backslash) comme un type de caractère shell Unix pour distinguer tout caractère spécial qu'il précède. Bien entendu, cela signifie que vous devez utiliser `/` (slash) comme délimiteur de chemin. Notez que ce comportement est nouveau dans Subversion 1.6 et ne fonctionnera pas avec les versions antérieures.



## Utiliser des numéros de révision explicites

Vous devriez vous astreindre à l'utilisation de numéros de révision explicites dans toutes vos définitions externes, comme décrit ci-dessus. Cela revient à décider quand choisir un

instantané différent d'informations externes et quel instantané particulier intégrer. En plus du bon sens consistant à ne pas être pas étonné par les changements aux référentiels tiers sur lesquels vous pourriez n'avoir aucun contrôle, utiliser des numéros de révision explicites signifie aussi que lorsque vous ramenez votre copie de travail à une révision précédente, vos définitions externes réintègrent également l'état qu'elles avaient dans cette révision précédente, ce qui signifie en conséquence que les copies de travail externes seront mises à jour pour correspondre à l'état qu'elles avaient quand votre référentiel était à cette révision précédente. Pour des projets logiciels, ce pourrait être la différence entre une réussite et un échec lors de la construction d'un instantané antérieur de votre complexe base de code.



## définitions svn:externals plus anciennes

Le format présenté ici a été introduit dans Subversion 1.5. Vous pouvez également rencontrer l'ancien format, constitué des mêmes informations dans un ordre différent. Le nouveau format est préféré, car il supporte plusieurs fonctionnalités utiles décrites ci-dessous, mais il ne fonctionne pas avec les clients des versions précédentes. Les différences sont présentées dans le *Subversion Book* [<http://svnbook.red-bean.com/en/1.5/svn.advanced.externals.html>].

If the external project is in the same repository, any changes you make there there will be included in the commit list when you commit your main project.

Si le projet externe est dans un référentiel différent, les changements que vous faites au projet externe seront signalés quand vous livrez le projet principal, mais vous devez livrer ces changements externes séparément.

Si vous utilisez des URL absolues dans la valeur de l'option `svn:externals` and que vous déplacez votre copie de travail (i.e., si l'URL de votre référentiel change), alors vos références externes ne seront pas mises à jour et ne fonctionneront plus.

Pour éviter de tels problèmes, à partir de la version 1.5 du client, Subversion permet d'utiliser les chemins relatifs dans les références externes. Quatre méthodes différentes de spécifications d'URL relatives sont supportées. Dans les exemples suivants, considérez qu'il y a deux référentiels: un premier ici `http://exemple.com/svn/repos-1` et un second là `http://exemple.com/svn/repos-2`. Nous avons une extraction de `http://exemple.com/svn/repos-1/project/trunk` dans `C:\Working` et la propriété `svn:externals` est activée sur le répertoire `trunk`.

Relatif au répertoire parent.

Ces URL commencent toujours par la chaîne de caractère `../` par exemple:

```
../../widgets/foo common/foo-widget
```

Extraira `http://exemple.com/svn/repos-1/widgets/foo` dans `C:\Working\common\foo-widget`.

Notez que l'URL est relative à l'URL du répertoire ayant la propriété `svn:externals`, pas à l'emplacement physique où le répertoire ayant la référence externe est stocké.

Relatif à la racine du référentiel

Ces URLs commencent toujours par la chaîne de caractère `^/` par exemple:

```
^/widgets/foo common/foo-widget
```

Extraira `http://exemple.com/svn/repos-1/widgets/foo` dans `C:\Working\common\foo-widget`.

Vous pouvez facilement faire référence à d'autres référentiels ayant le même `SVNParentPath` (un répertoire commun hébergeant plusieurs référentiels). Par exemple:

```
^/../../repos-2/hammers/claw common/claw-hammer
```

Extraira `http://exemple.com/svn/repos-2/hammers/claw` dans `C:\Working\common\claw-hammer`.

#### Relatif au thème

Les URL qui commencent par `//` ne comportent que la partie "scheme" de l'URL. C'est utile quand l'hôte est accédé de différentes manières selon l'emplacement du client ; i.e. on utilise `http://` depuis l'intranet alors que le protocole `svn+ssh://` est nécessaire depuis l'extérieur. Par exemple :

```
//exemple.com/svn/repos-1/widgets/foo common/foo-widget
```

Extraira `http://exemple.com/svn/repos-1/widgets/foo` ou `svn+ssh://exemple.com/svn/repos-1/widgets/foo` selon la méthode utilisée pour faire la livraison de `C:\Working`.

#### Relatif au nom de domaine du serveur

Les URL qui commencent par la chaîne `/` ne contiennent que la partie "scheme" et le nom d'hôte de l'URI, par exemple :

```
/svn/repos-1/widgets/foo common/foo-widget
```

Extraira `http://exemple.com/svn/repos-1/widgets/foo` dans `C:\Working\common\foo-widget`. Mais si vous extrayez votre copie de travail depuis un autre serveur `svn+ssh://autre.miroir.net/svn/repos-1/projet1/trunk` alors la référence externe extraira `svn+ssh://autre.miroir.net/svn/repos-1/widgets/foo`.

Vous pouvez également préciser une révision d'ancrage ("peg") après l'URL, si nécessaire, par exemple: `http://sounds.red-bean.com/repos@19`.

Si vous avez besoin de plus d'informations sur la façon dont TortoiseSVN manipule les propriétés, lisez [Section 4.17, « Configuration des projets »](#).

Pour découvrir des méthodes différentes d'accéder aux sous-projets communs, lisez [Section B.6, « Inclure un sous-projet commun »](#).

## 4.18.2. Fichiers externes

A partir de Subversion 1.6 vous pouvez ajouter des références externes relatives à des fichiers en utilisant la même syntaxe que pour les dossiers. Il y a néanmoins quelques limitations.

- Le chemin vers le fichier en référence externe doit placer le fichier dans un dossier existant versionné. En général, il est logique de placer le fichier directement dans le dossier dont la propriété `svn:externals` est activée, mais cela peut être dans un sous-dossier versionné si nécessaire. A l'inverse, les répertoires externes vont créer automatiquement tout dossier intermédiaire non versionné requis.
- L'URL pour un fichier en référence externe doit se trouver dans le même référentiel que l'URL dans laquelle le fichier externe sera inséré; les fichiers en référence externe inter-référentiel ne sont pas supportés.

Par beaucoup d'aspects, les fichiers en référence externe se comportent comme n'importe quel autre fichier versionné. Néanmoins, ils ne peuvent pas être déplacés ou supprimés avec les commandes standards; il faut passer par la modification de la propriété `svn:externals`.



## Le support des fichiers en référence externe est incomplet dans Subversion 1.6

Dans Subversion 1.6, il n'est pas possible de retirer un fichier en référence externe de votre copie de travail une fois que vous l'y avez ajouté, même si vous supprimez complètement la propriété `svn:externals`. Vous devez extraire une nouvelle copie de travail pour retirer le fichier.

### 4.19. Brancher / Étiqueter

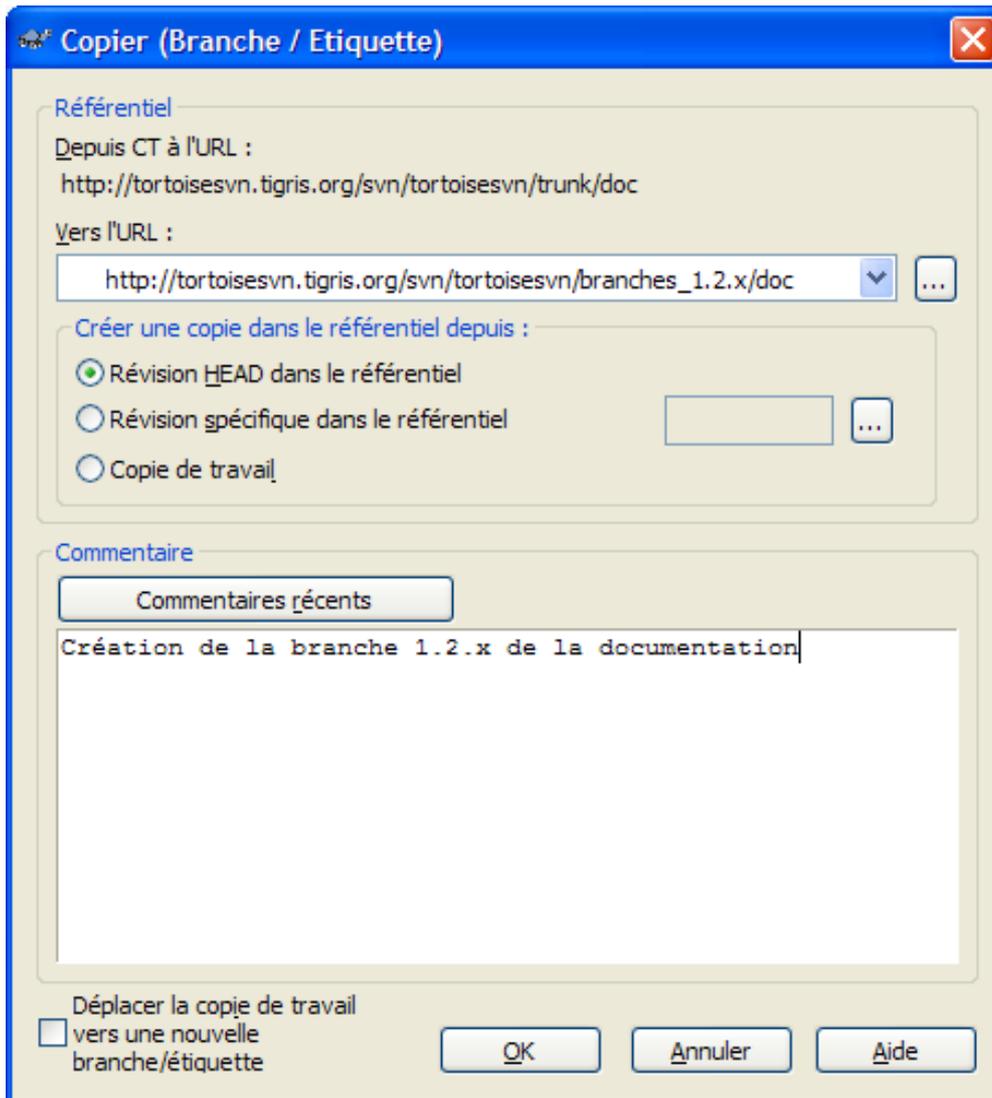
La capacité d'isoler des changements sur une ligne de développement séparée est une des fonctionnalités des systèmes de contrôle de version. Cette ligne est connue comme une *branche*. Les branches sont souvent utilisées pour expérimenter de nouvelles fonctionnalités sans déranger la ligne de développement principale avec des erreurs de compilation et des bugs. Dès que la nouvelle fonction est assez stable alors la branche de développement est *fusionnée* vers la branche principale (le tronc).

Une autre fonctionnalité des systèmes de contrôle de version est la capacité de marquer des révisions particulières (par exemple une version déployée), donc vous pouvez à tout moment recréer une certaine version de votre application ou un environnement. Ce processus est connu comme l'*étiquetage*.

Subversion n'a pas de commandes spéciales pour créer des branches ou des étiquettes, mais utilise de prétendues `copies bon marché` à la place. Les copies bon marché sont semblables aux hard links d'Unix, ce qui signifie qu'au lieu de faire une copie complète dans le référentiel, un lien interne est créé, pointant vers un arbre/révision spécifique. En conséquence, branches et étiquettes sont très rapides à créer et ne prennent presque aucun espace supplémentaire dans le référentiel.

#### 4.19.1. Créer une branche ou une étiquette

Si vous avez importé votre projet avec la structure de répertoire recommandée, créer une branche ou une étiquette est très simple :



**Figure 4.33. La boîte de dialogue Branche/Etiquette**

Sélectionnez le dossier de votre copie de travail que vous voulez copier dans une branche ou une étiquette, choisissez ensuite la commande TortoiseSVN → Branche/Étiquette....

L'URL de destination par défaut de la nouvelle branche sera l'URL source sur laquelle votre copie de travail est basée. Vous devrez éditer cette URL vers le nouveau chemin de votre branche/étiquette. Ainsi au lieu de

```
http://svn.collab.net/repos/ProjectName/trunk
```

vous pourriez maintenant utiliser quelque chose comme

```
http://svn.collab.net/repos/ProjectName/tags/Release_1.10
```

Si vous ne pouvez pas vous rappeler la convention de nommage que vous avez utilisée la dernière fois, cliquer sur le bouton à droite pour ouvrir l'explorateur de référentiel pour voir la structure du référentiel existante.

Maintenant vous devez choisir la source de la copie. Ici vous avez trois options :

#### Révision HEAD dans le référentiel

La nouvelle branche est copiée directement dans le référentiel de la révision HEAD. Aucune donnée n'a besoin d'être transférée depuis votre copie de travail et la branche est créée très rapidement.

#### Révision spécifique dans le référentiel

La nouvelle branche est copiée directement dans le référentiel mais vous pouvez choisir une révision plus vieille. C'est utile si vous avez oublié de faire une étiquette quand vous avez sorti votre projet la semaine dernière. Si vous ne pouvez pas vous rappeler le numéro de révision, cliquez sur le bouton à droite pour afficher le journal de révision et choisir le numéro de révision de là. Là encore, aucune donnée n'est transférée de votre copie de travail et la branche est créée très rapidement.

#### Copie de travail

La nouvelle branche est une copie identique à votre copie de travail locale. Si vous avez mis à jour quelques fichiers à une révision plus ancienne dans votre CdT, ou si vous avez fait des changements locaux, c'est exactement ceux-ci qui vont dans la copie. Naturellement cette sorte d'étiquette complexe peut impliquer des transferts de données de votre CdT vers le référentiel si elles n'y existent pas déjà.

Si vous voulez que votre copie de travail soit basculée automatiquement sur la branche nouvellement créée, utilisez la case à cocher **Déplacer la copie de travail vers une nouvelle branche/étiquette**. Mais si vous le faites, assurez-vous d'abord que votre copie de travail ne contient pas de modifications. Si c'est le cas, ces changements seront fusionnés dans la CdT de branche quand vous basculerez.

Appuyez sur **OK** pour livrer la nouvelle copie dans le référentiel. N'oubliez pas de mettre un commentaire. Notez que la copie est créée *dans le référentiel*.

Notez qu'à moins que vous choisissiez de basculer votre copie de travail vers la branche juste créée, créer une branche ou une étiquette *ne modifie pas* votre copie de travail. Même si vous créez la branche depuis votre CdT, ces changements sont livrés dans la nouvelle branche, pas dans le tronc, donc votre CdT peut toujours être marquée comme modifiée sans altérer le tronc.

### 4.19.2. Extraire ou aller sur...

... telle (n')est (pas) la question. Tandis qu'une extraction extraie tout de la branche désirée dans votre répertoire de travail, TortoiseSVN → **Aller sur...** transfère seulement les données changées dans votre copie de travail. Bon pour la charge réseau, bon pour votre patience. :-)

Pour pouvoir travailler avec votre branche ou votre tag récents générés, vous avez plusieurs méthodes. Vous pouvez :

- TortoiseSVN → **Extraire** pour faire une extraction récente dans un dossier vide. Vous pouvez extraire à n'importe quel emplacement sur votre disque local et vous pouvez créer autant de copies de travail de votre référentiel que vous souhaitez.
- Basculez votre copie de travail courante vers la copie nouvellement créée dans le référentiel. Choisissez de nouveau le dossier de niveau supérieur de votre projet et utilisez TortoiseSVN → **Aller sur...** du menu contextuel.

Dans la boîte de dialogue suivante, entrez l'URL de la branche que vous venez juste de créer. Choisissez le bouton radio **Révision HEAD** et cliquez sur **OK**. Votre copie de travail est basculée vers la nouvelle branche/étiquette.

Aller sur... fonctionne comme **Mettre à jour** dans le sens où il ne se débarrasse jamais de vos changements locaux. Les changements que vous avez faits à votre copie de travail qui n'ont pas encore été livrés seront fusionnés quand vous faites **Aller sur**. Si vous ne voulez pas que cela arrive alors vous devez ou livrer les changements avant la bascule, ou faire revenir votre copie de travail à une révision déjà livrée (typiquement HEAD).

- Si vous voulez travailler sur le tronc et une branche, mais sans faire une extraction, vous pouvez utiliser Windows Explorer pour copier votre extraction du tronc dans un autre répertoire, puis utilisez la commande TortoiseSVN → Aller sur... sur cette copie pour en faire la branche.



**Figure 4.34. La boîte de dialogue Aller sur**

Bien que Subversion lui-même ne fasse aucune distinction entre les étiquettes et les branches, la manière dont elles sont typiquement utilisées diffère un peu.

- Tags are typically used to create a static snapshot of the project at a particular stage. As such they not normally used for development - that's what branches are for, which is the reason we recommended the `/trunk /branches /tags` repository structure in the first place. Working on a tag revision is *not a good idea*, but because your local files are not write protected there is nothing to stop you doing this by mistake. However, if you try to commit to a path in the repository which contains `/tags/`, TortoiseSVN will warn you.
- Il peut arriver que vous deviez faire de nouveaux changements à une version déployée que vous avez déjà étiquetée. La façon correcte de le gérer est de créer d'abord une nouvelle branche à partir de l'étiquette et de livrer cette branche. Faites vos changements sur cette branche et créez ensuite une nouvelle étiquette depuis cette nouvelle branche, par exemple `Version_1.0.1`.
- Si vous modifiez une copie de travail créée à partir d'une branche et livrez, alors tous les changements seront livrés sur la nouvelle branche et *pas* sur le tronc. Seules les modifications sont stockées. Le reste reste une copie peu coûteuse.

## 4.20. Fusionner

Quand les branches sont utilisées pour maintenir des lignes séparées de développement, à une certaine étape vous voudrez fusionner les changements faits sur une branche vers le tronc, ou vice versa.

Il est important de comprendre comment les branches et la fusion fonctionnent dans Subversion avant de les utiliser, car cela peut devenir assez complexe. Il est fortement recommandé de lire le chapitre [Branching and Merging](http://svnbook.red-bean.com/en/1.5/svn.branchmerge.html) [http://svnbook.red-bean.com/en/1.5/svn.branchmerge.html] dans le manuel de Subversion, lequel donne une description complète et beaucoup d'exemples d'utilisation.

Il faut aussi noter que fusionner arrive *toujours* à l'intérieur d'une copie de travail. Si vous voulez fusionner les modifications *dans* une branche, vous devez avoir une copie de travail pour la branche extraite, et appeler l'assistant de fusion depuis cette copie de travail en utilisant TortoiseSVN → Fusionner....

En général, c'est une bonne idée d'exécuter une fusion dans une copie de travail inchangée. Si vous avez fait d'autres changements dans votre CdT, livrez les d'abord. Si la fusion ne se déroule pas comme prévu,

vous pouvez vouloir annuler les changements et la commande **Revenir en arrière** supprimera *tous* les changements en incluant ceux effectués avant la fusion.

Il y a trois cas d'utilisation de la fusion qui sont gérés de façons légèrement différentes, comme décrit ci-dessous. La première page de l'assistant de fusion vous demande de quelle méthode vous avez besoin.

#### Fusionner une plage de révisions

Cette méthode couvre le cas où vous avez fait une ou plusieurs révisions sur une branche (ou sur le tronc) et vous voulez reporter ces changements vers une autre branche.

Ce que vous demandez à Subversion est de : « Calculer les modifications nécessaires pour passer [DE] la révision 1 de la branche A [A] la révision 7 de la branche A, et d'appliquer ces modifications à la copie de travail (de la racine ou de la branche B). »

#### Réintégrer une branche

Cette méthode gère le cas où vous avez développé une nouvelle fonctionnalité dans une branche comme conseillé dans le manuel de Subversion. Semaine après semaine, tous les changements du tronc ont été reportés dans cette branche, et maintenant que la fonctionnalité est terminée, vous voulez l'intégrer dans le tronc. Vu que vous avez gardé la branche synchronisée avec le tronc, les dernières versions de la branche et du tronc seront absolument identiques à part vos modifications de la branche.

C'est un cas particulier de fusion de l'arborescence comme décrite plus haut, et (normalement) seule l'URL des sources à fusionner avec votre branche de développement est nécessaire. Cette fusion utilise la fonctionnalité de Subversion permettant de calculer la plage révision correcte à utiliser, et fait quelques autres tests pour s'assurer que la branche a été complètement synchronisée avec les évolutions du tronc. Cette technique empêche de défaire par erreur le travail que les autres auraient livré depuis votre dernière mise à jour.

Après la fusion, toutes les branches de développement ont été complètement fusionnées avec la version de tête. La branche est donc redondante et peut être supprimée.

A partir du moment où vous avez effectué une réintégration vous ne devriez pas continuer à travailler dans la branche. La raison est que si vous essayez plus tard de resynchroniser celle-ci au tronc, le système verra votre réintégration comme une modification du tronc qui n'a pas encore été fusionné avec la branche, et essaiera de le fusionner avec la branche ! La solution est simplement de créer une nouvelle branche depuis le tronc et continuer vers la prochaine phase de développement.

#### Fusionner deux arborescences différentes

C'est le cas le plus général de réintégration. Ce que vous demandez à Subversion est de « Calculer les modifications nécessaires pour aller [DE] la version de tête du tronc [A] la version de tête de la branche, et appliquer ces modifications à ma copie de travail (sur le tronc). ». Le résultat net est que le tronc est parfaitement identique à la branche.

Si votre serveur/référentiel ne supporte pas les fonctionnalités de suivi de fusion c'est alors le seul moyen de fusionner une branche avec le tronc. Vous pouvez aussi avoir besoin de cette méthode lorsque vous utilisez du code tiers et que vous devez fusionner les modifications d'un autre tiers dans votre code source. Lisez le chapitre suivant dans le livre de Subversion pour plus d'information : [Branches tiers](http://svnbook.red-bean.com/en/1.5/svn.advanced.vendorbr.html) [http://svnbook.red-bean.com/en/1.5/svn.advanced.vendorbr.html].

### 4.20.1. Fusionner une plage de révisions

Dans le champ **De** : entrez l'URL complète du dossier de la branche ou de l'étiquette contenant les changements que vous voulez reporter dans votre copie de travail. Vous pouvez aussi cliquer sur ... pour parcourir le référentiel et trouver la branche désirée. Si vous avez déjà fusionné depuis cette branche, alors utilisez simplement la liste déroulante contenant l'historique des URLs utilisées.

Dans le champ texte **Plage de révisions à fusionner** entrez la liste des révisions à fusionner. Ce peut être une seule révision, une liste de révisions spécifiques séparées par des virgules, une plage de révisions séparées par un tiret, ou une combinaison de ces différentes notations.



## Important

Dans TortoiseSVN, il y a une différence importante dans la manière dont est spécifiée une plage de révisions par rapport au client en ligne de commande.

Avec le client en ligne de commande vous spécifiez les modifications à fusionner avec deux révisions « limites » qui spécifient les bornes *avant* et *après*.

Dans TortoiseSVN vous spécifiez la liste des modifications à fusionner en utilisant des « panneaux de clôture/quote ». La raison de ceci devient plus clair lorsque vous utilisez la fenêtre de commentaire pour spécifier les révisions à fusionner, où chaque révision apparaît comme une liste de modifications. »

Si vous fusionnez les révisions par parties, la méthode exposée dans le livre de Subversion vous fera fusionner 100-200 cette fois ci et 200-300 la prochaine fois. Avec TortoiseSVN vous fusionneriez 100-200 cette fois et 201-300 la prochaine fois.

Cette différence a fait couler beaucoup d'encre dans les listes de diffusion. Nous reconnaissons les différences avec le client en ligne de commande, mais nous pensons qu'il est plus facile de comprendre la notation que nous avons implémenté pour la majorité des utilisateurs de l'interface.

La façon la plus facile de choisir la plage de révisions dont vous avez besoin est de cliquer sur **Voir le journal**, puisqu'ainsi les changements récents seront affichés avec les commentaires associés. Si vous voulez fusionner les changements d'une seule révision, sélectionnez juste cette révision. Si vous voulez fusionner les changements de plusieurs révisions, alors sélectionnez cette plage (en utilisant comme de coutume la touche **Maj**). Cliquez sur **OK** et les numéros de révision seront automatiquement insérés.

Si vous voulez *revenir* sur des modifications déjà livrées de votre copie de travail, sélectionnez les révisions à annuler et vérifiez bien que la case **Fusion inversée** est cochée.

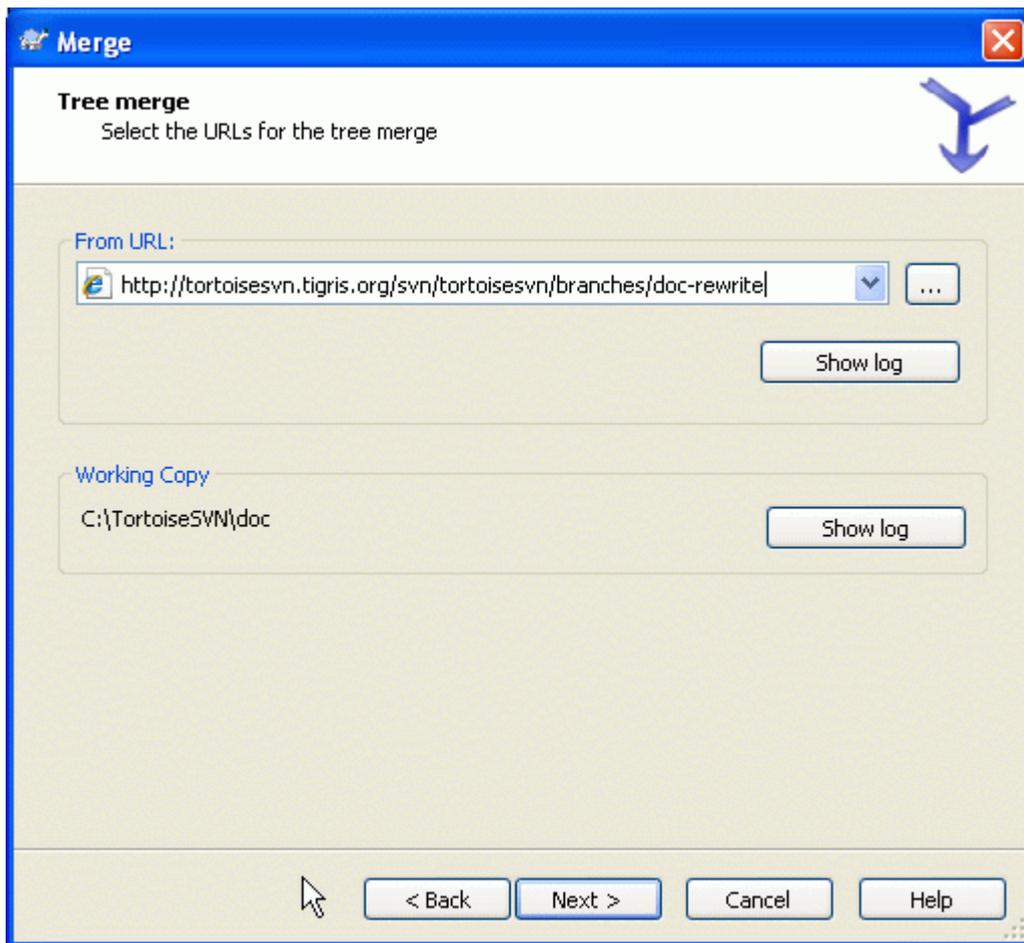
Si vous avez déjà fusionné des changements de cette branche, avec bon espoir vous aurez fait une note de la dernière révision fusionnée dans le commentaire quand vous avez livré la modification. Dans ce cas, vous pouvez utiliser **Voir le journal** dans la CdT pour retrouver ce commentaire. Dans la mesure où nous considérons les révisions comme des listes de modifications, vous devriez utiliser la révision de fin de la dernière fusion comme point de départ pour celle ci. Par exemple, si vous avez fusionné les révisions 37 à 39 la dernière fois, alors le point de départ pour cette fusion devrait être la révision 40.

Si vous vous servez de la fonctionnalité de suivi de fusion de Subversion, vous n'avez pas besoin de vous souvenir des révisions déjà fusionnées - Subversion les enregistrera pour vous. Si vous laissez le champ plage de révisions vide, toutes les révisions n'ayant pas déjà été fusionnées y seront notées. Lisez [Section 4.20.6, « Suivi des fusions »](#) pour en savoir plus.

Si d'autres personnes peuvent livrer des changements alors soyez prudents en utilisant de la révision HEAD. Elle peut ne pas faire référence à la révision à laquelle vous pensez si quelqu'un d'autre a fait une livraison après votre dernière mise à jour.

Cliquez le bouton **Suivant** et allez à [Section 4.20.4, « Options de fusion »](#)

### 4.20.2. Réintégrer une branche



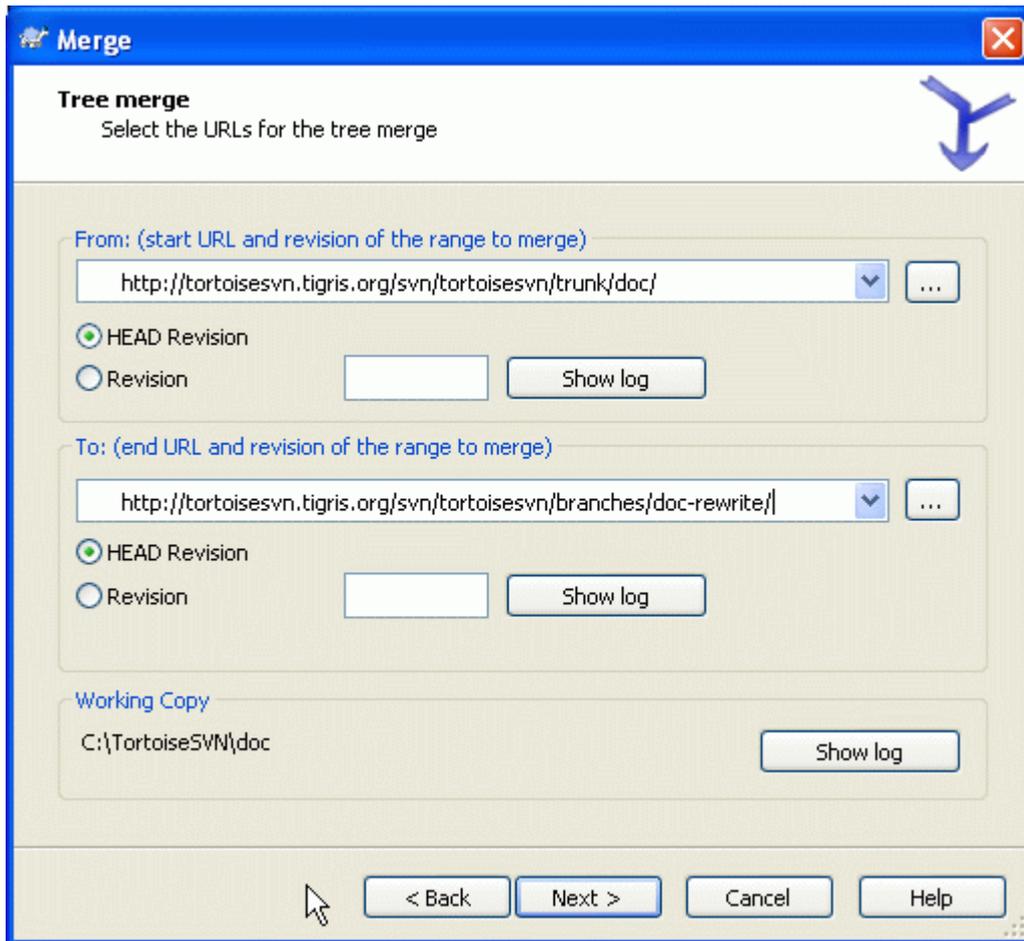
**Figure 4.35. Assistant de fusion - Réintégration**

Pour fusionner une branche de fonctionnalité dans le tronc vous devez démarrer l'assistant de fusion depuis une CdT du tronc.

Dans le champ De : entrez l'URL complète du dossier contenant la branche que vous voulez réintégrer. Vous pouvez également cliquer sur le bouton ... pour parcourir le référentiel.

Il y a quelques prérequis pour pouvoir faire une réintégration. Premièrement, le serveur doit supporter le suivi de fusion. La CdT doit contenir toute la profondeur complète de l'arborescence (pas d'extractions éparées), et elle ne doit avoir aucune modifications locales, d'éléments déplacés ou mis à jour à une version autre que la tête. Toutes les modifications faite sur le tronc pendant la phase de développement de la branche doivent avoir été intégrées à la branche (ou marquées comme ayant été fusionnées). La plage de révisions à fusionner sera calculée automatiquement.

#### 4.20.3. Fusionner deux arbres différents



**Figure 4.36. Assitant de de fusion - Fusion d'arborescence**

Si vous utilisez cette méthode pour fusionner une branche avec le tronc, vous devez démarrer l'assistant de fusion depuis une CdT du tronc.

Dans le champ **De** : entrez l'URL complète du dossier du *tronc*. Cela peut sembler erroné, mais souvenez-vous que le tronc est l'endroit auquel vous souhaitez ajouter les modifications de la branche. Vous pouvez aussi cliquer sur ... pour parcourir le référentiel.

Remplissez le champ **To**: par l'adresse complète de la branche.

Dans les deux champs **Depuis la révision** et **À la révision**, entrez le numéro de la dernière révision à laquelle les deux arbres ont été synchronisés. Si vous êtes sûrs que personne d'autre ne fait de livraison vous pouvez utiliser la révision **HEAD** dans les deux cas. S'il y a une chance que quelqu'un d'autre ait fait une livraison depuis cette synchronisation, utilisez le numéro de la révision particulier pour éviter de perdre des livraisons plus récentes.

Vous pouvez également utiliser **Voir les Messages de Log** pour sélectionner la révision.

#### 4.20.4. Options de fusion

Cette page de l'assistant vous permet d'accéder à des options avancées, avant de commencer le processus de fusion. La plupart du temps vous ne pouvez utiliser que les options par défaut.

Vous pouvez donner la profondeur de fusion, i.e. à quelle profondeur dans la CdT, la fusion doit aller. Les termes de profondeur utilisés sont décrits dans [Section 4.3.1, « Profondeur d'extraction »](#). La profondeur par défaut est **Copie de travail**, qui utilise la valeur existante de profondeur, et est presque toujours ce qui est nécessaire.

La plupart du temps vous voulez fusionner pour tenir compte de l'historique du fichier, de sorte que les changements relatifs à un ancêtre commun soient fusionnés. Parfois vous pouvez avoir besoin de fusionner des fichiers qui sont liés, mais pas à l'intérieur de votre référentiel. Par exemple vous pouvez avoir importé les versions 1 et 2 d'une bibliothèque tiers dans deux répertoires séparés. Bien qu'ils soient logiquement liés, Subversion n'en a aucune connaissance puisqu'il ne voit que le package que vous avez importé. Si vous essayez de fusionner ces deux arborescences vous verriez une suppression totale suivie d'un ajout total. Pour faire en sorte que Subversion n'utilise que les différences basées sur le chemin plutôt que les différences basées sur l'historique, cochez la case **Ignorer l'ascendance**. Plus d'informations à ce sujet dans le manuel de Subversion, *Noticing or Ignoring Ancestry* [<http://svnbook.red-bean.com/en/1.5/svn.branchmerge.advanced.html#svn.branchmerge.advanced.ancestry>]

Vous pouvez spécifier la gestion des caractères de fin de ligne et des espaces. Ces options sont décrites dans **Section 4.10.2, « Options de fins de ligne et d'espacement »**. Le comportement par défaut est de traiter tous les espaces et les fins de lignes comme étant des modifications à part entière.

Si vous utilisez la fonction de suivi de fusion et que vous souhaitez marquer une révision comme ayant été fusionnée, sans vraiment avoir fait de fusion ici, cochez la case **Enregistrer uniquement la fusion**. Il y a deux raisons pour lesquelles vous pourriez être amenés à faire ceci. La fusion à effectuer est trop compliquée pour les algorithmes de fusion, vous faites donc la fusion à la main et marquez les modifications comme ayant été fusionnées de manière à ce que le suivi de fusion le sache. Ou vous voulez éviter qu'une révision particulière soit fusionnée. La marquer comme ayant été fusionnée empêchera les clients supportant le suivi de fusion de faire effectivement la fusion.

Now everything is set up, all you have to do is click on the **Merge** button. If you want to preview the results **Test Merge** performs the merge operation, but does *not* modify the working copy at all. It shows you a list of the files that will be changed by a real merge, and notes those areas where conflicts will occur.

La barre de progression de la fusion montre chaque étape du processus, avec les plages de révision concernées. Cela peut indiquer une révision de plus que prévu. Par exemple, si vous demandez de fusionner la révision 123 la barre de progression affichera « Fusion des révisions 122 à 123 ». Pour comprendre ce qui se passe, vous devez garder en mémoire que l'opération de fusion est très proche du diff. La fusion génère une liste de différences entre deux endroits du référentiel, et applique ces différences à votre CdT. La barre de progression montre simplement les points de départ et d'arrivée du diff.

#### 4.20.5. Prévisualiser les résultats de la fusion

La fusion est à présent effectuée. C'est une bonne idée le résultat et vérifier que tout est conforme à vos attentes. Fusionner est normalement assez compliqué. Il y a souvent des conflits si la branche s'est beaucoup éloignée du tronc.

Pour les clients et serveurs Subversion antérieurs à la version 1.5, aucune information de fusion n'est gardée et les révisions à fusionner sont à trouver manuellement. Lorsque vous avez testé et que vous allez livrer, le commentaire de livraison devrait *toujours* inclure les numéros de révision inclus dans la fusion. Si vous voulez faire une autre fusion plus tard vous aurez besoin de savoir ce que vous avez déjà fusionné pour ne pas appliquer vos modifications plusieurs fois. Pour plus d'informations, consultez *Best Practices for Merging* [<http://svnbook.red-bean.com/en/1.4/svn.branchmerge.copychanges.html#svn.branchmerge.copychanges.bestprac>] dans la bible de Subversion.

Pour des clients et serveurs Subversion postérieurs à la version 1.5, la fonctionnalité de suivi de fusion enregistrera les révisions fusionnées et empêchera d'en fusionner une plus d'une fois. Cela vous simplifie la tâche dans la mesure où vous pouvez tout fusionner à chaque fois en étant sûr que seules les nouvelles révisions seront vraiment fusionnées.

La gestion de branches est importante. Si vous voulez conserver une branche synchronisée avec le tronc, vous devrez vous assurer de fusionner souvent de sorte que la branche et le tronc ne soient pas trop éloignés. Bien sûr, vous devez toujours éviter de fusionner plusieurs fois les mêmes modifications comme expliqué ci-dessus.



### Astuce

Si vous venez de fusionner une branche avec le tronc, celui ci contient le code de la nouvelle fonctionnalité, et la branche est obsolète. Vous pouvez donc la supprimer du référentiel si besoin est.



### Important

Subversion ne peut pas fusionner un fichier avec un dossier et vice versa - seulement entre dossiers et entre fichiers. Si vous cliquez sur un fichier et ouvrez la boîte de dialogue fusionner, vous devez alors donner un chemin vers un fichier dans cette boîte de dialogue. Si vous choisissez un dossier et affichez la boîte de dialogue, vous devez alors spécifier une URL de dossier pour la fusion.

## 4.20.6. Suivi des fusions

A partir de la version 1.5, Subversion inclus des aides au suivi de fusion. Lorsque vous fusionnez des modifications depuis une arborescence vers une autre, les numéros de révision sont stockés et peuvent être utilisés à plusieurs fins.

- Vous pouvez éviter le danger de fusionner deux fois la même révision (problème de fusions répétées). Dès qu'une révision a été marquée comme ayant été fusionnée, les fusions futures incluant cette révision dans leur plage de révisions l'ignoreront.
- Quand vous fusionnez une branche dans le tronc, la fenêtre de commentaires peut vous montrer les livraisons de la branche comme faisant partie des commentaires du tronc, donnant une meilleure traçabilité des modifications.
- Lorsque vous affichez la fenêtre de commentaires depuis la fenêtre de fusion, les révisions déjà fusionnées sont grisées.
- Quand l'annotation d'un fichier est affichée, vous pouvez choisir de montrer les auteurs des révisions fusionnées, au lieu de la personne ayant fait la fusion.
- Vous pouvez marquer les révisions comme n'étant *pas à fusionner* en les incluant dans la liste des révisions fusionnées mais sans vraiment faire la fusion.

Les informations de suivi de fusion sont stockées dans la propriété `svn:mergeinfo` par le client lorsqu'il effectue la fusion. Au moment de la livraison de cette fusion, le serveur stocke l'information dans la base de données, et quand vous souhaitez fusionner, commenter ou annoter, le serveur peut répondre de manière appropriée. Afin que le système fonctionne correctement, vous devez vous assurer que le serveur, le référentiel et les clients sont à jour. Les clients plus anciens ne stockeront pas la propriété `svn:mergeinfo` et les serveurs plus anciens ne pourront pas satisfaire à toutes les demandes des clients récents.

Apprenez en plus à propos du suivi de fusion dans la bible Subversion [Merge tracking documentation](http://subversion.tigris.org/merge-tracking/index.html) [http://subversion.tigris.org/merge-tracking/index.html].

## 4.20.7. Gérer les conflits durant la fusion.

La fusion ne se passe pas toujours très bien. Parfois, il y a des conflits, et si vous fusionnez plusieurs plages de révisions, vous voudrez généralement résoudre les conflits avant de passer à la plage de révisions suivante. TortoiseSVN vous aide dans ce processus en affichant la boîte de dialogue *de conflit de fusion*.



**Figure 4.37. La boîte de dialogue de conflit de fusion**

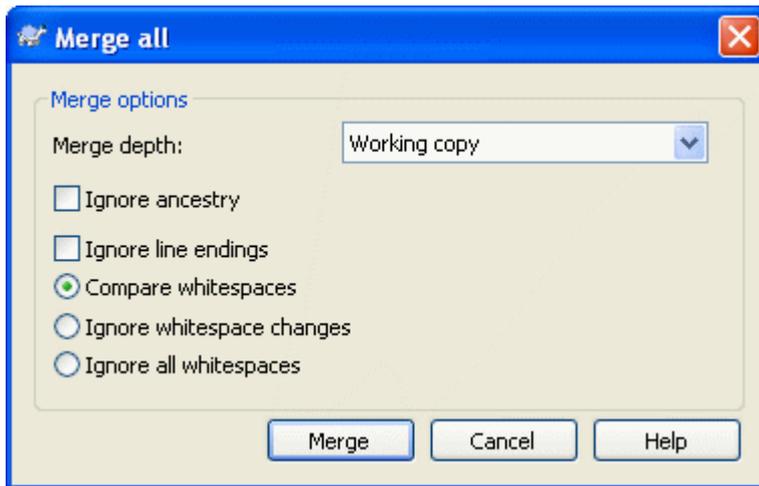
Quand la fusion génère un conflit, vous avez trois moyens de le gérer :

1. You may decide that your local changes are much more important, so you want to discard the version from the repository and keep your local version. Or you might discard your local changes in favour of the repository version. Either way, no attempt is made to merge the changes - you choose one or the other.
2. Normally you will want to look at the conflicts and resolve them. In that case, choose the **Edit Conflict** which will start up your merge tool. When you are satisfied with the result, click **Resolved**.
3. La dernière option est de reporter la résolution et continue l'opération de fusion. Vous pouvez choisir cette option pour le conflit courant, ou pour tous les fichiers conflictuels de la fusion. De toutes façons, s'il y a d'autres modifications dans ce fichier, la fusion ne pourra pas se faire.

Si vous ne voulez pas utiliser le mode interactif, il y a une case à cocher dans la barre de progression de la fusion **Fusion non-interactive**. Si celle-ci est cochée pour une fusion, et que la fusion génère des conflits, le(s) fichier(s) sont marqués comme conflictueux et l'opération de fusion continue. Vous aurez à résoudre les conflits lorsque la fusion sera terminée. Si la case n'est pas cochée, vous avez la possibilité de résoudre le conflit avant que le fichier soit marqué comme conflictuel *pendant* la fusion. L'avantage est que si un fichier est impliqué dans plusieurs fusions (plusieurs révisions concernent celui ci), les fusions successives pourront se passer sans encombre selon les lignes affectées. Mais bien sûr vous ne pouvez pas aller prendre un café pendant que la fusion s'exécute ;)

#### 4.20.8. Fusionner une branche complétée

Si vous voulez fusionner toutes les modifications d'une branche de fonctionnalité dans le tronc, vous pouvez utiliser la commande TortoiseSVN → Fusion de réintégration... depuis le menu contextuel (appuyez sur la touche **Maj** en faisant un click droit sur le fichier).



**Figure 4.38. La boîte de dialogue de fusion**

Cette fenêtre est très simple. Tout ce que vous avez à faire est de paramétrer la fusion, comme décrit dans [Section 4.20.4, « Options de fusion »](#). Le reste est fait automatiquement par TortoiseSVN grâce au suivi de fusion.

#### 4.20.9. Branche de maintenance d'une fonctionnalité

Quand vous développez une nouvelle fonctionnalité dans une branche séparée c'est une bonne pratique de garder en tête la réintégration de cette branche lorsque la nouvelle fonctionnalité sera prête. Si la branche principale (le `trunk`) évolue en même temps, il est probable que les différences seront nombreuses, rendant l'intégration cauchemardesque.

Si la fonctionnalité est relativement simple et que le développement ne prend pas beaucoup de temps, vous pouvez adopter une solution simple, qui consiste à garder une branche distincte, que vous intégrerez lorsque le développement sera fini. Dans l'assistant de fusion, ce devrait n'être qu'un simple Fusionner une plage de révisions, la plage de révisions couvrant la période de développement de la branche.

Dans le cas où la fonctionnalité prend du temps et que vous souhaitez récupérer les évolutions du `trunk`, alors vous devez garder votre branche synchronisée. Cela signifie simplement que vous devez fusionner de temps en temps les évolutions du tronc dans votre branche, de manière à ce que votre branche contiennent toutes les modifications du tronc *en plus* de la nouvelle fonctionnalité. Le processus de synchronisation utilise la commande Fusionner une plage de révisions. Quand le développement de la fonctionnalité est achevé vous pouvez alors fusionner la branche avec le tronc (i.e. `trunk`) en utilisant Réintégrer une branche ou Fusionner deux arborescences.

### 4.21. Verrouiller

Subversion fonctionne généralement mieux sans verrouillage, en utilisant les méthodes « Copier-Modifier-Fusionner » décrites précédemment dans [Section 2.2.3, « La solution Copier-Modifier-Fusionner »](#). Cependant il y a quelques cas où vous pouvez devoir appliquer une certaine forme de politique de verrouillage.

- Vous utilisez des fichiers « non fusionnables », par exemple, des fichiers graphiques. Si deux personnes changent le même fichier, la fusion n'est pas possible, donc l'une d'entre elles perdra ses modifications.
- Votre société a toujours utilisé un VCS verrouillant par le passé et il y a eu une décision du management stipulant que « le verrouillage est la meilleure solution ».

Premièrement, vous devez vous assurer que votre serveur Subversion est mis à jour au moins à la version 1.2. Les versions antérieures ne supportent pas du tout le verrouillage. Si vous utilisez un accès de type `file://`, alors bien sûr seul votre client doit être mis à jour.

### 4.21.1. Comment le verrouillage fonctionne dans Subversion

Par défaut, rien n'est verrouillé et quiconque a un accès en livraison peut livrer des modifications de n'importe quel fichier à tout moment. Les autres mettront à jour leurs copies de travail périodiquement et les modifications du référentiel seront fusionnées avec les modifications locales.

Si vous *obtenez un verrou* sur un fichier, alors vous seul pouvez livrer ce fichier. Les livraisons des autres utilisateurs seront bloquées jusqu'à ce que vous retiriez le verrou. Un fichier verrouillé ne peut être modifié d'aucune façon dans le référentiel, il ne peut donc pas non plus être supprimé ou renommé, sauf par le propriétaire du verrou.

Toutefois, les autres utilisateurs ne sauront pas nécessairement que vous avez posé un verrou. À moins qu'ils ne vérifient le verrouillage régulièrement, ils ne le découvriront que lorsque leur livraison échouera, ce qui n'est pas très utile dans la plupart des cas. Pour rendre la gestion des verrous plus facile, il existe une nouvelle propriété Subversion `svn:needs-lock`. Quand cette propriété est définie (avec n'importe quelle valeur) sur un fichier, chaque fois que le fichier est extrait ou mis à jour, la copie locale est mise en lecture seule à *moins que* cette copie de travail ne détienne un verrou pour le fichier. Cela agit comme un avertissement de ne pas éditer ce fichier à moins que vous n'ayez d'abord posé un verrou. Les fichiers qui sont versionnés et en lecture seule sont identifiés par une coloration spéciale dans TortoiseSVN, pour indiquer que vous devez poser un verrou avant l'édition.

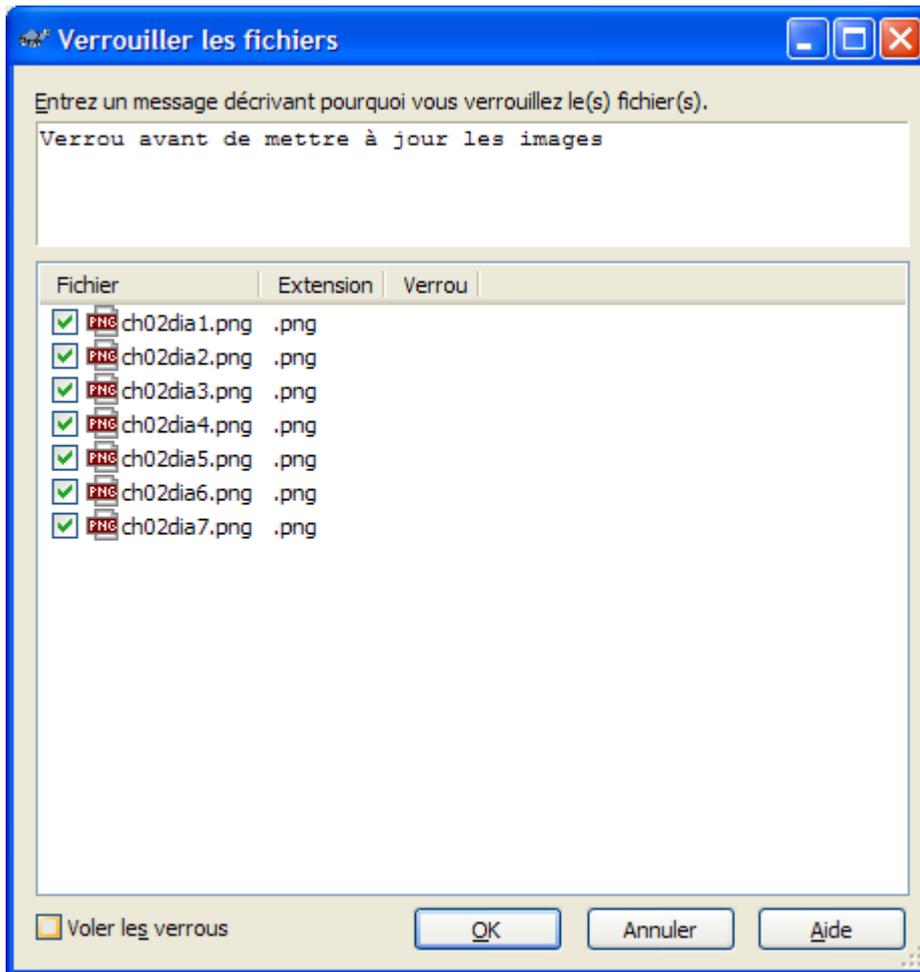
Les verrous sont enregistrés par emplacement de copie de travail et par propriétaire. Si vous avez plusieurs copies de travail (à la maison, au travail) alors vous ne pouvez détenir un verrou que dans *une* de ces copies de travail.

Si l'un de vos collègues pose un verrou et part ensuite en vacances sans le retirer, que faites-vous ? Subversion fournit un moyen de forcer les verrous. Retirer un verrou détenu par quelqu'un d'autre est décrit comme *Casser le verrou* et prendre de force un verrou déjà détenu par quelqu'un d'autre s'appelle *Voler le verrou*. Naturellement, ce ne sont pas des choses que vous devriez faire à la légère, si vous voulez rester en bons termes avec vos collègues.

Les verrous sont enregistrés dans le référentiel et un jeton de verrouillage est créé dans votre copie de travail locale. S'il y a une incohérence, par exemple si quelqu'un d'autre a cassé le verrou, le jeton de verrouillage local devient invalide. Le référentiel est toujours la référence absolue.

### 4.21.2. Obtenir un verrou

Sélectionnez les fichiers de votre copie de travail pour lesquels vous voulez un verrou, puis sélectionnez la commande TortoiseSVN → Obtenir un verrou....



**Figure 4.39. La boîte de dialogue Verrouiller**

Une boîte de dialogue apparaît, vous permettant de saisir un commentaire, pour que les autres puissent voir pourquoi vous avez verrouillé le fichier. Le commentaire est facultatif et actuellement utilisé avec les seuls référentiels basés sur Svnserve. Si (et *seulement* si ) vous deviez voler le verrou de quelqu'un d'autre, cochez la case **Voler les verrous**, cliquez ensuite sur **OK**.

Si vous sélectionnez un dossier et utilisez ensuite TortoiseSVN → Obtenir un verrou... la boîte de dialogue Verrouiller s'ouvrira avec *tous* les fichiers de *tous* les sous-dossiers choisis pour les verrouiller. C'est la bonne manière de faire si vous voulez vraiment verrouiller une arborescence complète, cependant vous pourriez devenir très impopulaire auprès de vos collègues si vous les empêchez d'accéder au projet. À utiliser avec parcimonie...

### 4.21.3. Retirer un verrou

Pour vous assurer de ne pas oublier de retirer un verrou dont vous n'avez plus besoin, les fichiers verrouillés sont affichés dans la boîte de dialogue de livraison et sélectionnés par défaut. Si vous continuez la livraison, les verrous que vous détenez sur les fichiers sélectionnés sont supprimés, même si les fichiers n'ont pas été modifiés. Si vous ne voulez pas retirer les verrous de certains fichiers, vous pouvez les décocher (s'ils ne sont pas modifiés). Si vous voulez garder un verrou sur un fichier que vous avez modifié, vous devez activer la case à cocher **Garder les verrous** avant de livrer vos changements.

Pour retirer un verrou manuellement, sélectionnez les fichiers de votre copie de travail dont vous voulez retirer les verrous, choisissez ensuite la commande TortoiseSVN → Retirer un verrou Il n'y a rien d'autre à préciser, et TortoiseSVN va communiquer avec le référentiel et retirer les verrous. Vous pouvez aussi utiliser cette commande sur un dossier pour retirer tous les verrous récursivement.

#### 4.21.4. Vérifier le statut des verrous

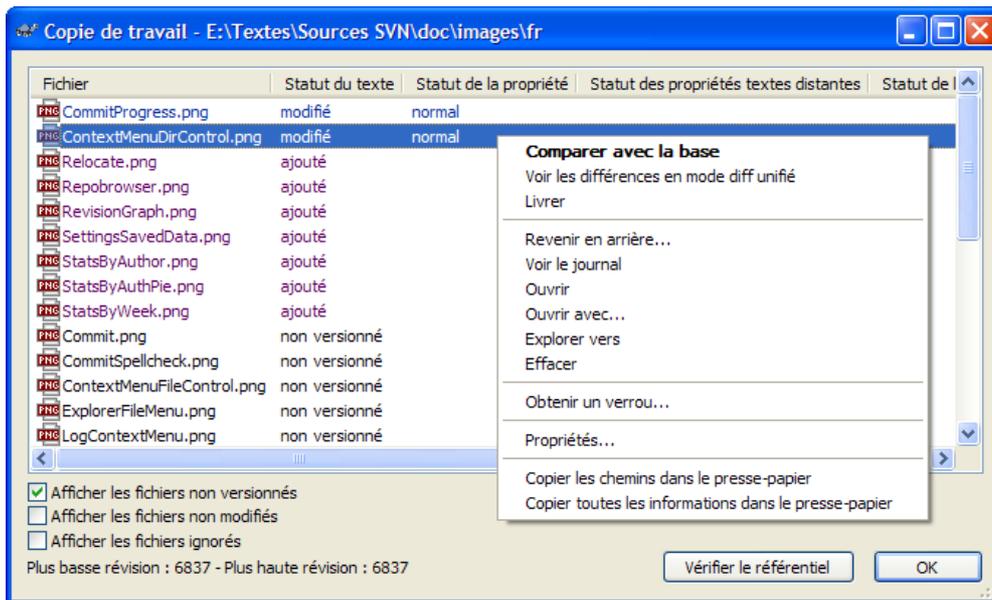


Figure 4.40. La boîte de dialogue Vérifier les modifications

Pour voir quels verrous les autres et vous détenez, vous pouvez utiliser TortoiseSVN → Vérifier les modifications.... Les jetons de verrouillage détenus localement s'afficheront immédiatement. Pour vérifier les verrous détenus par les autres (et voir si l'un de vos verrous est cassé ou volé) vous devez cliquer sur Vérifier le référentiel.

À partir du menu contextuel accessible ici, vous pouvez aussi bien obtenir et retirer des verrous, que casser et voler des verrous détenus par d'autres.



#### Évitez de casser et de voler les verrous

Si vous cassez ou volez le verrou de quelqu'un d'autre sans le lui dire, vous pourriez potentiellement causer une perte de travail. Si vous travaillez avec des types de fichier non fusionnables et que vous volez le verrou de quelqu'un d'autre, une fois que vous retirez le verrou, il est libre de livrer ses modifications et d'écraser les vôtres. Subversion ne perd pas de données, mais vous avez perdu la protection de travail d'équipe que le verrouillage vous avait donnée.

#### 4.21.5. Mettre les fichiers non verrouillés en Lecture seule

Comme mentionné ci-dessus, la façon la plus efficace d'utiliser le verrouillage est de mettre la propriété `svn:needs-lock` sur les fichiers. Référez-vous à [Section 4.17, « Configuration des projets »](#) pour les instructions sur le mode d'activation des propriétés. Les fichiers avec cette propriété activée seront toujours extraits et mis à jour avec l'attribut Lecture seule activé à moins que votre copie de travail ne détienne un verrou.



Pour vous le rappeler, TortoiseSVN utilise une coloration spéciale pour l'indiquer.

Si vous appliquez une politique où tout fichier doit être verrouillé, vous pouvez trouver alors plus facile d'utiliser la fonctionnalité auto-props de Subversion pour activer la propriété automatiquement à chaque fois vous ajoutez de nouveaux fichiers. Lisez [Section 4.30, « Configuration de TortoiseSVN »](#) pour plus d'informations.

### 4.21.6. Les scripts hook de verrouillage

Quand vous créez un nouveau référentiel avec Subversion 1.2 ou supérieur, quatre modèles de hooks sont créés dans le répertoire `hooks` du référentiel. Ceux-ci sont appelés avant et après l'obtention d'un verrou et avant et après le retrait d'un verrou.

C'est une bonne idée d'installer un script hook `post-lock` et `post-unlock` sur le serveur qui envoie un e-mail indiquant que le fichier a été verrouillé. Avec un tel script d'installé, tous vos utilisateurs peuvent être informés si quelqu'un verrouille/déverrouille un fichier. Vous pouvez trouver un script hook d'exemple `hooks/post-lock.tmpl` dans le dossier de votre référentiel.

Vous pourriez aussi utiliser des hooks pour ne pas permettre la casse ou le vol de verrous, ou peut-être les limiter à un administrateur nommé. Ou peut-être vous voulez envoyer un email au propriétaire quand l'un de ses verrous est cassé ou volé.

Lisez [Section 3.3, « Scripts de hook côté serveur »](#) pour en savoir plus.

## 4.22. Créer et appliquer des patches

Pour les projets open-source (comme celui-ci), tout le monde a accès au référentiel en lecture et tout le monde peut contribuer au projet. Alors comment ces contributions sont-elles contrôlées ? Si tout le monde pouvait livrer des changements, le projet serait instable en permanence et probablement cassé en permanence. Dans cette situation, le changement est géré en soumettant un fichier *patch* à l'équipe de développement, qui a accès en écriture. Ils peuvent passer en revue le patch d'abord et ensuite le soumettre au référentiel ou le renvoyer à l'auteur.

Les fichiers patch sont simplement des fichiers de différences unifiées montrant les différences entre votre copie de travail et la révision de base.

### 4.22.1. Créer un patch

D'abord vous devez faire *et tester* vos changements. Alors au lieu d'utiliser TortoiseSVN → Livrer... sur le dossier parent, vous sélectionnez TortoiseSVN → Créer un patch...

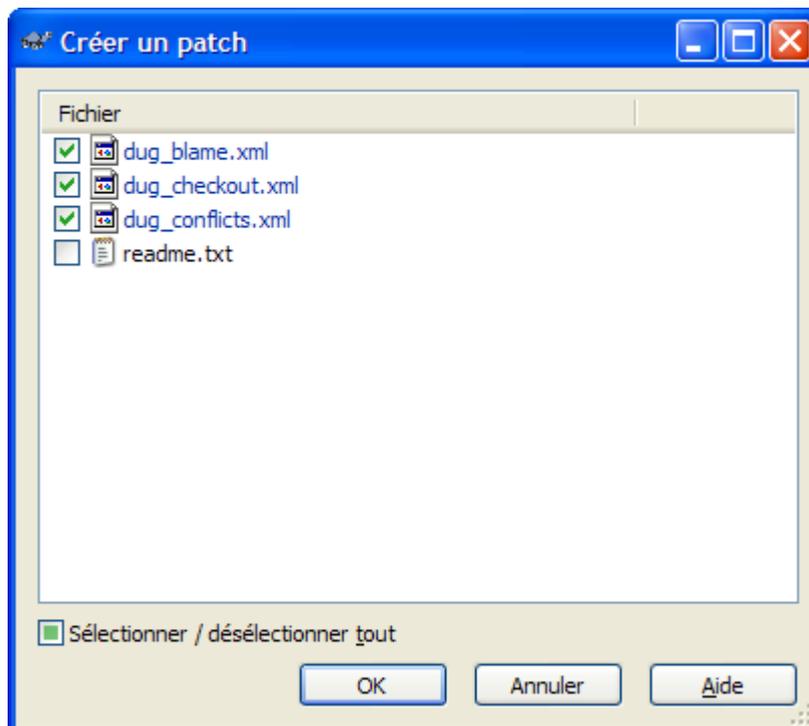


Figure 4.41. La boîte de dialogue Créer un patch

Vous pouvez maintenant choisir les fichiers que vous voulez inclure dans le patch, comme vous le feriez avec une livraison complète. Cela produira un unique fichier contenant un résumé de tous les changements que vous avez faits aux fichiers sélectionnés depuis la dernière mise à jour du référentiel.

Les colonnes dans cette boîte de dialogue peuvent être personnalisées de la même manière que les colonnes dans la boîte de dialogue **Vérifier les modifications**. Lisez [Section 4.7.3, « Statut local et distant »](#) pour plus de détails.

Vous pouvez produire des patches séparés contenant des changements sur des jeux différents de fichiers. Bien sûr, si vous créez un fichier patch, faire un peu plus de modifications aux *mêmes* fichiers et créez ensuite un autre patch, le deuxième fichier patch inclura les *deux* jeux de changements.

Sauvegardez juste le fichier en utilisant un nom de fichier de votre choix. Les patches peuvent avoir l'extension de votre choix, mais selon la convention ils devraient utiliser les extensions `.patch` ou `.diff`. Vous êtes maintenant prêt à soumettre votre patch.

Vous pouvez aussi sauvegarder le patch dans le presse-papier plutôt que dans un fichier. Cela permet, par exemple, de le coller dans un mail afin de le transmettre à d'autres personnes pour une revue. Ou alors, si vous avez deux copies de travail sur une machine et que vous vouliez reporter des changements de l'une à l'autre, copier le patch dans le presse-papier est un moyen commode pour le faire.

### 4.22.2. Appliquer un patch

Les fichiers patch s'appliquent à votre copie de travail. Cela doit être fait au même niveau de dossier qui a été utilisé pour créer le patch. Si vous n'êtes pas sûr de le savoir, regardez simplement la première ligne du fichier patch. Par exemple, si le premier fichier travaillé était `doc/source/english/chapter1.xml` et la première ligne dans le patch est `Index: english/chapter1.xml` alors vous devez appliquer le patch sur le répertoire `english`. Cependant, pour le cas où vous êtes dans la bonne copie de travail, si vous choisissez le mauvais niveau de dossier, TSVN le remarquera et suggérera le niveau correct.

Pour appliquer un patch à votre copie de travail, vous devez avoir au moins l'accès en lecture pour le référentiel. La raison à cela est que le programme de fusion doit faire référence aux changements de la révision avec laquelle ils ont été faits par le développeur distant.

From the context menu for that folder, click on TortoiseSVN → Apply Patch... This will bring up a file open dialog allowing you to select the patch file to apply. By default only `.patch` or `.diff` files are shown, but you can opt for « All files ». If you previously saved a patch to the clipboard, you can use `Open from clipboard...` in the file open dialog.

Alternativement, si le fichier patch a une extension `.patch` ou `.diff`, vous pouvez faire un clic droit dessus directement et sélectionnez `TortoiseSVN → Appliquer un patch...` Dans ce cas, vous serez invité à entrer un emplacement de copie de travail.

Ces deux méthodes offrent juste des façons différentes de faire la même chose. Avec la première méthode, vous choisissez la CdT et naviguez jusqu'au patch. Avec la deuxième, vous choisissez le patch et naviguez jusqu'à la CdT.

Une fois que vous avez sélectionné le fichier patch et l'emplacement de la copie de travail, TortoiseMerge se lance pour fusionner les changements du fichier patch avec votre copie de travail. Une petite fenêtre liste les fichiers qui ont été changés. Double-cliquez sur chacun à son tour, passez en revue les changements et sauvegardez les fichiers fusionnés.

Le patch du développeur distant a maintenant été appliqué à votre copie de travail, donc vous devez livrer pour permettre à tous les autres d'avoir accès aux changements depuis le référentiel.

### 4.23. Qui a changé quelle ligne ?

Parfois, vous avez besoin de connaître non seulement quelles lignes ont changé, mais aussi exactement qui a changé des lignes spécifiques dans un fichier. C'est à ce moment que la commande TortoiseSVN → Annoter..., parfois aussi mentionnée comme la commande *annoter* devient pratique.

Cette commande liste, pour chaque ligne d'un fichier, l'auteur et la révision où la ligne a été changée.

#### 4.23.1. Annoter pour les fichiers

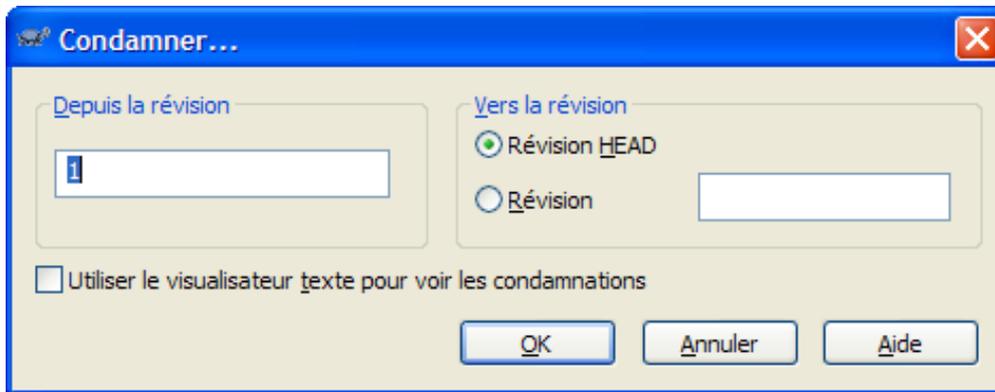


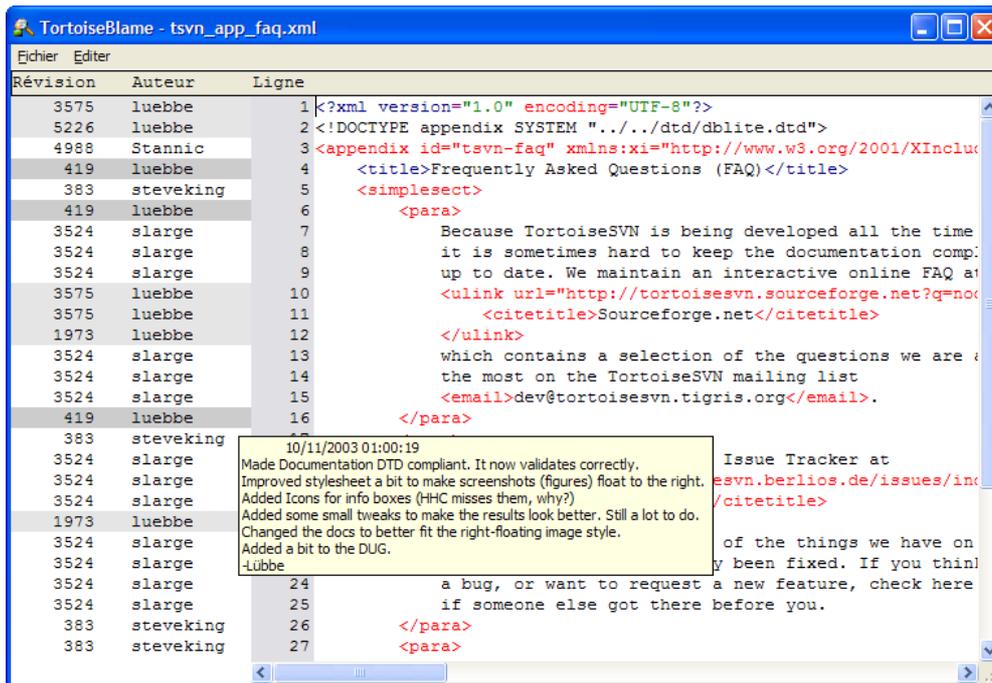
Figure 4.42. La boîte de dialogue Annoter

Si vous n'êtes pas intéressés par les changements des révisions précédentes, vous pouvez définir la révision d'où l'annotation devrait commencer. Mettez cela à 1, si vous voulez l'annotation pour *toutes* les révisions.

Par défaut le fichier d'annotations est visualisé en utilisant *TortoiseBlame*, qui met en évidence les différentes révisions pour le rendre plus facile à lire. Si vous voulez imprimer ou éditer le fichier d'annotations, sélectionnez *Utiliser le visualisateur texte pour voir les annotations*

Vous pouvez indiquer la manière de gérer les changements de caractères de fin de ligne et d'espacement. Ces options sont décrites dans [Section 4.10.2, « Options de fins de ligne et d'espacement »](#). Par défaut, toutes les modifications de caractères de fin de ligne et d'espacement sont considérés comme des changements à part entière, ceci dit si vous souhaitez ignorer les changements d'indentation et trouver l'auteur original, vous pouvez appliquer l'option appropriée ici.

Une fois que vous cliquez sur OK TortoiseSVN commence à récupérer les données pour créer le fichier d'annotation. Veuillez noter : Cela peut prendre plusieurs minutes, selon le nombre de fois que le fichier a été modifié et bien sûr, selon votre connexion réseau au référentiel. Une fois que le processus d'annotation est terminé, le résultat est écrit dans un fichier temporaire et vous pouvez voir les résultats.



**Figure 4.43. TortoiseBlame**

TortoiseBlame, qui est inclus avec TortoiseSVN, rend le fichier d'annotation plus facile à lire. Quand vous survolez avec la souris une ligne dans la colonne de renseignements de l'annotation, toutes les lignes avec la même révision sont affichées avec un fond plus sombre. Les lignes d'autres révisions qui ont été changées par le même auteur sont affichées avec un fond clair. La coloration peut ne pas fonctionner aussi clairement si vous avez votre affichage en mode 256 couleurs.

Si vous faites un clic gauche sur une ligne, toutes les lignes avec la même révision sont mises en évidence et les lignes d'autres révisions du même auteur sont mises en évidence dans une couleur plus claire. Cette accentuation est figée, vous permettant de déplacer la souris sans perdre les points en évidence. Cliquez sur cette révision de nouveau pour arrêter l'accentuation.

Les commentaires de révision (commentaire du journal) sont affichés dans une info-bulle à chaque fois que la souris survole la colonne d'information de l'annotation. Si vous voulez copier le commentaire de cette révision, utilisez le menu contextuel qui apparaît quand vous faites un clic droit sur la colonne d'information de l'annotation.

Vous pouvez faire des recherches dans le rapport d'annotation en utilisant **Éditer** → **Rechercher...** Cela vous permet de chercher des numéros de révision, des auteurs et le contenu du fichier lui-même. Les commentaires ne sont pas inclus dans la recherche - vous devriez utiliser la boîte de dialogue de Journal pour la recherche.

Vous pouvez aussi aller à une ligne particulière en utilisant **Editer** → **Aller à la ligne...**

Lorsque la souris survole la colonne annotation, un menu contextuel s'affiche vous permettant de comparer les révision et de consulter l'historique, en prenant comme référence le numéro de ligne sous le curseur de la souris. **Menu Contextuel** → **Annoter la révision précédente** crée une annotation sur ce même fichier, mais en utilisant la révision précédente comme limite supérieure. Ce qui donne les annotations du fichier juste avant la ligne que vous survolez tel qu'il était avant les dernières modifications. **Menu Contextuel** → **Montrer les modifications** démarre votre visionneuse de différences, vos montrant ce qui a changé à la révision référencée. **Menu Contextuel** → **Montrer les commentaires** affiche la fenêtre de commentaire des révisions, commençant par la révision référencée.

Si vous souhaitez un meilleur indicateur concernant les modifications les plus récentes et les plus anciennes, Voir → **Colorer l'âge des lignes**. Cette option utilisera un gradient de couleur pour afficher les nouvelles lignes en rouge et les plus anciennes en bleu. La coloration par défaut est assez légère, mais vous pouvez la modifier en utilisant les préférences de TortoiseBlame.

Si vous utilisez le traqueur de fusion, où les modifications des lignes sont le résultat d'une fusion depuis un autre chemin, TortoiseBlame affichera la révision et l'auteur de la dernière modification dans le fichier original au lieu de la révision où la fusion a été effectuée. La révision et l'auteur de ces lignes sont affichés en italique. Si vous ne voulez pas de cette notation, décochez la case **Inclure les informations de fusion**.

Si vous souhaitez voir les chemins concernés dans la fusion, sélectionnez Voir → **Chemins Fusionnés**.

Les paramètres de TortoiseBlame peuvent être affichés en sélectionnant TortoiseSVN → **Paramètres...** dans l'onglet TortoiseBlame. Voir [Section 4.30.9, « Configuration de TortoiseBlame »](#).

### 4.23.2. Annoter les différences

L'une des limitations du rapport d'annotation est qu'il ne montre que le fichier tel qu'il était à une révision particulière et montre la dernière personne à avoir changé chaque ligne. Parfois vous voulez connaître quel changement a été fait, et aussi qui l'a fait. Ce dont vous avez ici besoin est une combinaison des rapports d'annotation et de comparaison.

La boîte de dialogue du journal de révision inclut plusieurs options vous permettant de faire cela.

#### Annoter les révisions

Dans le panneau supérieur, sélectionnez 2 révisions, puis sélectionnez **Menu contextuel → Annoter les révisions**. Cela parcourra les annotations pour les 2 révisions puis utilisera le visualisateur de différences pour comparer les deux fichiers d'annotation.

#### Bannir les modifications

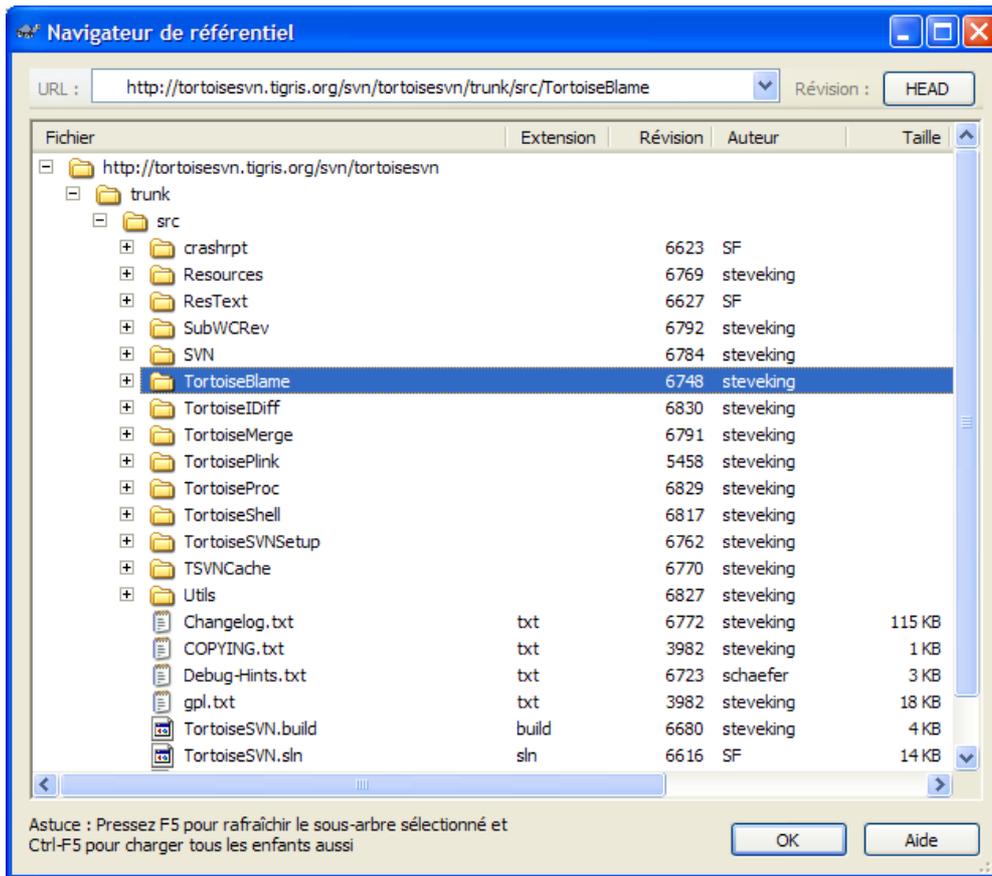
Sélectionnez une révision dans le panneau supérieur, puis choisissez un fichier dans le panneau inférieur et sélectionnez **Menu contextuel → Annoter les différences**. Cela ira chercher les annotations pour la révision sélectionnée et la révision précédente, puis utilisera le visualisateur de différences pour comparer les deux fichiers d'annotations.

#### Comparez et annoter avec la BASE de travail

Affichez le journal pour un seul fichier et, dans le panneau supérieur, choisissez une seule révision, puis sélectionnez **Menu contextuel → Comparer et annoter avec la BASE de travail**. Cela ira chercher les annotations pour la révision choisie et pour le fichier dans la BASE de travail, puis utilisera ensuite le visualisateur de différences pour comparer les deux fichiers d'annotations.

## 4.24. l'explorateur de référentiel

Parfois, vous devez travailler directement sur le référentiel, sans avoir de copie de travail. C'est à cela que sert le *Explorateur de référentiel*. Comme l'explorateur et les recouvrements d'icône qui vous permettent de voir votre copie de travail, l'explorateur de référentiel vous permet de voir la structure et le statut du référentiel.



**Figure 4.44. l'explorateur de référentiel**

Avec l'explorateur de référentiel vous pouvez exécuter des commandes comme copier, déplacer, renommer... directement sur le référentiel.

L'explorateur de référentiel est très similaire à l'explorateur Windows, excepté le fait qu'il affiche le contenu du référentiel à une révision donnée au lieu de fichiers sur votre ordinateur. Dans le panneau de droite vous pouvez voir l'arborescence des répertoires, et à droite le contenu du répertoire sélectionné. En haut de l'explorateur vous pouvez entrer l'URL du référentiel et la révision que vous souhaitez afficher.

Tout comme dans l'explorateur Windows, vous pouvez cliquer sur les en-têtes des colonnes dans le panneau de droite si vous voulez modifier le tri. Et comme dans l'explorateur il y a des menus contextuels dans les deux panneaux.

Le menu contextuel pour un fichier vous permet de :

- Ouvrir le fichier sélectionné, avec le visualisateur par défaut pour ce type de fichier ou avec le programme de votre choix.
- Récupérer une copie non versionnée du fichier sur votre disque dur.
- Afficher les commentaires de révision de ce fichier, ou afficher le graphique de toutes les révisions de manière à voir d'où vient le fichier.
- Annoter le fichier pour voir qui a modifié quelle ligne et quand.
- Supprimez ou renommez le fichier.
- Faire une copie du fichier, à un autre endroit du référentiel ou dans une copie de travail hébergée par le même référentiel.
- Voir/Modifier les propriétés du fichier

Le menu contextuel pour un dossier vous permet de :

- Voir l'historique des messages de log de ce répertoire, ou voir le graphique de toutes les révisions afin de pouvoir savoir d'où vient ce répertoire.
- Exporter le répertoire vers une copie non versionnée sur votre disque dur.
- Extraire le répertoire pour avoir une version de travail sur votre disque dur.
- Créez un nouveau répertoire dans le référentiel
- Ajouter directement des fichiers ou des répertoires dans le référentiel.
- Supprimez ou renommez le répertoire.
- Make a copy of the folder, either to a different part of the repository, or to a working copy rooted in the same repository.
- Voir/Modifier les propriétés du dossier.
- Marque le répertoire pour la comparaison. Un nom d'un répertoire marqué est en gras.
- Comparez le dossier avec un dossier préalablement marqué, soit avec des différences unifiées, soit comme une liste de fichiers modifiés qui peuvent être comparés visuellement en utilisant l'outil de comparaison par défaut. Cela peut être particulièrement utile pour comparer deux tags, ou le tronc et les branches pour voir ce qui a changé.

Si vous choisissez deux éléments dans le panneau de droite, vous pouvez voir les différences soit avec des différences unifiées, soit comme une liste des fichiers qui peuvent être comparés visuellement en utilisant l'outil de comparaison par défaut.

Si vous sélectionnez plusieurs répertoires dans le panneau de droite, vous pouvez les extraire tous simultanément dans un dossier parent commun.

Si vous sélectionnez 2 étiquettes qui sont copiées depuis la même racine (typiquement /trunk/), vous pouvez utiliser **Menu contextuel** → **Voir le journal** pour voir la liste des révisions entre les deux points d'étiquettes.

Vous pouvez utiliser **F5** pour actualiser l'affichage. Cela mettra à jour tout ce qui est affiché. Si vous souhaitez chercher ou actualiser les informations pour les nœuds qui n'ont pas encore été ouverts, utilisez **Ctrl-F5**. Ouvrir n'importe quel nœud se fera alors instantanément, sans délai dû au réseau, lors de l'affichage de l'information.

Vous pouvez aussi utiliser l'explorateur de référentiel pour les opérations de glisser-déplacer. Si vous glissez un dossier de l'explorateur dans l'explorateur de référentiel, il sera importé dans le référentiel. Notez que si vous glissez plusieurs éléments, ils seront importés dans des livraisons séparées.

Si vous voulez déplacer un élément au sein du référentiel, effectuez un glisser-déplacer avec le bouton gauche de la souris vers le nouvel emplacement. Si vous voulez créer une copie plutôt que de déplacer l'élément, effectuez un glisser-déplacer avec le bouton gauche de la souris, tout en maintenant la touche **Ctrl** enfoncée. Lors de la copie, le curseur affiche un symbole « plus », comme dans l'Explorateur.

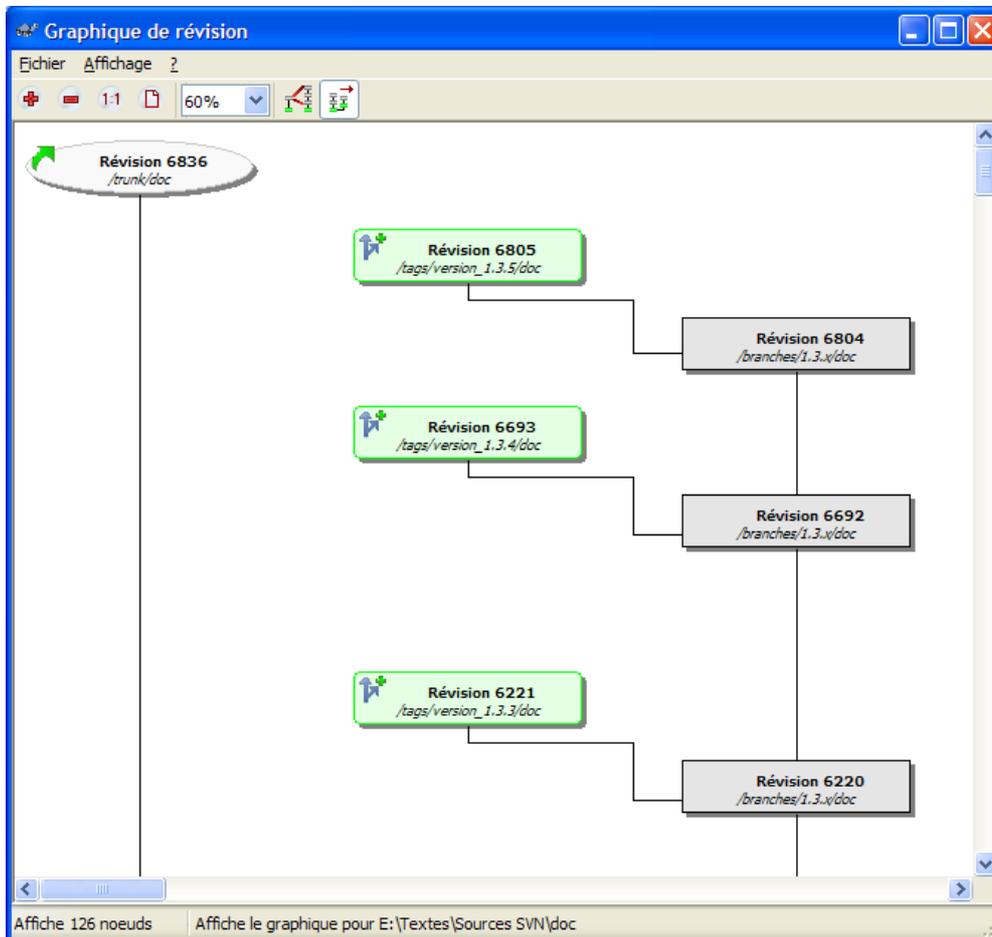
Si vous voulez copier/déplacer un fichier ou un dossier vers un autre emplacement et lui donner en même temps un nouveau nom, vous pouvez le glisser avec le bouton droit ou le glisser avec le bouton droit et **Ctrl** au lieu de le glisser avec le bouton gauche. Dans ce cas, une boîte de dialogue renommer s'affiche où vous pouvez saisir un nouveau nom pour le fichier ou le dossier.

Quand vous faites des changements dans le référentiel en utilisant l'une de ces méthodes, une boîte de dialogue de saisie de commentaire vous sera présentée. Si vous avez glissé quelque chose par erreur, c'est aussi une chance pour annuler l'action.

Parfois quand vous essayez d'ouvrir un chemin, vous obtiendrez un message d'erreur à la place des détails de l'élément. Cela pourrait arriver si vous spécifiez une URL invalide, ou si vous n'avez pas

d'autorisation d'accès, ou s'il y a d'autres problèmes serveur. Si vous devez copier ce message pour l'inclure dans un e-mail, faites simplement un clic droit dessus et utilisez Menu contextuel → Copier le message d'erreur dans le presse-papiers, ou utilisez simplement CTRL+C.

## 4.25. Graphiques de révision



**Figure 4.45. Un graphique de révision**

Parfois vous avez besoin de savoir à quel moment les branches et les étiquettes ont été prises du tronc et la façon idéale de voir ce genre d'informations est comme un graphique ou une structure arborescente. C'est à ce moment que vous devez utiliser TortoiseSVN → Graphique de révision

Cette commande analyse l'historique des révisions et essaye de créer un arbre montrant les points auxquels les copies ont été prises et quand les branches/étiquettes ont été supprimées.



### Important

Pour générer le graphique, TortoiseSVN doit aller chercher tous les commentaires dans la racine du référentiel. Inutile de dire que cela peut prendre plusieurs minutes même avec un référentiel de seulement quelques milliers de révisions, selon la vitesse du serveur, la bande passante du réseau, etc. Si vous essayez avec quelque chose comme le projet Apache, qui a actuellement plus de 500 000 révisions, vous pourriez attendre un moment.

La bonne nouvelle est que vous si mettez le journal en cache, vous n'aurez à attendre qu'une fois. Par la suite, ces données seront stockées localement. Ce paramètre est activé dans la configuration de TortoiseSVN.

### 4.25.1. Graphiques des révisions

Chaque noeud du graphe représente une révision dans le référentiel dans lequel quelque chose a changé. Les différents types de nœuds peuvent être distingués par leur forme et leur couleur. Les formes sont fixes, mais les couleurs peuvent être définies à l'aide de TortoiseSVN → Paramètres

#### Eléments ajoutés ou copiés

Les éléments qui ont été ajoutés, ou créés en copiant un autre fichier/dossier sont affichés en utilisant un rectangle arrondi. La couleur par défaut est verte. Les tags et les troncs sont traités à part et utilisent une teinte différente, en fonction de TortoiseSVN → Paramètres

#### Elements supprimés

Les éléments supprimés, par exemple une branche qui n'est plus nécessaire, sont montrées en utilisant un octogone (un rectangle avec des coins coupés). La couleur par défaut est rouge.

#### Elements renommés

Les éléments renommés seront également affichés avec un octogone, mais seront de couleur bleue par défaut.

#### Révision de pointe de branche

Le graphique est normalement limité aux points de branchement, mais il peut aussi être souvent utile de voir la révision "HEAD" respective de chaque branche. Si vous sélectionnez **Afficher les révisions HEAD**, chaque nœud de révision HEAD sera affiché comme une ellipse. Notez que HEAD se réfère ici à la dernière révision livrée sur ce chemin, et non à la révision HEAD du référentiel.

#### Révision de la Copie de Travail

Si vous avez invoqué le graphique de révision à partir d'une copie de travail, vous pouvez choisir de montrer la révision BASE sur le graphique à l'aide de **Afficher la révision WC**, qui marque le noeud BASE avec un contour en gras.

#### Copie de travail modifiée

Si vous avez invoqué le graphique de révision à partir d'une copie de travail, vous pouvez choisir de montrer un nœud supplémentaire représentant votre copie de travail modifiée à l'aide de **Afficher les modifications WC**. Il s'agit d'un noeud elliptique avec un contour en gras rouge par défaut.

#### Element normal

Tous les autres éléments sont affichés en utilisant un rectangle plat.

Notez que par défaut le graphique ne montre que les points auxquels des éléments ont été ajoutés, copiés ou supprimés. Afficher chaque révision d'un projet pourrait produire un graphique très grand pour des cas peu communs. Si vous voulez vraiment voir *toutes* les révisions pour lesquelles des modifications ont été apportées, il existe une option pour ce faire dans le menu **Affichage** et sur la barre d'outils.

L'affichage par défaut (sans regroupement) place les nœuds de telle sorte que leur position verticale reflète l'ordre strict de révision, ainsi vous avez un repère visuel pour l'ordre dans lequel les choses ont été faites. Lorsque deux nœuds sont dans la même colonne, l'ordre est très évident. Lorsque deux nœuds sont dans des colonnes voisines, le décalage est beaucoup plus faible car il n'est pas nécessaire d'empêcher les nœuds de se chevaucher, et en conséquence l'ordre est un peu moins évident. Ces optimisations sont nécessaires pour que les graphiques complexes restent à une taille raisonnable. Veuillez noter que cette organisation utilise le *bord* du noeud sur l'*ancien* côté comme une référence, c'est à dire le bord inférieur du nœud lorsque le graphique est représenté avec le plus ancien noeud en bas. Le bord de référence est important, car les formes des nœuds ne font pas toutes la même hauteur.

### 4.25.2. Changer l'affichage

Parce qu'un graphique de révision est souvent très complexe, il existe un certain nombre de caractéristiques qui peuvent être utilisées pour adapter la façon dont vous souhaitez le voir. Celles-ci sont disponibles dans le menu **Voir** et à partir de la barre d'outils.

### Groupe les branches

Le comportement par défaut (sans regroupement) montre toutes les lignes triées strictement par révision. En conséquence, les branches très anciennes avec des modifications clairsemées occupent une colonne entière pour seulement quelques changements et le graphe devient très large.

Ce mode regroupe les changements par branche, il n'y a alors pas de classement des révisions: des révisions consécutives sur une branche seront (souvent) affichées sur des lignes consécutives. Les sous-branches, cependant, sont disposés de telle manière que les futures branches seront affichées dans la même colonne au-dessus des plus anciennes branches pour conserver un graphique le plus léger possible. En conséquence, une ligne donnée peut contenir des modifications de différentes révisions.

### Les plus anciens en haut

Normalement, le graphique affiche les révisions les plus anciennes en bas, et l'arborescence se développe par le haut. Utilisez cette option pour que cette dernière s'affiche de haut en bas.

### Aligner les arborescences en haut

Lorsqu'un graphique est divisé en plusieurs petits arbres, les arbres peuvent apparaître soit dans l'ordre naturel de révision, ou alignés au bas de la fenêtre, selon que vous utilisiez ou pas l'option **Grouper les branches**. Utilisez cette option pour développer tous les arbres du haut vers le bas.

### Réduire les intersections

Si la mise en page du graphique a produit un grand nombre de lignes se croisant, utilisez cette option pour la nettoyer. Cela peut rendre la présentation des colonnes moins logique, par exemple dans une diagonale plutôt que dans une colonne, et le graphique peut demander une plus grande surface pour être dessiné.

### Chemins d'accès différentiels

Des noms de chemin longs peuvent prendre beaucoup d'espace et rendre les cases des noeuds très large. Utilisez cette option pour afficher seulement la partie modifiée d'un chemin, en remplaçant la partie commune avec des points. Par exemple, si vous créez une branche `/branches/1.2.x/doc/html` à partir de `/trunk/doc/html`, la branche pourrait être présentée sous forme compacte `/branches/1.2.x/. .` parce que les deux derniers niveaux, `doc` et `html` n'ont pas changés.

### Montrer toutes les révisions

Cela fait exactement ce que vous attendiez et montre chaque révision où quelque chose (dans l'arbre que vous dessinez) a changé. Pour un long historique, cela peut produire un graphique véritablement énorme.

### Montrer la version de tête (HEAD)

Cela garantit que la dernière révision de chaque branche est toujours indiquée sur le graphique.

### Copie exacte des sources

Quand une branche / étiquette est créée, le comportement par défaut est de montrer la branche comme tirée du dernier nœud où une modification a été apportée. Strictement parlant cela est inapproprié puisque les branches sont souvent issues du HEAD courant plutôt que d'une révision spécifique. Il est donc possible de montrer la révision la plus juste (mais la moins utile) qui a été utilisé pour créer la copie. Notez que cette révision peut être plus récente que la révision HEAD de la branche source.

### Replier les étiquettes

Lorsqu'un projet a de nombreux tags, les montrer tous dans des nœuds distincts sur le graphique prend beaucoup d'espace et rend moins lisible la structure développée des branches la plus intéressante. Dans le même temps vous pourriez avoir besoin d'accéder au contenu du tag facilement afin de pouvoir comparer les révisions. Cette option cache les nœuds pour les tags et les montre plutôt dans l'info-bulle pour le nœud à partir duquel ils ont été copiés. Une icône de tag sur le côté droit du nœud de la source indique que des tags ont été créés.

### Cacher les répertoires supprimés

Cache les chemins qui ne sont plus présents dans la révision HEAD du référentiel, par exemple, des branches supprimées.

#### Cacher les branches non modifiées

Cacher les branches où aucun changement n'a été validé dans le fichier ou le sous-dossier respectif. Cela ne signifie pas nécessairement que la branche n'a pas été utilisée, simplement qu'aucune modification n'a été apportée à *cette* partie de la branche.

#### Montrer la version de la CdT

Marquer la révision sur le graphique qui correspond à la révision mise à jour de l'élément pour lequel vous affichez le graphique. Si vous avez juste mis à jour, ce sera HEAD, mais si d'autres personnes ont livrés des changements depuis votre dernière mise à jour, votre CdT (copie de travail) peut se retrouver à quelques révisions antérieures. Le nœud est marqué en lui donnant un contour gras.

#### Montrer les modifications de la CdT

Si votre CdT contient des modifications locales, cette option la représente comme un nœud séparé elliptique, lié au nœud pour lequel votre CdT a été mise à jour en dernier. La couleur du contour par défaut est le rouge. Vous pouvez avoir besoin d'actualiser le graphique à l'aide de **F5** pour afficher les récents changements.

#### filtrer

Parfois, le graphique de révision contient plus de révisions que vous ne voulez en voir. Cette option ouvre une boîte de dialogue qui vous permet de restreindre la plage des révisions affichées, et de cacher des chemins particuliers par leur nom.

#### Couleur des arbres

Lorsque le graphique contient plusieurs arbres, il est parfois intéressant d'utiliser des couleurs en alternance pour pouvoir les distinguer.

#### Montrer l'aperçu

Affiche une petite image du graphique en entier, avec la fenêtre courante dans un rectangle que vous pouvez faire glisser. Cela vous permet de naviguer sur le graphique plus facilement. Notez que pour les très grands graphiques, l'aperçu peut devenir inutile en raison du facteur de zoom extrême et ne sera donc pas montré dans de tels cas.

### 4.25.3. Utiliser le Graphique de révisions

Pour rendre plus facile la navigation dans un large graphe, utilisez la fenêtre Aperçu. Cela montre le graphique entier dans une petite fenêtre, avec la partie actuellement affichée en surbrillance. Vous pouvez faire glisser la zone en surbrillance pour changer la région affichée.

La date de révision, l'auteur et les commentaires sont affichés dans une info-bulle chaque fois que la souris survole une boîte de révision.

Si vous sélectionnez deux révisions (faites un clic gauche avec Ctrl), vous pouvez utiliser le menu contextuel pour afficher les différences entre ces révisions. Vous pouvez vouloir afficher les différences comme aux points de création de branche, mais le plus souvent vous voudrez afficher les différences aux points de fin de branche, c'est-à-dire à la révision HEAD.

Vous pouvez voir les différences avec un fichier de différences unifiées, qui montre toutes les différences dans un seul fichier avec un contexte minimal. Si vous optez pour Menu contextuel → Comparer les révisions, vous vous retrouvez avec une liste des fichiers modifiés. Double-cliquez sur un nom de fichier pour aller chercher les deux révisions du fichier et les comparer en utilisant l'outil de comparaison visuel.

Si vous faites un clic droit sur une révision, vous pouvez utiliser Menu contextuel → Voir le journal pour voir l'historique.

Vous pouvez également fusionner les modifications dans la (les) révision(s) sélectionnée(s) dans une autre copie de travail. Une boîte de dialogue de sélection de dossier vous permet de choisir la copie de travail vers laquelle fusionner, mais après cela il n'y a pas de demande de confirmation, ni possibilité de mener un test de fusion. C'est une bonne idée de fusionner dans une copie de travail non modifiée afin

que vous puissiez annuler les modifications si elles ne fonctionnent pas! Cette fonctionnalité est utile si vous souhaitez fusionner les versions sélectionnées d'une branche à l'autre.



## Apprendre à Lire le Graphe de Révisions

Les utilisateurs qui affichent un graphique pour la première fois peuvent être surpris par le fait que le graphique de révision montre quelque chose qui ne correspond pas au modèle auquel pense l'utilisateur. Si une révision change des copies multiples ou des branches d'un fichier ou d'un dossier, par exemple, alors il y aura plusieurs nœuds pour cette seule révision. Il est bon de commencer avec les options par défaut dans la barre d'outils et de personnaliser graphique étape par étape jusqu'à ce qu'il se rapproche de votre vision.

Toutes les options de filtre essaient de perdre le moins d'informations possible. Cela peut provoquer le changement de couleur de certains nœuds, par exemple. Chaque fois que le résultat est inattendu, annuler la dernière opération de filtrage et essayer de comprendre ce qui est particulier à cette révision ou cette branche. Dans la plupart des cas, le résultat initialement prévu de l'opération de filtrage est soit inexact soit trompeur.

### 4.25.4. Refraîchissement de l'affichage

Si vous voulez à nouveau vérifier le serveur pour de nouvelles informations, il vous suffit de rafraîchir la vue en utilisant **F5**. Si vous utilisez le cache log (activé par défaut), cela va examiner le référentiel pour les livraisons les plus récentes et récupérer seulement les nouvelles. Si le cache log était en mode hors connexion, cela permettra également de revenir en arrière en ligne.

Si vous utilisez le cache log et vous pensez que le contenu du message ou l'auteur a changé, vous devez utiliser la boîte de dialogue de log pour actualiser les messages dont vous avez besoin. Puisque le graphique des révisions se base sur la racine du référentiel, nous aurions à invalider intégralement le cache log, et le remplir à nouveau pourrait prendre un temps *très* long.

### 4.25.5. Visualisation des arborescences

Il peut être difficile de naviguer dans un grand arbre et parfois vous souhaitez en cacher certaines parties, ou le décomposer en une forêt de plus petits arbres. Si vous passez la souris sur le point où il y a un lien de noeud, vous verrez un ou plusieurs boutons popup qui vous permettent de le faire.



Cliquez sur le bouton "moins" pour réduire la sous arborescence.



Cliquez sur le bouton "plus" pour développer une sous arborescence. Quand une arborescence a été réduite, ce bouton reste pour montrer la sous arborescence cachée.



Cliquez sur le bouton "croix" pour séparer la sous arborescence et l'afficher comme une arborescence séparée.

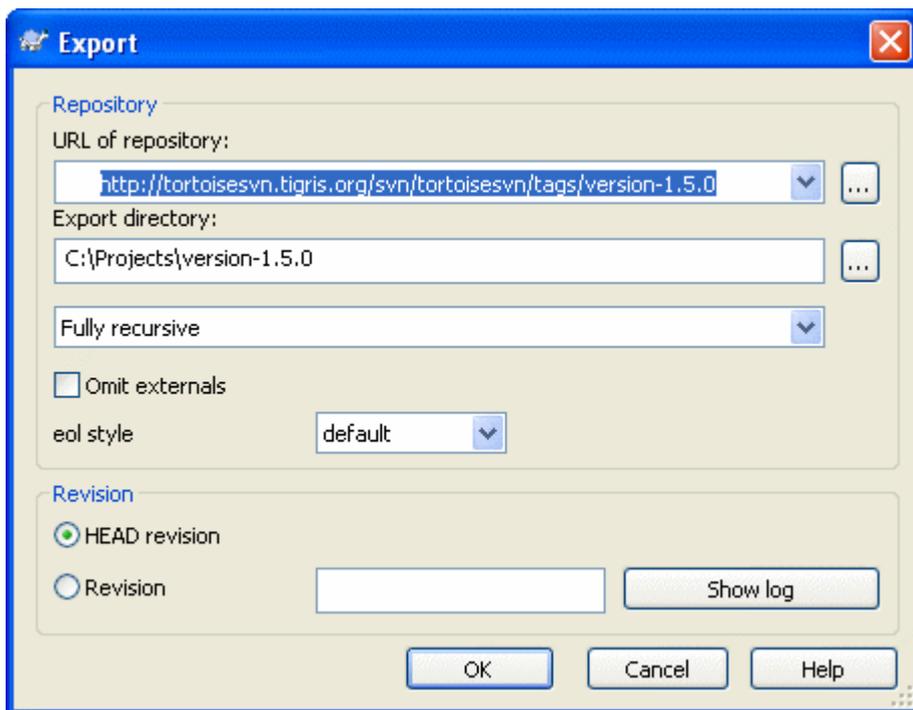


Cliquez sur le bouton "rond" pour ré-attacher une sous arborescence préalablement séparée. Quand une arborescence a été séparée, ce bouton reste visible pour indiquer qu'il y a une sous arborescence séparée.

Cliquez sur l'arrière plan du graphique pour le menu contextuel principal, qui offre des options pour Tout développer et Tout replier. Si aucune branche n'a été repliée ou scindée, le menu contextuel ne sera pas montré.

## 4.26. Exporter une copie de travail Subversion

Parfois vous pouvez vouloir une copie de votre arborescence de travail sans ces répertoires `.svn`, par exemple pour créer un paquet zippé de votre source ou pour l'exporter vers un serveur web. Au lieu de faire une copie puis de supprimer tous ces répertoires `.svn` manuellement, TortoiseSVN propose la commande TortoiseSVN → Exporter.... Exporter à partir d'une URL et exporter à partir d'une copie de travail sont traités un peu différemment.



**Figure 4.46. La fenêtre extraction-depuis-une-URL**

Si vous exécutez cette commande sur un dossier non versionné, TortoiseSVN va supposer que le dossier sélectionné est la cible, et va ouvrir une boîte de dialogue pour que vous entriez l'URL et la révision vers lesquelles exporter. Cette boîte de dialogue a des options pour exporter seulement le dossier de niveau supérieur, pour omettre des références externes et pour remplacer le style de fin de ligne pour les fichiers pour lesquels `svn:eol-style` est activé.

Bien sûr, vous pouvez aussi exporter directement depuis le référentiel. Utilisez l'explorateur de référentiel pour naviguer au sous-arbre pertinent dans votre référentiel, utilisez ensuite **Menu contextuel** → **Exporter**. Vous obtiendrez la boîte de dialogue **Exporter depuis l'URL** décrite ci-dessus.

Si vous exécutez cette commande sur votre copie de travail on vous demandera un emplacement pour sauvegarder la copie de travail propre sans les répertoires `.svn`. Par défaut, seuls les fichiers versionnés sont exportés, mais vous pouvez utiliser la case **Exporter aussi les fichiers non versionnés** pour inclure tous les autres fichiers non versionnés qui existent dans votre CdT et pas dans le référentiel. Les références externes utilisant `svn:externals` peuvent être omises au besoin.

Une autre façon de faire une exportation consiste à glisser avec le bouton droit un dossier de la copie de travail vers un autre emplacement et de choisir **Menu contextuel** → **SVN Exporter ici** ou **Menu contextuel** → **SVN Exporter tout ici**. La seconde option inclut aussi les fichiers non versionnés.

Lors de l'export à partir d'une CdT, si le répertoire cible contient déjà un répertoire du même nom que celui que vous exportez, vous aurez la possibilité d'écraser le contenu existant ou de créer un nouveau répertoire avec un nom généré automatiquement, par exemple `Cible (1)`.



### Extraire quelques fichiers

La boîte de dialogue d'exportation ne permet pas d'extraire juste quelques fichiers, même si Subversion le permet.

Pour exporter des fichiers individuels avec TortoiseSVN, vous devez utiliser le navigateur du référentiel ([Section 4.24, « l'explorateur de référentiel »](#)). Il suffit de glisser le(s) fichier(s) que vous souhaitez exporter du navigateur du référentiel à l'endroit où vous le souhaitez dans l'explorateur, ou utiliser le menu contextuel dans le navigateur du référentiel pour exporter les fichiers.



### Exporter une Arborescence des Modifications

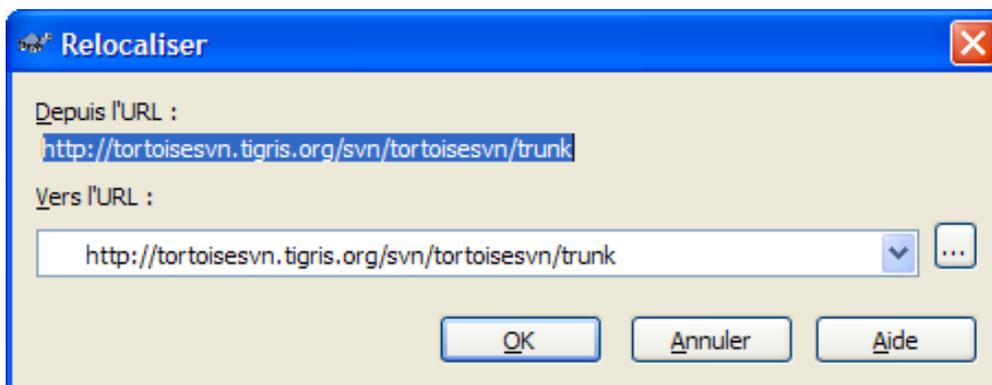
Si vous voulez exporter une copie de votre arborescence de projet mais ne contenant que les fichiers qui ont changé dans une révision particulière, ou entre deux révisions, utilisez la fonctionnalité décrite dans [Section 4.10.3, « Comparer des répertoires »](#).

#### 4.26.1. Retirer une copie de travail du contrôle de version

Parfois, vous avez une copie de travail que vous souhaitez convertir vers un dossier normal sans le répertoire `.svn`. Ce dont vous avez alors besoin est une commande d'"export en place", qui supprime simplement les répertoires de contrôle au lieu de générer une nouvelle et complète arborescence des répertoires.

La réponse est étonnamment simple - exporter le dossier vers lui-même! TortoiseSVN détecte ce cas particulier et vous demande si vous voulez rendre la copie de travail non versionnée. Si vous répondez *oui* les répertoires de contrôle seront supprimés et vous aurez une arborescence de répertoires complète et non versionnée.

#### 4.27. Relocaliser une copie de travail



**Figure 4.47. La boîte de dialogue Relocaliser**

Si votre référentiel a, pour quelque raison que ce soit, changé d'emplacement (IP/URL). Peut-être que vous êtes même bloqué(e) et que vous ne pouvez pas livrer, que vous ne voulez pas extraire votre copie

de travail à nouveau depuis le nouvel emplacement et déplacer toutes vos données modifiées dans la nouvelle copie de travail, alors TortoiseSVN → Relocaliser est la commande que vous recherchez. Son action principale est très succincte : elle parcourt toutes les fichiers `entries` dans les dossiers `.svn` et change l'URL des entrées avec la nouvelle valeur.

Vous pourriez être surpris de constater que TortoiseSVN contacte le référentiel dans le cadre de cette opération. Tout ce qu'il fait est l'exécution de contrôles simples pour faire en sorte que la nouvelle URL se réfère vraiment au même référentiel que la copie de travail existante.



### Avertissement

*C'est une opération très rare.* La commande `relocaliser` est utilisée *seulement* si l'URL de la racine du référentiel a changé. Les raisons possibles sont :

- L'adresse IP du serveur a changé.
- Le protocole a changé (par exemple `http://` en `https://`).
- Le chemin de la racine du référentiel a changé dans le paramétrage du serveur.

Autrement, vous devez relocaliser quand votre copie de travail se réfère au même emplacement dans le même référentiel, mais le référentiel lui-même a été déplacé.

Elle ne s'applique pas si :

- Vous voulez vous déplacer vers un référentiel Subversion différent. Dans ce cas, vous devriez exécuter une extraction fraîche à partir du nouvel emplacement du référentiel.
- Vous voulez aller sur une branche différente ou sur un répertoire dans le même référentiel. Pour ce faire, vous devriez utiliser TortoiseSVN → Aller sur.... Lisez [Section 4.19.2](#), « **Extraire ou aller sur...** » pour plus d'informations.

Si vous utilisez `relocaliser` dans n'importe lequel des cas ci-dessus, cela *corrompra votre copie de travail* et vous obtiendrez beaucoup de messages d'erreur inexplicables en mettant à jour, en livrant, etc. Une fois que cela s'est produit, la seule correction est une extraction fraîche.

## 4.28. Intégration avec des systèmes de gestion de bug / gestion d'incidents

Il est très fréquent dans le développement logiciel que les changements soient liés à un bug spécifique ou à un ID d'incident. Les utilisateurs de systèmes de traque de bug (traqueurs d'incidents) peuvent associer les changements qu'ils font dans Subversion avec un ID spécifique dans leur traqueur d'incidents. La plupart des traqueurs fournissent donc un script hook de pre-commit qui analyse syntaxiquement le commentaire pour trouver l'ID du bug auquel la livraison est associée. C'est une source d'erreurs puisque l'utilisateur a la responsabilité d'écrire correctement le commentaire afin que le script hook de pre-commit puisse en faire l'analyse syntaxique correctement.

TortoiseSVN peut aider l'utilisateur de deux manières :

1. Quand l'utilisateur entre un commentaire, une ligne bien définie incluant le numéro de l'incident associé à la livraison peut être ajoutée automatiquement. Cela réduit le risque que l'utilisateur entre le numéro de l'incident d'une façon que les outils de traque de bugs ne peuvent pas analyser correctement.

Ou TortoiseSVN peut mettre en évidence la partie du commentaire saisi qui est reconnu par le gestionnaire d'incidents. De cette façon, l'utilisateur sait que le commentaire peut être correctement analysé syntaxiquement.

2. Quand l'utilisateur parcourt les commentaires, TortoiseSVN crée un lien à partir de chaque ID de bug dans le commentaire qui lance le navigateur à l'incident mentionné.

#### 4.28.1. Ajouter des numéros d'incidents aux messages de log

Vous pouvez intégrer le traqueur de bugs de votre choix dans TortoiseSVN. Pour ce faire, vous devez définir quelques propriétés, qui commencent par `bugtraq:`. Elles doivent être définies sur les dossiers : (Section 4.17, « Configuration des projets »)

Il y a deux façons d'intégrer TortoiseSVN avec les gestionnaires d'incidents. L'une est basée sur des chaînes simples, l'autre sur les expressions régulières. Les propriétés utilisées par les deux approches sont :

`bugtraq:url`

Cette propriété correspond à l'URL de votre traqueur de bugs. Cette URL doit être correctement encodée et doit contenir `%BUGID%`. `%BUGID%` est remplacé par le numéro de l'incident que vous avez saisi. Cela permet à TortoiseSVN d'afficher un lien dans la boîte de dialogue de journal, qui pointe vers l'entrée correspondant au développement dans votre outil de traque de bugs. Vous n'êtes pas obligé de fournir cette propriété, TortoiseSVN ne montre alors que le numéro de l'incident, pas le lien. Par exemple le projet de TortoiseSVN utilise `http://issues.tortoisesvn.net/?do=details&id=%BUGID%`

Vous pouvez utiliser des URLs relatives à la place des URLs absolues. Cela peut être utile quand votre gestionnaire d'incidents est sur le même serveur que votre référentiel. Ainsi, si vous changez de nom de domaine, vous n'aurez pas à modifier la propriété `bugtraq:url`. Il y a deux manières de donner des URL relatives :

Si ça commence avec le caractère `^/`, cela suppose que c'est relatif à la racine du référentiel. Par exemple, `^/../?do=details&id=%BUGID%` deviendra `http://tortoisesvn.net/?do=details&id=%BUGID%` si votre référentiel est situé sur `http://tortoisesvn.net/svn/trunk/`.

Une URL commençant par le caractère `/` est supposé être relatif au nom de l'hôte du serveur. Par exemple `/?do=details&id=%BUGID%` se résume à `http://tortoisesvn.net/?do=details&id=%BUGID%` si votre référentiel est situé n'importe où sur `http://tortoisesvn.net`.

`bugtraq:warnifnoissue`

Mettez cela à `true`, si vous voulez que TortoiseSVN vous avertisse quand le champ du numéro d'incident est vide. Les valeurs valables sont `true/false`. *Si elle n'est pas définie, false est prise par défaut.*

##### 4.28.1.1. Numéro d'incident dans un Champ Texte

Dans l'approche simple, TortoiseSVN montre à l'utilisateur un champ de saisie séparé où un ID de bug peut être entré. Une ligne séparée est alors ajoutée avant/après le commentaire que l'utilisateur a saisi.

`bugtraq:message`

Cette propriété active le système de gestion de bugs en mode *champ de saisie*. Si cette propriété est définie, alors TortoiseSVN vous demandera d'entrer un numéro d'incident quand vous livrez vos changements. Elle est utilisée pour ajouter une ligne à la fin du commentaire. Elle doit contenir `%BUGID%`, qui est remplacé par le numéro d'incident à la livraison. Cela assure que votre journal de livraison contient une référence au numéro d'incident qui est toujours dans un format cohérent et peut être analysé syntaxiquement par votre outil de gestion de bugs pour associer le numéro d'incident à une livraison particulière. Par exemple vous pourriez utiliser `Incident : %BUGID%`, mais cela dépend de votre outil.

`bugtraq:append`

Cette propriété définit si l'ID-bug est ajouté (`true`) à la fin du commentaire ou inséré (`false`) au début du commentaire. Les valeurs valables sont `true/false`. *Si elle n'est pas définie, true est par défaut, pour que les projets existants ne cassent pas.*

`bugtraq:label`

Ce texte est affiché par TortoiseSVN sur la boîte de dialogue de livraison comme label pour la boîte de saisie où vous entrez le numéro d'incident. S'il n'est pas défini, `Bug-ID/ N° d'incident` sera affiché. Gardez à l'esprit que la fenêtre ne sera pas redimensionnée pour s'adapter à ce label, gardez donc la taille du label en-dessous de 20-25 caractères.

`bugtraq:number`

Si elle est définie à `true`, seuls les nombres sont autorisés dans le champ du numéro d'incident. La virgule est une exception, donc vous pouvez séparer plusieurs numéros par des virgules. Les valeurs valides sont `true/false`. *Si elle n'est pas définie, true est la valeur par défaut.*

#### 4.28.1.2. Numéros d'incidents utilisant des expressions régulières

Dans l'approche avec les expressions régulières, TortoiseSVN n'affiche pas de champ de saisie séparé, mais marque la partie du commentaire que l'utilisateur entre qui est reconnue par le gestionnaire d'incidents. Cela se fait pendant que l'utilisateur écrit le commentaire. Cela signifie aussi que l'ID de bug peut être n'importe où à l'intérieur d'un commentaire ! Cette méthode est beaucoup plus souple et c'est celle utilisée par le projet TortoiseSVN lui-même.

`bugtraq:logregex`

Cette propriété active le système de bug tracking en mode *Regex*. Elle contient soit une expression régulière, soit deux séparées par un retour à la ligne.

Si deux expressions sont définies, la première expression est utilisée comme un pré-filtre pour trouver des expressions qui contiennent les ID de bug. Ensuite, la deuxième expression extrait les ID du résultat de la première regex. Cela vous permet d'utiliser une liste d'ID de bugs et des expressions du langage naturel si vous le souhaitez. Par exemple, vous pourriez corriger plusieurs bugs et inclure un texte comme ceci : « Ce changement résout les incidents #23, #24 et #25 ».

Si vous voulez isoler les ID des bugs tels qu'utilisés à l'intérieur des messages de log, vous pouvez utiliser les expressions régulières suivantes, qui sont celles utilisées par TortoiseSVN en personne : `[Ii]ssues?:?(\\s*( , |and)?\\s*#\\d+)+ et (\\d+)`

La première expression extrait « incidents #23, #24 et #25 » du message de log. La seconde regex extrait les nombres décimaux du résultat de la première regex, on aura donc « 23 », « 24 » et « 25 » à utiliser comme ID des bugs.

Détaillons un peu la première regex. Elle doit commencer par le mot « incident », qui peut être en majuscule. Ceci est éventuellement suivie d'un « s » (plus d'un problème) et éventuellement une virgule. C'est suivi par un ou plusieurs groupes ayant chacun zéro ou plusieurs espaces en tête, une virgule optionnelle ou « et » et d'autres espaces optionnels. Enfin, il y a un « # » obligatoire et un nombre décimal obligatoire.

Si seule une expression est définie, alors l'ID du bug seul doit correspondre aux groupes de la chaîne regex. Exemple : `[Ii]ssue(?:s)? #?(\\d+)`. Cette méthode est requise par quelques gestionnaires d'incidents, par exemple trac, mais il est plus difficile de construire la regex. Nous vous recommandons d'utiliser cette méthode uniquement si la documentation de votre gestionnaire d'incidents vous le demande.

Si vous êtes peu familier avec les expressions régulières, jetez un coup d'oeil à l'introduction [http://en.wikipedia.org/wiki/Regular\\_expression](http://en.wikipedia.org/wiki/Regular_expression) et à la documentation en ligne et au didacticiel à <http://www.regular-expressions.info/>.

Si les deux propriétés `bugtraq:message` et `bugtraq:logregex` sont définies, `logregex` a la priorité.



## Astuce

Même si vous n'avez pas de gestionnaire d'incidents avec un hook de pré-livraison analysant syntaxiquement vos commentaires, vous pouvez toujours utiliser cela pour transformer les incidents mentionnés dans vos commentaires en liens !

Et même si vous n'avez pas besoin de liens, les numéros des incidents apparaissent dans une colonne distincte dans la boîte de dialogue de log, facilitant la recherche des changements qui se rapportent à un incident particulier.

Certaines propriétés `tsvn:` exigent une valeur `true/false`. TortoiseSVN comprend aussi `yes` comme synonyme de `true` et `no` comme synonyme de `false`.



## Mettre les propriétés sur les dossiers

Pour que le système fonctionne elles doivent être définies sur des dossiers. Quand vous livrez un fichier ou un dossier les propriétés sont lues depuis ce dossier. Si les propriétés n'y sont pas trouvées, TortoiseSVN cherchera vers le haut dans l'arborescence des dossiers pour les trouver jusqu'à ce qu'il arrive à un dossier non versionné, ou à la racine de l'arborescence (par exemple `C:\`). Si vous pouvez être sûr que chaque utilisateur n'extrait que depuis `trunk/` par exemple et pas d'un autre sous-dossier, alors il est suffisant de définir les propriétés sur `trunk/`. Dans le cas contraire, vous devriez définir les propriétés récursivement sur chaque sous-dossier. Une propriété sur un répertoire remplace les réglages des niveaux supérieurs (plus près de `trunk/`).

For `tsvn: properties only` you can use the **Recursive** checkbox to set the property to all sub-folders in the hierarchy, without also setting it on all files.



## Aucune information sur le suivi des incidents à partir du navigateur du référentiel

Because the issue tracker integration depends upon accessing subversion properties, you will only see the results when using a checked out working copy. Fetching properties remotely is a slow operation, so you will not see this feature in action from the repo browser.

Cette intégration de gestionnaire d'incidents n'est pas limitée à TortoiseSVN ; elle peut être utilisée avec n'importe quel client Subversion. Pour plus d'informations, lisez [Spécification d'Intégration du Gestionnaire d'Incidents](http://tortoisesvn.googlecode.com/svn/trunk/doc/issuetrackers.txt) [http://tortoisesvn.googlecode.com/svn/trunk/doc/issuetrackers.txt] dans le référentiel source de TortoiseSVN. (Section 3, « [TortoiseSVN est gratuit !](#) » explique comment accéder au référentiel).

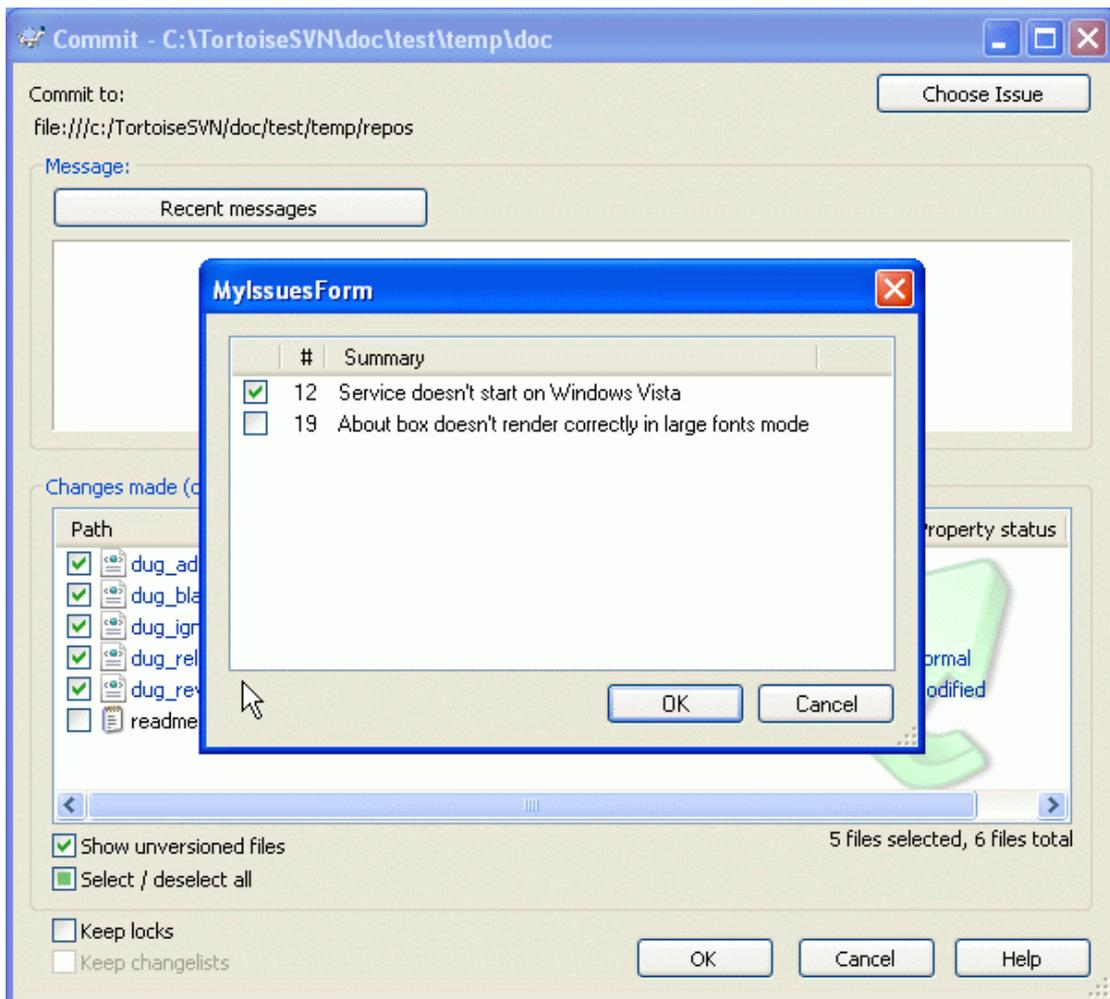
### 4.28.2. Récupérer des informations depuis le gestionnaire d'incidents

La section précédente porte sur l'ajout d'information concernant les incidents dans des messages de log. Mais que faire si vous avez besoin d'obtenir des informations du gestionnaire d'incidents ? La boîte de dialogue de révision possède une interface COM qui permet l'intégration d'un programme externe qui peut dialoguer avec votre logiciel de gestion. Typiquement, vous voudrez peut-être interroger le gestionnaire pour obtenir la liste des incidents ouverts qui vous sont assignés, de sorte que vous pouvez choisir les incidents qui sont abordés dans cette livraison.

Une telle interface est bien entendu très spécifique à votre système de gestion d'incidents, nous ne pouvons donc pas fournir cette partie, et décrire la façon de créer un tel programme est au-delà du champ d'application de ce manuel. La définition de l'interface et les plug-ins échantillon en C# et C++/ATL peuvent être obtenus dans le dossier `contrib` à l'adresse [Dépôt TortoiseSVN](http://tortoisesvn.googlecode.com/svn/trunk/contrib/issue-tracker-plugins) [http://tortoisesvn.googlecode.com/svn/trunk/contrib/issue-tracker-plugins]. (Section 3, « [TortoiseSVN](#)

est gratuit ! » explique comment accéder au dépôt). Un résumé de l'API est également donné à [Chapitre 6, \*IBugtraqProvider interface\*](#). Un autre exemple de plugin (fonctionnel) en C# est *Gurtle* [<http://code.google.com/p/gurtle/>], qui implémente l'interface COM nécessaire pour interagir avec le gestionnaire d'incidents *Google Code* [<http://code.google.com/hosting/>].

À titre d'exemple, supposons que votre administrateur système vous a fourni un plugin de gestionnaire d'incidents que vous avez installé, et que vous avez paramétré certaines de vos copies de travail pour utiliser le plugin dans la boîte de configuration de TortoiseSVN. Lorsque vous ouvrez la boîte de dialogue de livraison à partir d'une copie de travail à laquelle le plugin a été attribué, vous pourrez voir un nouveau bouton en haut de la boîte de dialogue.



**Figure 4.48. Exemple de fenêtre de gestionnaire d'incidents**

Dans cet exemple, vous pouvez sélectionner un ou plusieurs incidents en suspens. Le plugin peut alors générer du texte spécialement mis en forme qu'il ajoute à votre message de log.

## 4.29. Intégration avec des explorateur de référentiel de type web.

Il existe beaucoup de visionneuse de référentiel web compatibles avec Subversion comme *ViewVC* [<http://www.viewvc.org/>] et *WebSVN* [<http://websvn.tigris.org/>]. TortoiseSVN facilite l'intégration de celles ci.

Vous pouvez intégrer la visionneuse de référentiel de votre choix dans TortoiseSVN. Pour ce faire, vous n'avez qu'à renseigner quelques propriétés permettant la liaison. Elles sont à mettre sur les dossier : ([Section 4.17, « Configuration des projets »](#))

#### webviewer:revision

Cette propriété doit contenir l'URL de votre visionneuse de référentiel permettant de voir les modifications spécifiques à une révision. Ce doit être une URI valide et contenir %REVISION%. %REVISION% sera remplacé par le numéro de révision en question. Cela permet à TortoiseSVN d'afficher une commande dans le menu contextuel de la fenêtre de commentaire **Menu Contextuel** → Voir cette révision dans la visionneuse web

#### webviewer:pathrevision

Set this property to the URL of your repo viewer to view changes to a specific file in a specific revision. It must be properly URI encoded and it has to contain %REVISION% and %PATH%. %PATH% is replaced with the path relative to the repository root. This allows TortoiseSVN to display a context menu entry in the log dialog **Context Menu** → View revision and path in webviewer For example, if you right-click in the log dialog bottom pane on a file entry /trunk/src/file then the %PATH% in the URL will be replaced with /trunk/src/file.

Vous pouvez utiliser des URLs relatives à la place des URLs absolues. Cela peut être utile quand votre visionneuse est sur le même serveur que votre référentiel. Ainsi, si vous changez de nom de domaine, vous n'aurez pas à modifier les propriétés `webviewer:revision` et `webviewer:pathrevision`. Le format est le même que pour la propriété `bugtraq:url`. Voir [Section 4.28, « Intégration avec des systèmes de gestion de bug / gestion d'incidents »](#).



### Mettre les propriétés sur les dossiers

Pour que le système fonctionne elles doivent être définies sur des dossiers. Quand vous livrez un fichier ou un dossier les propriétés sont lues depuis ce dossier. Si les propriétés n'y sont pas trouvées, TortoiseSVN cherchera vers le haut dans l'arborescence des dossiers pour les trouver jusqu'à ce qu'il arrive à un dossier non versionné, ou à la racine de l'arborescence (par exemple C:\). Si vous pouvez être sûr que chaque utilisateur n'extrait que depuis `trunk/` par exemple et pas d'un autre sous-dossier, alors il est suffisant de définir les propriétés sur `trunk/`. Dans le cas contraire, vous devriez définir les propriétés récursivement sur chaque sous-dossier. Une propriété sur un répertoire remplace les réglages des niveaux supérieurs (plus près de `trunk/`).

For `tsvn`: properties *only* you can use the **Recursive** checkbox to set the property to all sub-folders in the hierarchy, without also setting it on all files.



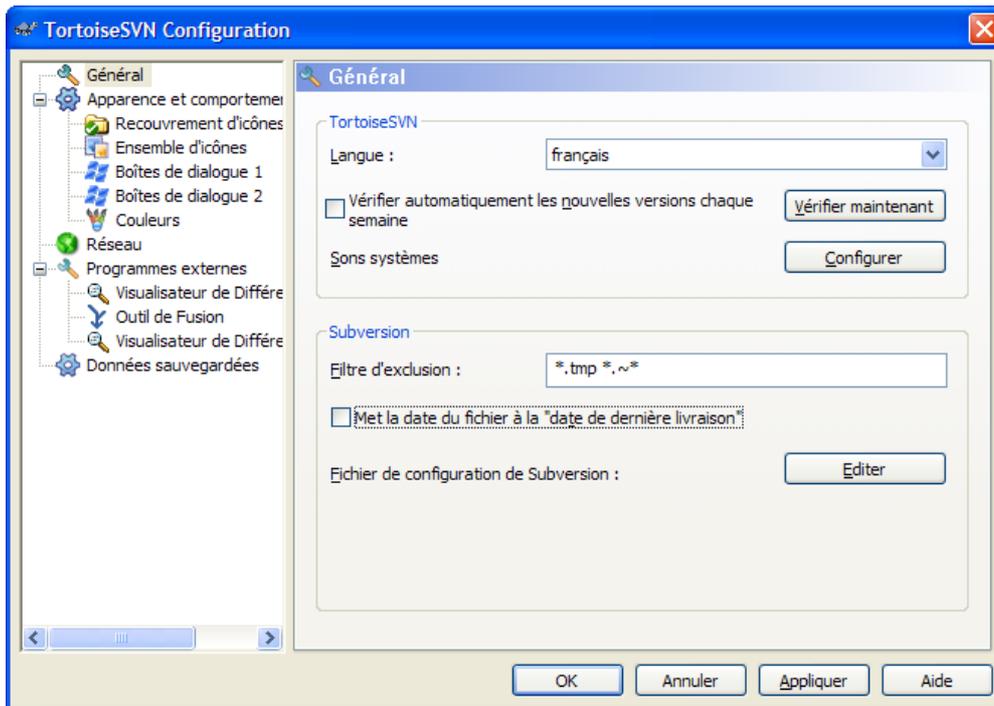
### Aucun visualiseur dans l'explorateur de référentiel

Because the repo viewer integration depends upon accessing subversion properties, you will only see the results when using a checked out working copy. Fetching properties remotely is a slow operation, so you will not see this feature in action from the repo browser.

## 4.30. Configuration de TortoiseSVN

Pour découvrir à quoi servent les réglages, laissez simplement votre pointeur de souris une seconde sur la saisie/coche... et une info-bulle utile apparaîtra.

### 4.30.1. Configuration générale



**Figure 4.49. La boîte de dialogue Configuration, page Général**

Cette boîte de dialogue vous permet de spécifier votre langue préférée et la configuration spécifique à Subversion.

#### Langue

Sélectionne la langue de votre interface utilisateur. À quoi d'autre vous attendiez-vous ?

#### Vérifier automatiquement les nouvelles versions chaque semaine

If checked, TortoiseSVN will contact its download site once a week to see if there is a newer version of the program available. Use **Check now** if you want an answer right away. The new version will not be downloaded; you simply receive an information dialog telling you that the new version is available.

#### Sons système

TortoiseSVN a trois sons personnalisés qui sont installés par défaut.

- Erreur
- Avis
- Avertissement

Vous pouvez choisir des sons différents (ou désactiver ces sons complètement) en utilisant le Panneau de Configuration Windows. **Configurer** est un raccourci vers le Panneau de Configuration.

#### Modèle d'exclusion global

Les modèles d'exclusion globaux sont utilisés pour empêcher des fichiers non versionnés d'apparaître, par exemple dans la boîte de dialogue de livraison. Les fichiers correspondant aux modèles sont aussi ignorés lors d'un import. Ignorez des fichiers ou des répertoires en saisissant les noms ou les extensions. Les modèles sont séparés par des espaces, par exemple \*/bin \*/obj \*.bak \*.~?? \*.jar \*. [Tt]mp. Ces modèles ne doivent contenir aucun séparateur de chemin. Notez également qu'il n'y a aucun moyen de différencier les fichiers des répertoires. Lisez

Section 4.13.1, « L'utilisation des pattern matching dans la liste des fichier à ignorer » pour plus d'information sur la syntaxe de correspondance des modèles.

Notez que les modèles d'exclusion que vous spécifiez ici affecteront aussi les autres clients Subversion fonctionnant sur votre PC, y compris le client en ligne de commande.



### Attention

Si vous utilisez le fichier de configuration de Subversion pour définir un modèle `global-ignores`, il ignorera les réglages que vous faites ici. Le fichier de configuration de Subversion est accessible en utilisant **Éditer** comme décrit ci-dessous.

Ce modèle d'exclusion affectera tous vos projets. Ce n'est pas versionné, donc il n'affectera pas les autres utilisateurs. En comparaison, vous pouvez aussi utiliser la propriété versionnée `svn:ignore` pour exclure des fichiers ou des répertoires du contrôle de version. Lisez [Section 4.13, « Ignorer des fichiers et des répertoires »](#) pour plus d'informations.

Remplace la date des fichiers par la « date de dernière livraison »

Cette option indique à TortoiseSVN de mettre la date des fichiers à la date de dernière livraison lors d'une extraction ou d'une mise à jour. Autrement TortoiseSVN utilisera la date actuelle. Si vous développez des logiciels, il est généralement mieux d'utiliser la date actuelle parce que les systèmes de compilation regardent normalement les dates pour décider quels fichiers ont besoin d'être compilés. Si vous utilisez « la date de dernière livraison » et revenez à une révision de fichier plus vieille, votre projet peut ne pas se compiler comme vous vous y attendez.

Fichier de configuration de Subversion

Utilisez **Éditer** pour éditer directement le fichier de configuration de Subversion. Quelques réglages ne peuvent pas être modifiés directement par TortoiseSVN et ont besoin d'être définis ici à la place. Pour plus d'informations concernant le fichier `config` de Subversion, regardez la [Runtime Configuration Area](http://svnbook.red-bean.com/en/1.5/svn.advanced.confarea.html) [http://svnbook.red-bean.com/en/1.5/svn.advanced.confarea.html]. La section sur [Automatic Property Setting](http://svnbook.red-bean.com/en/1.5/svn.advanced.props.html#svn.advanced.props.auto) [http://svnbook.red-bean.com/en/1.5/svn.advanced.props.html#svn.advanced.props.auto] a beaucoup d'intérêt et c'est ce qui se configure ici. Notez que Subversion peut lire l'information de configuration depuis plusieurs endroits et vous devez savoir lequel a la priorité. Référez-vous à [Configuration and the Windows Registry](http://svnbook.red-bean.com/en/1.5/svn.advanced.confarea.html#svn.advanced.confarea.windows-registry) [http://svnbook.red-bean.com/en/1.5/svn.advanced.confarea.html#svn.advanced.confarea.windows-registry] pour en savoir plus.

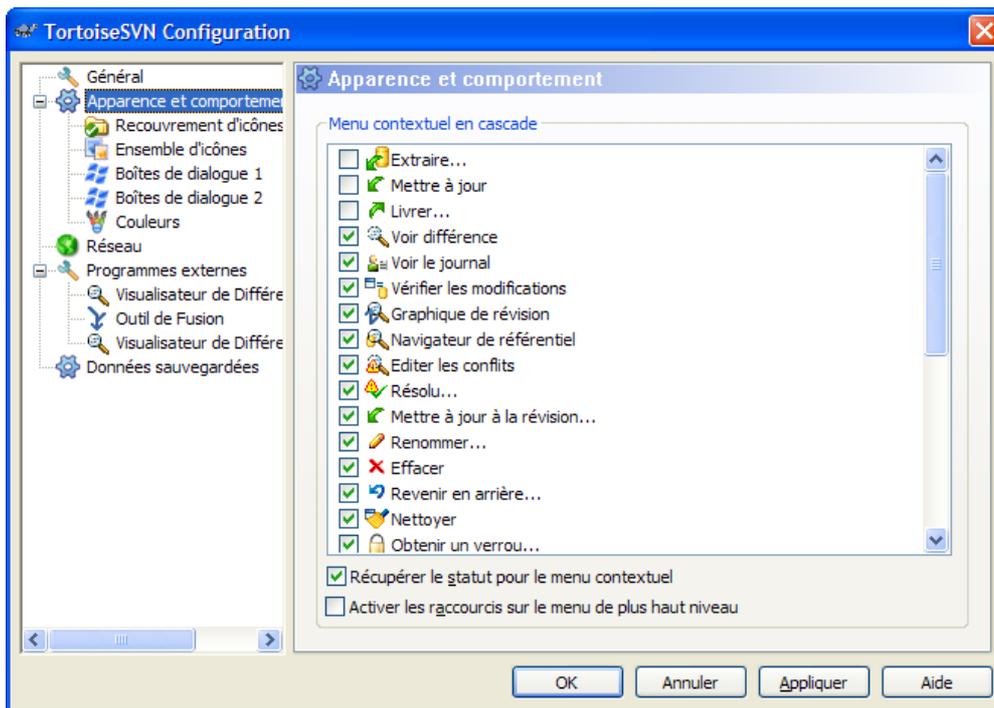
Utiliser les répertoires `_svn` au lieu de `.svn`

VS.NET quand il est utilisé avec des projets web ne peut pas manipuler les répertoires `.svn` que Subversion utilise pour stocker son information interne. Ce n'est pas un bug de Subversion. Le bug est dans VS.NET et les extensions frontpage qu'il utilise. Lisez [Section 4.30.11, « Dossiers de travail de Subversion »](#) pour en savoir plus sur cet incident.

Si vous voulez changer le comportement de Subversion et de TortoiseSVN, vous pouvez utiliser cette case à cocher pour définir la variable d'environnement qui contrôle cela.

Vous devriez noter que changer cette option ne convertira pas automatiquement les copies de travail existantes pour utiliser le nouveau répertoire d'administration. Vous devrez le faire vous-même en utilisant un script (Voir notre FAQ) ou en extrayant simplement une copie de travail récente.

#### 4.30.1.1. Paramètres du Menu Contextuel



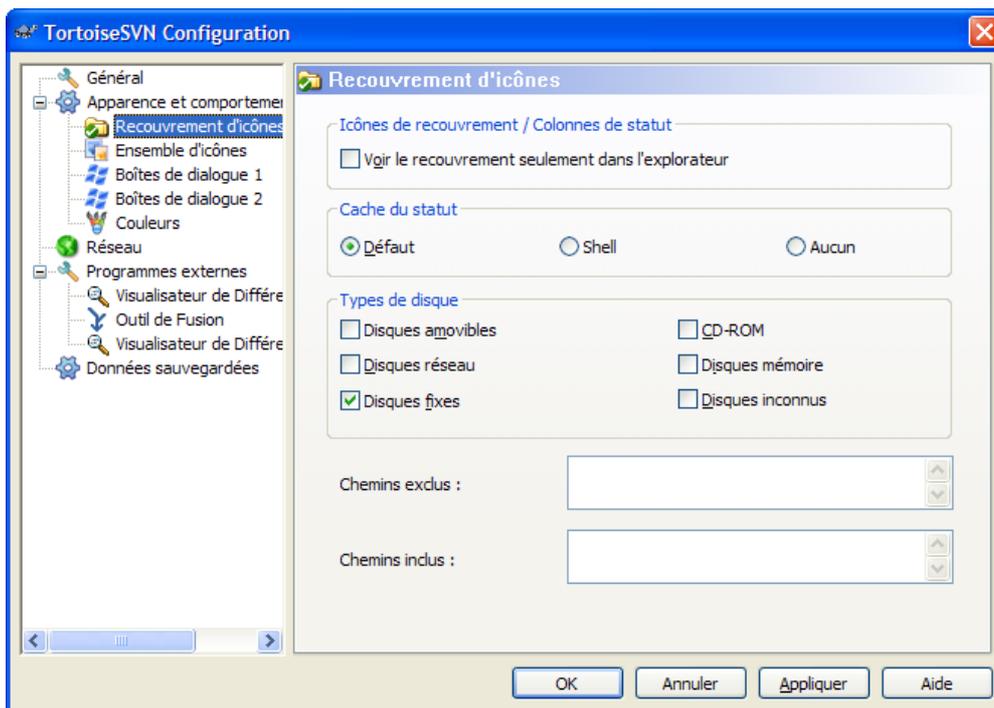
**Figure 4.50. La boîte de dialogue de Configuration, Page Menu Contextuel**

Cette page vous permet de spécifier quelles entrées du menu contextuel de TortoiseSVN s'afficheront dans le menu contextuel principal et lesquelles apparaîtront dans le sous-menu de TortoiseSVN. Par défaut, la plupart des éléments sont décochés et apparaissent dans le sous-menu.

Il existe un cas spécial pour **Obtenir un verrou**. Vous pouvez bien sûr le promouvoir au niveau supérieur en utilisant la liste ci-dessus mais puisque la plupart des fichiers n'ont pas besoin du verrouillage, cela ajoute juste du désordre. Cependant, un fichier avec la propriété `svn:needs-lock` nécessite cette action à chaque fois qu'il est modifié, donc dans ce cas, il est très utile de l'avoir au premier niveau. Ici, cocher la case signifie que lorsqu'un fichier ayant la propriété `svn:needs-lock` est sélectionné, **Obtenir un verrou** apparaîtra toujours au premier niveau.

S'il y a quelques chemins sur votre ordinateur où vous ne voulez pas qu'apparaisse le menu contextuel de TortoiseSVN, vous pouvez les lister dans la zone en bas.

### 4.30.1.2. Réglages des boîtes de dialogues TortoiseSVN 1



**Figure 4.51. La boîte de dialogue Configuration, page Boîtes de dialogue 1**

Cette boîte de dialogue vous permet de configurer certaines des boîtes de dialogue de TortoiseSVN de la façon que vous préférez.

#### Nombre par défaut de commentaires

Limite le nombre de commentaires que TortoiseSVN va d'abord chercher quand vous choisissez TortoiseSVN → Voir le journal Utile avec des connexions serveur lentes. Vous pouvez toujours utiliser Afficher tout ou 100 suivants pour obtenir plus de messages.

#### Police des commentaires

Sélectionne le type et la taille de la police de caractères utilisée pour afficher le commentaire lui-même dans le panneau du milieu de la boîte de dialogue du Journal de révision et lors de la rédaction des commentaires dans la boîte de dialogue Livrer.

#### Format court des dates dans les commentaires

Si les longs messages standards prennent trop place sur votre écran, utilisez le format court.

#### Vous pouvez double cliquer sur la liste des messages de log pour comparer avec la version précédente

Si vous vous trouvez régulièrement à comparer les révisions dans le panneau supérieur de la fenêtre de commentaires, vous pouvez utiliser cette option pour exécuter cette action suite à un double-click. Elle n'est pas activée par défaut car consulter les différences est souvent un long processus et beaucoup de personnes préfèrent éviter l'attente suite à un double-click accidentel.

#### Boîte de dialogue de progression

TortoiseSVN peut fermer automatiquement toutes les boîtes de dialogues de progression quand l'action s'est terminée sans erreur. Ce réglage vous permet de choisir les conditions pour fermer les boîtes de dialogues. Le réglage par défaut est Fermeture manuelle ce qui vous permet de passer en revue tous les messages et de contrôler ce qui s'est passé. Cependant, vous pouvez décider que vous voulez ignorer quelques types de message et vouloir que la boîte de dialogue se ferme automatiquement s'il n'y a aucun changement critique.

Fermeture automatique s'il n'a y pas eu de fusions, d'ajouts ou de suppression signifie que la boîte de dialogue de progression se fermera s'il y a que de simples mises à jour, mais si les

changements du référentiel ont été fusionnés avec les vôtres, ou si des fichiers ont été ajoutés ou supprimés, la boîte de dialogue restera ouverte. Elle restera aussi ouverte s'il n'y a pas eu de conflits ou d'erreurs pendant l'opération.

**Fermeture automatique pour les opérations locales** signifie que la boîte de dialogue de progression se fermera comme pour **Fermeture automatique s'il n'a y pas eu de fusions, d'ajouts ou de suppression** mais seulement pour les opérations locales comme ajouter des fichiers ou revenir en arrière. Pour les opérations distantes, la boîte de dialogue restera ouverte.

**Fermeture automatique s'il n'y a pas de conflit** assoupli les critères un peu plus et fermera la boîte de dialogue même s'il y a eu des fusions, des ajouts ou des suppressions. Cependant, s'il y a des conflits ou des erreurs, la boîte de dialogue reste ouverte.

**Fermeture automatique s'il n'y a pas d'erreur** ferme toujours la boîte de dialogue même s'il y a eu des conflits. La seule condition qui maintient la boîte de dialogue ouverte est un cas d'erreur, qui se produit quand Subversion est incapable d'achever la tâche. Par exemple, une mise à jour échoue parce que le serveur est inaccessible, ou une livraison échoue parce que la copie de travail est périmée.

**Utiliser la poubelle lors d'un retour en arrière**

Lorsque vous annulez des modifications locales, vos changements sont oubliés. TortoiseSVN vous donne un filet de sécurité supplémentaire en envoyant le fichier modifié à la corbeille avant de rendre la copie primitive. Si vous préférez ne pas passer par la corbeille, décochez cette option.

**Utiliser l'URL de la WC comme valeur par défaut pour l'URL « From: »**

Dans la boîte de dialogue de fusion, le comportement par défaut est de mémoriser l'URL **De** : entre les fusions. Cependant, certaines personnes aiment exécuter les fusions depuis différents endroits de leur hiérarchie et trouvent plus facile de partir avec l'URL de la copie de travail courante. Elle peut alors être éditée pour se référer à un chemin parallèle sur une autre branche.

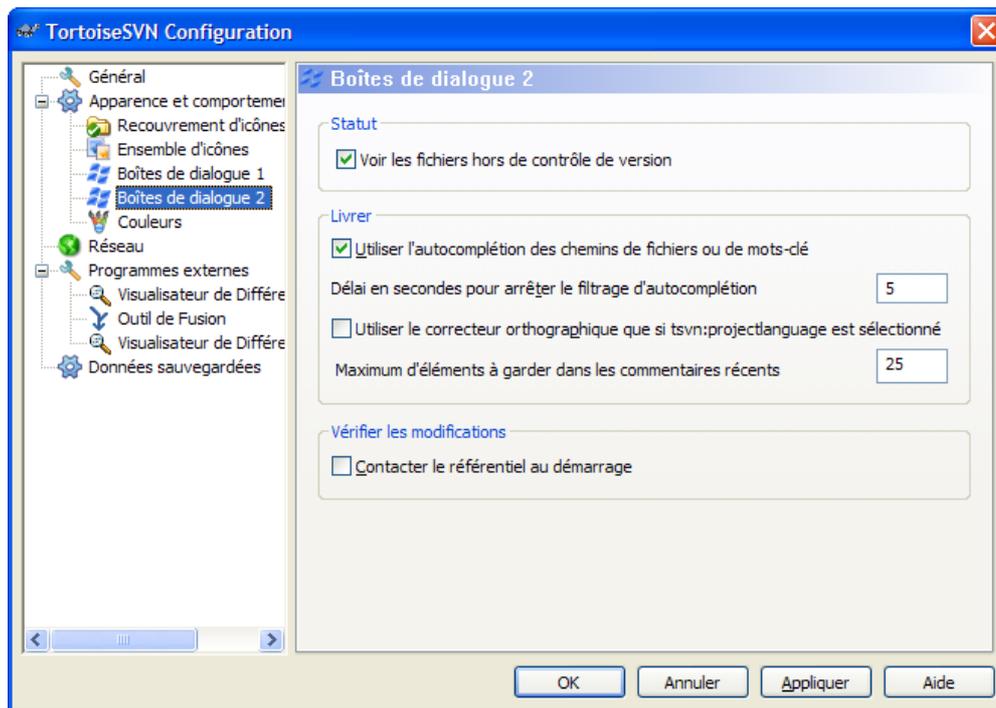
**Chemin d'extraction par défaut**

Vous pouvez spécifier le chemin par défaut pour les extractions. Si vous gardez toutes vos extractions à un seul endroit, il est utile d'avoir le disque et le dossier pré-rempli pour que vous n'ayez plus qu'à ajouter le nouveau nom du dossier à la fin.

**URL d'extraction par défaut**

Vous pouvez aussi spécifier le chemin par défaut l'URL par défaut des extractions. Si vous extrayez souvent des sous-projets d'un très gros projet, il peut être utile d'avoir l'URL pré-remplie pour que vous n'ayez plus qu'à ajouter le nom du sous-projet à la fin.

### 4.30.1.3. Réglages des boîtes de dialogues TortoiseSVN 2



**Figure 4.52. La boîte de dialogue Configuration, page Boîtes de dialogue 2**

#### Parcourir récursivement les répertoires non versionnés

Si cette case est cochée (l'état par défaut), alors à chaque fois que le statut d'un dossier non versionné est affiché dans les boîtes de dialogue **Ajouter**, **Livrer** ou **Vérifier les modifications**, tous les fichiers et tous les dossiers enfants sont aussi affichés. Si vous décochez cette case, seul le parent non versionné est affiché. Décocher réduit le désordre dans ces boîtes de dialogue. Dans ce cas, si vous sélectionnez un dossier non versionné à **Ajouter**, il est ajouté récursivement.

#### Utiliser la complétion automatique des chemins de fichiers et des mots-clé

La boîte de dialogue de livraison inclut une fonction pour analyser syntaxiquement la liste de noms de fichier à livrer. Quand vous tapez les 3 premières lettres d'un élément dans la liste, la boîte de complétion automatique s'ouvre et vous pouvez appuyer sur **Entrée** pour compléter le nom du fichier. Cochez la case pour activer cette fonctionnalité.

#### Délai en secondes après lequel arrêter le parcours des données de complétion automatique

L'analyseur syntaxique d'autocomplétion peut être assez lent s'il y a beaucoup de gros fichiers à vérifier. Ce délai empêche la boîte de dialogue de livraison de se bloquer trop longtemps. Si vous manquez d'informations d'autocomplétion importantes, vous pouvez étendre le délai.

#### Utiliser le correcteur orthographique que si `tsvn:projectlanguage` est défini

Si vous ne voulez pas utiliser le vérificateur d'orthographe pour toutes les livraisons, cochez cette case. Le vérificateur d'orthographe sera toujours activé quand les propriétés du projet l'exigent.

#### Maximum d'éléments à garder dans les commentaires récents

Lorsque vous tapez un commentaire dans la boîte de dialogue de livraison, TortoiseSVN le stocke pour une éventuelle réutilisation plus tard. Par défaut, il conserve les 25 derniers commentaires pour chaque référentiel, mais vous pouvez personnaliser ce nombre ici. Si vous avez beaucoup de référentiels différents, vous devriez le réduire pour éviter de remplir votre registre.

Notez que cette configuration s'applique uniquement aux messages que vous créez sur cet ordinateur. Cela n'a aucune relation avec le cache des commentaires.

Ré-ouvrir la boîte de dialogue de livraison après qu'une livraison ait échoué

Quand une livraison échoue pour une raison particulière (la copie de travail doit être mise à jour, les hooks pre-commit refusent la livraison, erreur réseau, etc), vous pouvez choisir cette option pour conserver la boîte de dialogue Livrer ouverte prête à recommencer. Cependant, vous devez prendre conscience que cela peut causer des problèmes. Si l'échec signifie que vous devez mettre à jour et que la mise à jour entraîne des conflits, vous devrez d'abord les résoudre.

Sélectionner les éléments automatiquement

Le comportement normal dans la fenêtre de livraison est la sélection automatique de tous les éléments modifiés (et versionnés) en vue d'une livraison. Si vous préférez démarrer sans aucune sélection et choisir les éléments manuellement, décochez cette case.

Contacteur le référentiel au démarrage

La boîte de dialogue Vérifier les modifications vérifie la copie de travail par défaut et entre seulement en contact avec le référentiel quand vous cliquez sur **Vérifier le référentiel**. Si vous voulez toujours vérifier le référentiel, vous pouvez utiliser ce réglage pour que cette action se fasse automatiquement.

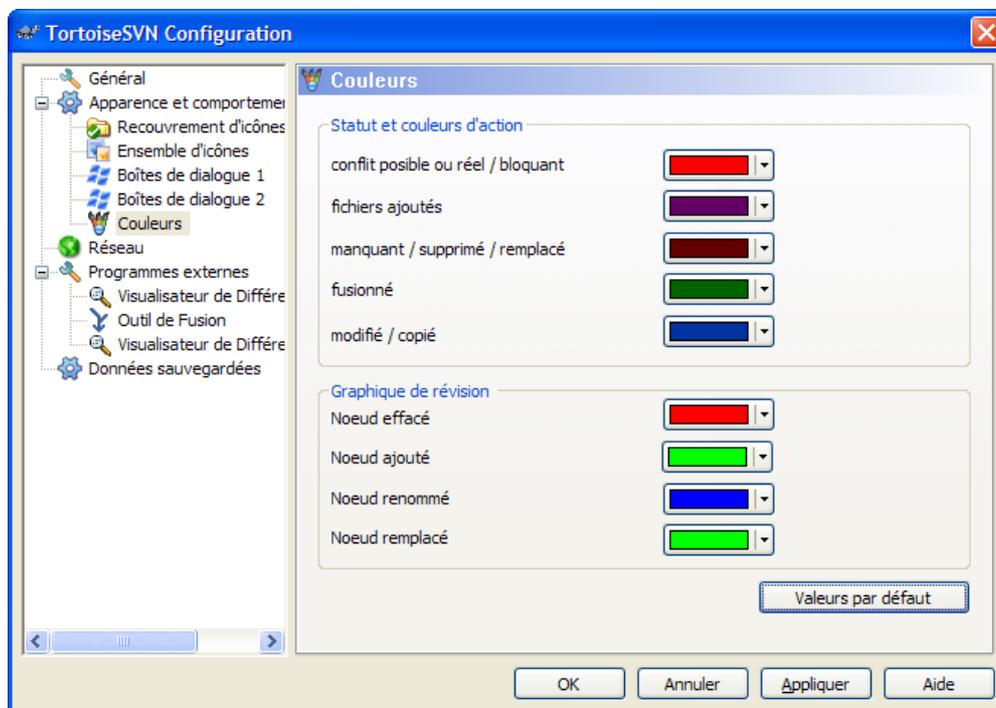
Afficher la boîte de dialogue de verrouillage avant de verrouiller des fichiers

Lorsque vous sélectionnez un ou plusieurs fichiers, puis que vous utilisez TortoiseSVN → **Verrouillage** pour retirer un verrou sur ces fichiers, sur certains projet il est demandé d'ajouter un message expliquant pourquoi vous avez verrouillé ces fichiers. Si vous n'utilisez pas de message de verrouillage, vous pouvez décocher la case afin de passer cette fenêtre et de verrouiller les fichiers immédiatement.

Si vous utilisez la commande verrouiller sur un dossier, une fenêtre s'ouvrira vous permettant de sélectionner d'autres fichiers à verrouiller.

Si votre projet a la propriété `tsvn:lockmsgminsize`, vous verrez tout de même la fenêtre de verrouillage parce que le projet a *besoin* de messages de verrouillage.

#### 4.30.1.4. Configuration des couleurs de TortoiseSVN



**Figure 4.53. La boîte de dialogue Configuration, page Couleurs**

Cette boîte de dialogue vous permet de configurer les couleurs de texte utilisées dans les boîtes de dialogue de TortoiseSVN de la façon que vous préférez.

**Conflit possible ou réel / bloquant**

Un conflit s'est produit pendant la mise à jour, ou peut arriver pendant une fusion. La mise à jour est entravée par un fichier/dossier non versionné existant du même nom qu'un fichier versionné.

Cette couleur est aussi utilisée pour des messages d'erreur dans les boîtes de dialogues de progression.

**Fichiers ajoutés**

Éléments ajoutés au référentiel.

**Manquant / supprimé / remplacé**

Éléments supprimés du référentiel, manquants de la copie de travail, ou supprimés de la copie de travail et remplacés par d'autres fichiers du même nom.

**Fusionné**

Changements du référentiel fusionnés avec succès dans la CdT sans créer de conflits.

**Modifié / copié**

Ajoutés avec l'historique, ou chemins copiés dans le référentiel. Aussi utilisé dans la boîte de dialogue de journal pour les entrées qui incluent des éléments copiés.

**Noeud Supprimé**

Un élément qui a été supprimé du référentiel.

**Noeud ajouté**

Un élément ajouté au référentiel, par un ajout, une copie ou un déplacement.

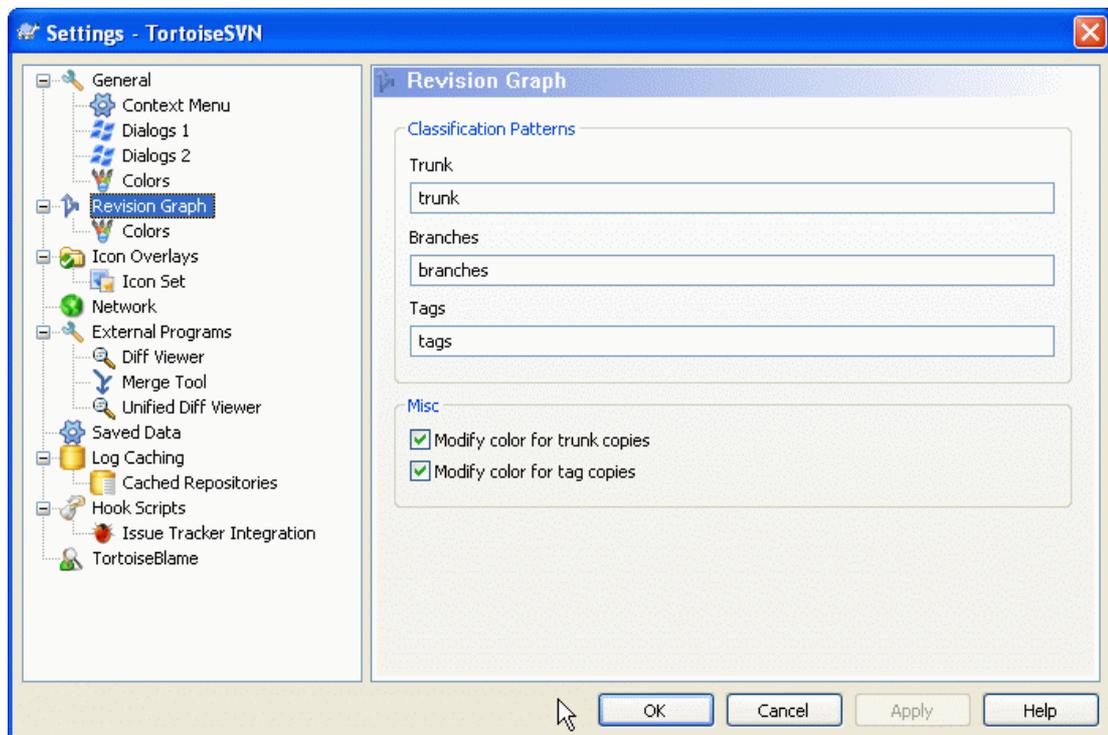
**Noeud renommé**

Un élément qui a été renommé dans le référentiel.

**Noeud remplacé**

L'élément original a été supprimé et un nouvel élément avec le même nom le remplace.

### 4.30.2. Options du Graphe des Révisions



**Figure 4.54. La boîte de dialogue Configuration, page graphique de révision**

### Filtres de classification

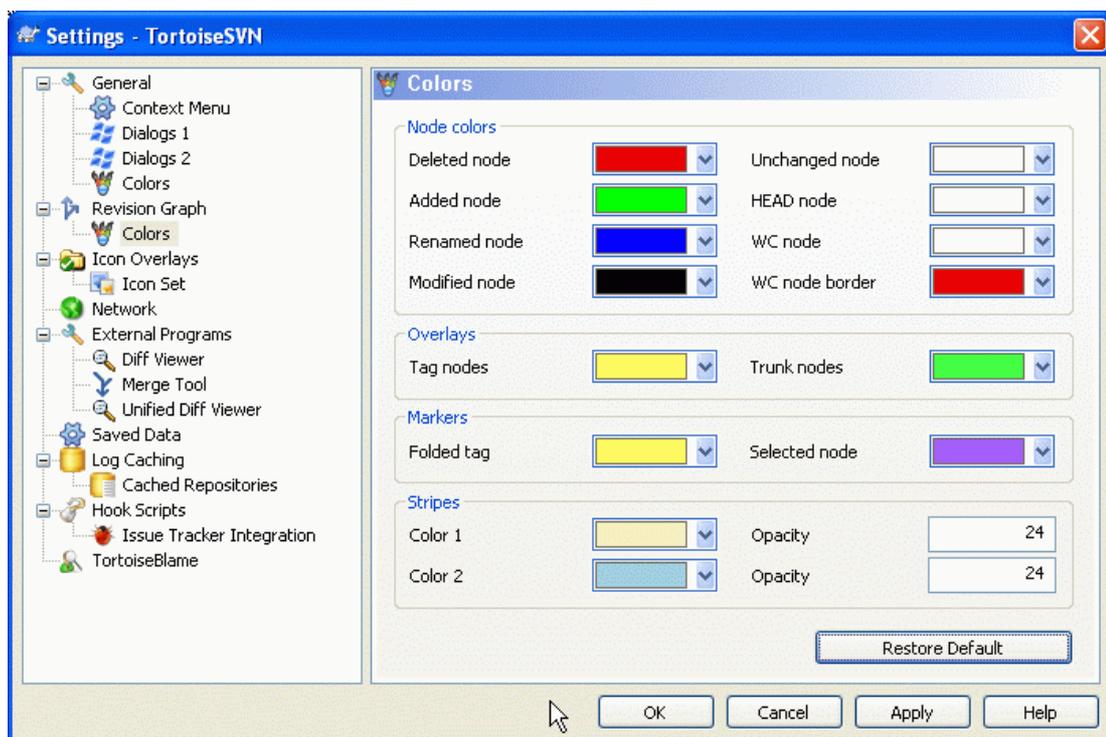
Le graphe de révision tente de montrer une image plus claire de la structure de votre dépôt en distinguant le trunk, les branches et les tags. Puisqu'il n'existe pas de telle classification construite dans Subversion, cette information est extraite à partir des chemins. La configuration par défaut suppose que vous utilisez les noms anglais conventionnels comme il est suggéré dans la documentation Subversion, mais bien sûr, votre utilisation peut varier.

Spécifiez les patrons utilisés pour reconnaître ces chemins dans les trois champs prévus à cette effet. Les patrons sont insensibles à la casse, mais vous devez les spécifier en minuscule. Les caractères spéciaux \* et ? fonctionneront comme d'habitude. Vous pouvez utiliser ; pour séparer plusieurs patrons. N'incluez pas d'espaces supplémentaires comme ils seraient inclus dans la spécification correspondante.

### Modifier les Couleurs

Les couleurs sont utilisées dans le graphe de révision pour indiquer le type de noeud, c'est à dire si le noeud a été ajouté, supprimé ou supprimé. Afin de faciliter la classification des noeuds, vous pouvez permettre au graphe de mélanger les couleurs pour donner une identification à la fois au type et à la classification des noeuds. Si la case est cochée, le mélange sera utilisé. Dans le cas contraire, la couleur est utilisée pour indiquer uniquement le type du noeud. Utilisez cette fenêtre de sélection de couleurs pour allouer les couleurs spécifiques utilisées.

#### 4.30.2.1. Couleurs du Graphes de Révision



**Figure 4.55. La boîte de dialogue Configuration, page des couleur du graphe de révision**

Cette page vous permet de configurer les couleurs utilisées. Notez bien que les couleurs spécifiées ici sont des couleurs pleines. La plupart des noeuds sont colorés en utilisant un mélange de la couleur de leur type, de celle du fond d'écran et éventuellement, de la couleur de classification

#### Noeud Supprimé

Éléments ayant été supprimés et non copiés ailleurs dans la même révision.

#### Noeud Ajouté

Éléments ajoutés récemment, ou copiés (ajout avec l'historique).

**Noeud Renommé**

Éléments supprimés d'un endroit et ajoutés ailleurs dans une même révision.

**Noeud Modifié**

Modification simple sans ajout ni suppression.

**Noeud inchangé**

Peut être utilisée pour montrer la révision utilisée en tant que source d'une copie, même lorsqu'aucune modification (de l'élément représenté sur le graphe) n'a eut lieu lors de cette révision.

**Noeud de tête**

Révision de tête courante dans le référentiel

**Noeud de la CdT**

Si vous choisissez de montrer, sur le graphe, un noeud supplémentaire pour votre copie de travail modifiée, attachée à la révision de la dernière livraison, utilisez cette couleur.

**Bordure du noeud CdT**

Si vous choisissez d'afficher si la copie de travail est modifiée, utilisez cette bordure de couleur sur le noeud WC lorsque des modifications sont trouvées.

**Noeud tag**

Les noeuds classifiés comme étant des tags peuvent être nuancés avec cette couleur.

**Noeud racine**

Les noeuds classifiés comme étant des trunk (racines) peuvent être nuancés avec cette couleur.

**Marqueurs de Tags Fermés**

Si vous avez l'habitude de replier les étiquettes pour économiser de l'espace, elles sont marquées sur la copie source en utilisant un bloc de cette couleur

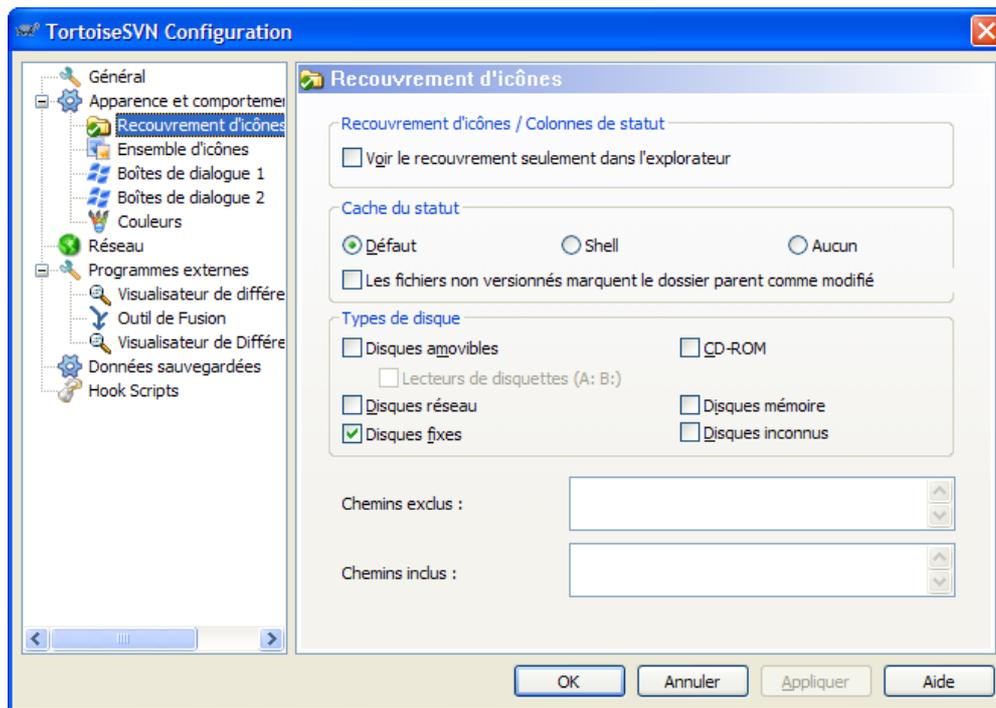
**Marqueurs du Noeud Sélectionné**

Lorsque vous cliquez sur un noeud pour le sélectionner, le marqueur utilisée pour indiquer la sélection est un bloc de cette couleur.

**Rayures**

Ces couleurs sont utilisées lorsque le graphe est divisé en sous-arbres et que le fond est coloré en rayures alternées pour aider à choisir les arbres séparés.

### 4.30.3. Configuration du recouvrement d'icônes



**Figure 4.56. La Boîte de Dialogue Configuration, Page des Icônes de Recouvrement**

Cette page vous permet de choisir à quels éléments TortoiseSVN doit associer des icônes de recouvrement

Par défaut, les icônes de recouvrement et les menus contextuels apparaîtront dans toutes les boîtes de dialogue Ouvrir/Enregistrer comme dans l'explorateur Windows. Si vous voulez qu'elles n'apparaissent *que* dans l'explorateur Windows, cochez la case *Voir les recouvrements et le menu contextuel seulement dans l'explorateur*.

Les éléments ignorés ainsi que les éléments non versionnés ne donnent généralement généralement pas de recouvrement. Si vous désirez afficher un recouvrement dans ces cas-là, cochez les cases.

Vous pouvez aussi choisir de marquer les répertoires comme modifiés s'ils contiennent des éléments non versionnés. Cela peut être utile pour vous rappeler que vous avez créé de nouveaux fichiers qui ne sont pas encore versionnés. Cette option n'est disponible que lorsque vous utilisez l'option de statut de cache *par défaut* (voir ci-dessous)

Puisque cela prend du temps pour récupérer le statut d'une copie de travail, TortoiseSVN utilise un cache pour stocker le statut pour que l'explorateur ne soit pas trop bloqué en affichant les recouvrements. Vous pouvez choisir quel type de cache TortoiseSVN devrait utiliser selon votre système et la taille de votre copie de travail ici :

#### Défaut

Met en cache toute l'information de statut dans un processus séparé (TSVNCache.exe). Ce processus observe tous les disques pour les changements et va chercher à nouveau le statut si les fichiers à l'intérieur d'une copie de travail sont modifiés. Le processus s'exécute avec la priorité la plus faible possible pour que les autres programmes ne soient pas ralentis à cause de lui. Cela signifie aussi que l'information de statut n'est pas *en temps réel* mais cela peut prendre quelques secondes pour que les recouvrements changent.

Avantage : les recouvrements montrent le statut récursivement, c'est-à-dire si un fichier est modifié dans les profondeurs d'une copie de travail, tous les dossiers jusqu'à la racine de la copie de travail

montreront aussi le recouvrement modifié. Et puisque le processus peut envoyer des notifications au shell, les recouvrements sur l'arborescence gauche changent aussi.

Inconvénient : le processus fonctionne constamment, même si vous ne travaillez pas sur vos projets. Il utilise aussi environ 10-50 Mo de RAM selon le nombre et la taille de vos copies de travail.

#### Shell

La mise en cache est faite directement à l'intérieur de la dll d'extension du shell, mais seulement pour le dossier actuellement visible. Chaque fois vous naviguez à un autre dossier, l'information de statut est parcourue de nouveau.

Avantage : a seulement besoin de très peu de mémoire (autour de 1 MO de RAM) et peut montrer le statut *en temps réel*.

Inconvénient : puisque un seul dossier est mis en cache, les recouvrements ne montrent pas le statut récursivement. Pour de grandes copies de travail, cela peut prendre plus de temps pour montrer un dossier dans l'explorateur qu'avec le cache par défaut. Aussi la colonne de type mime n'est pas disponible.

#### Aucun

Avec ce réglage, TortoiseSVN ne va pas du tout chercher le statut dans l'Explorateur. De ce fait, les fichiers n'ont pas de recouvrement et les dossiers ont seulement un recouvrement 'normal' s'ils sont versionnés. Aucun autre recouvrement n'est affiché et aucune colonne supplémentaire n'est disponible non plus.

Avantage : n'utilise absolument aucune mémoire supplémentaire et ne ralentit pas du tout l'Explorateur en parcourant.

Inconvénient : L'information de statut des fichiers et des dossiers n'est pas affichée dans l'Explorateur. Pour voir si vos copies de travail sont modifiées, vous devez utiliser la boîte de dialogue « Vérifier les modifications ».

Le groupe suivant vous permet de choisir les quels éléments de stockage montreront les recouvrements. Par défaut, seuls les disques durs sont sélectionnés. Vous pouvez même désactiver tous les recouvrements d'icône, mais qu'y-a-t-il d'amusant à cela ?

Les disques réseau peuvent être très lents, dont par défaut les icônes ne sont pas affichées pour les copies de travail situées dans des dossiers partagés.

Les disques USB Flash semblent être un cas particulier en cela que le type de disque est identifié par le périphérique lui-même. Certains apparaissent comme des disques fixes et d'autres comme des disques amovibles.

The **Exclude Paths** are used to tell TortoiseSVN those paths for which it should *not* show icon overlays and status columns. This is useful if you have some very big working copies containing only libraries which you won't change at all and therefore don't need the overlays. For example:

f:\développement\SVN\Subversion désactivera les recouvrements *seulement* sur ce dossier spécifique. Vous pouvez toujours voir les recouvrements sur tous les fichiers et tous les dossiers à l'intérieur de ce dossier.

f:\development\SVN\Subversion\* désactivera les recouvrements sur *tous* les fichiers et dossiers dont le chemin commence par f:\development\SVN\Subversion. Cela signifie que vous ne verrez pas de recouvrement pour aucun des fichiers et des dossiers en-dessous de ce chemin.

La même chose s'applique aux **Chemins inclus**. Sauf que pour ces chemins, les recouvrements s'affichent même s'ils sont désactivés pour ce type de disque spécifique, ou par un chemin exclus indiqué au-dessus.

Users sometimes ask how these three settings interact, and the definitive answer is:

```
if (path is in include list)
  show overlays
if (path is allowed drive type) AND (path is not in exclude list)
  show overlays
```

The include list *always* makes the overlays show. Otherwise, overlays are shown for all marked drive types *unless* the path is excluded.

TSVNCache.exe utilise aussi ces chemins pour limiter son balayage. Si vous voulez qu'il ne regarde que dans des dossiers particuliers, désactivez tous les types de disque et incluez seulement les dossiers que vous voulez spécifiquement être parcourus.



## Exclure les Lecteurs SUBST

Il est souvent pratique d'utiliser un lecteur SUBST pour accéder à vos copies de travail, par exemple en utilisant la commande

```
subst T: C:\TortoiseSVN\trunk\doc
```

. Cependant, cela peut empêcher la mise à jour des recouvrements, car TSVNCache ne recevra qu'une notification quand un fichier change, ce qui est normalement pour le chemin d'origine. Cela signifie que vos recouvrements sur le chemin subst pourraient ne jamais être mis à jour.

Un moyen simple de contourner cela est d'empêcher le chemin d'origine de montrer les recouvrements, qui se présentent alors sur le chemin subst à la place.

Parfois, vous excluez des zones qui contiennent des copies de travail, ce qui évitera à TSVNCache de scanner et vérifier les changements, mais vous voudrez quand même une indication visuelle montrant que ces dossiers sont versionnés. La case à cocher **Montrer les dossiers exclus comme «normaux»** vous permet de le faire. Avec cette option, les dossiers versionnés dans n'importe quelle zone exclue (type de lecteur non coché, ou spécifiquement exclu) seront affichés comme normaux et mis à jour, avec une coche verte. Cela vous rappelle que vous êtes dans une copie de travail, même si les recouvrements de dossier ne sont pas corrects. Les fichiers ne présentent jamais de recouvrement. Notez que les menus contextuels fonctionnent encore, même si les recouvrements ne sont pas affichés.

Une exception particulière à cela est que les lecteurs A : et B : ne sont jamais pris en compte dans l'option **Montrer les dossiers exclus comme 'normaux'**. C'est parce que Windows est obligé de chercher sur le disque, ce qui peut entraîner un retard de plusieurs secondes lors du démarrage d'Explorer, même si votre PC possède un lecteur de disquette.

### 4.30.3.1. Sélection du jeu d'icônes

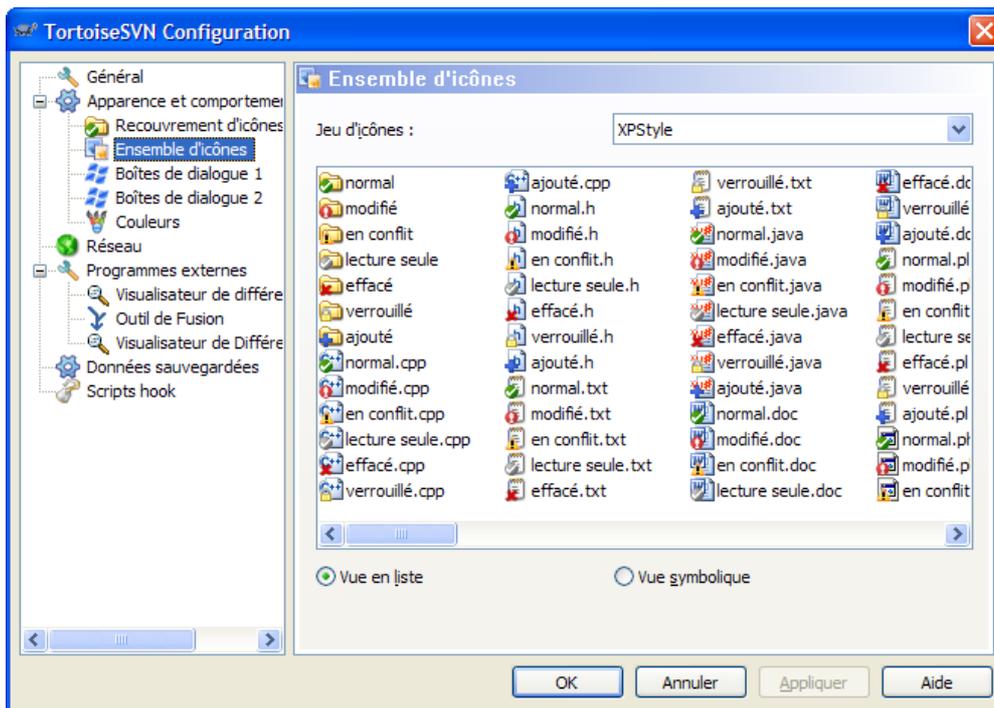


Figure 4.57. La boîte de dialogue Configuration, page Ensemble d'icônes

Vous pouvez changer le jeu d'icônes de recouvrement pour celui que vous aimez le plus. Notez que si vous changez le jeu de recouvrement, vous devriez redémarrer votre ordinateur pour que les changements prennent effet.

### 4.30.4. Configuration du réseau

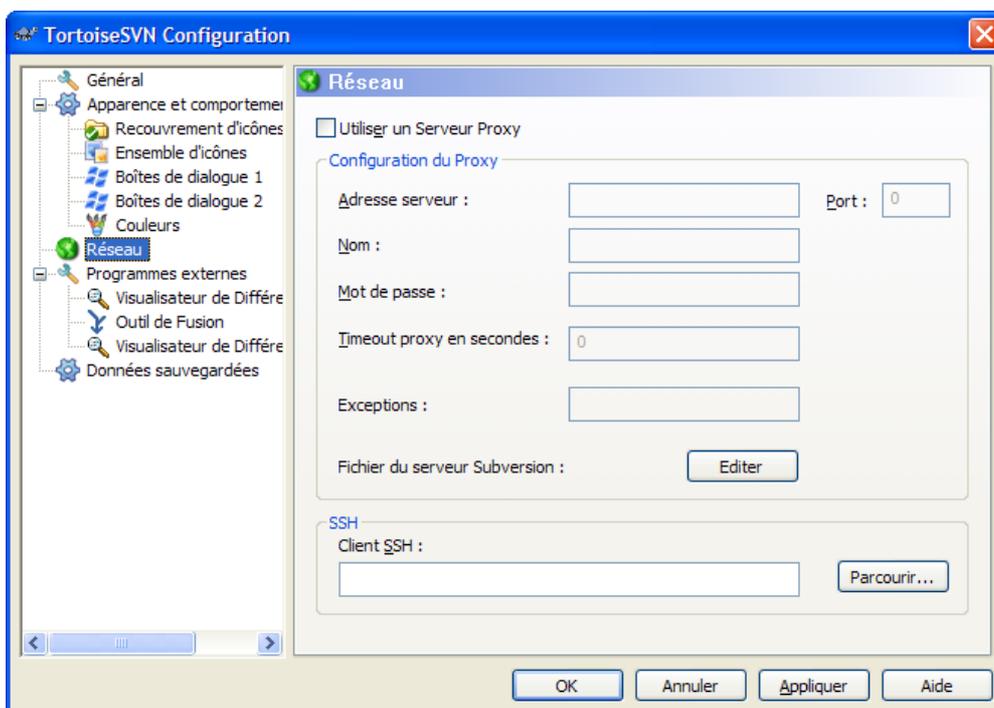


Figure 4.58. La boîte de dialogue Configuration, page Réseau

Vous pouvez ici configurer votre serveur proxy, si vous en avez besoin pour passer le pare-feu de votre société.

Si vous devez mettre en place des réglages de proxy par référentiel, vous devez utiliser le fichier `servers` de Subversion. Utilisez **Éditer** pour y accéder directement. Consultez [Runtime Configuration Area](http://svnbook.red-bean.com/en/1.5/svn.advanced.confarea.html) [http://svnbook.red-bean.com/en/1.5/svn.advanced.confarea.html] pour plus de détails sur la façon d'utiliser ce fichier.

Vous pouvez aussi spécifier quel programme TortoiseSVN devrait utiliser pour établir une connexion sécurisée à un référentiel `svn+ssh`. Nous vous recommandons d'utiliser `TortoisePlink.exe`. C'est une version du programme populaire `Plink` et elle est incluse avec TortoiseSVN, mais elle est compilée comme une application sans fenêtre, donc vous n'obtenez pas de boîte DOS surgissant chaque fois vous vous authentifiez.

Vous devez spécifier le chemin complet vers l'exécutable. Pour `TortoisePlink.exe` c'est le répertoire `bin` standard de TortoiseSVN. Utilisez le bouton **Parcourir** pour aider à le localiser. Notez que si le chemin contient des espaces, vous devez utiliser des guillemets, par exemple

```
"C:\Program Files\TortoiseSVN\bin\TortoisePlink.exe"
```

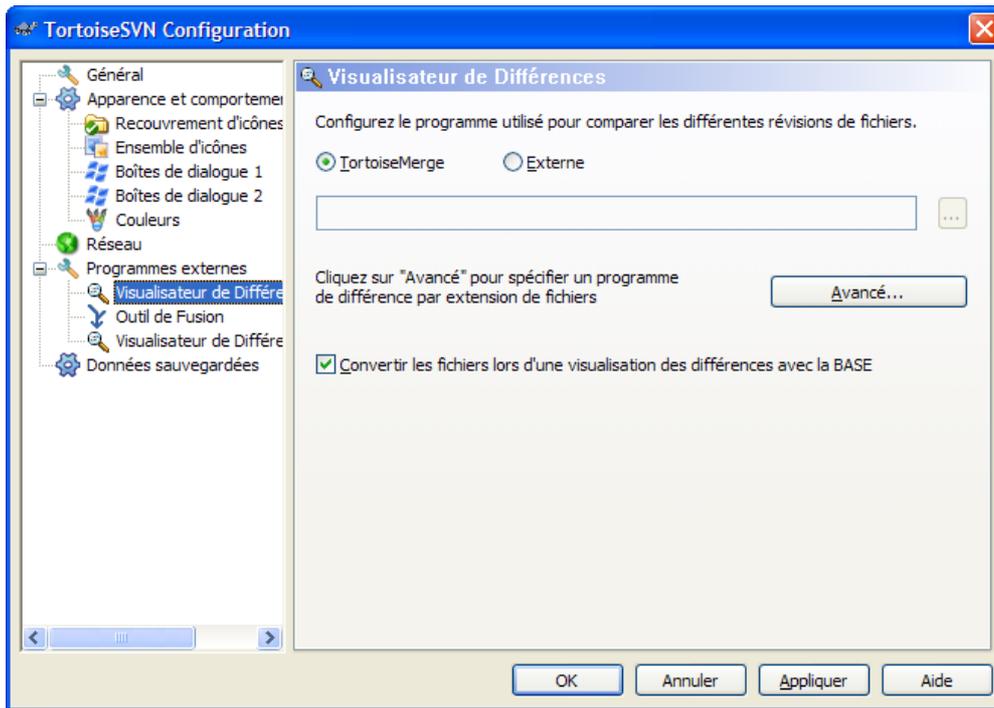
Un effet secondaire de ne pas avoir de fenêtre est qu'il n'y a nulle part où afficher les messages d'erreur, ainsi si l'authentification échoue, vous obtiendrez simplement un message disant quelque chose comme « Impossible d'écrire sur la sortie standard » Pour cette raison, nous vous recommandons de mettre d'abord en place en utilisant `Plink` standard. Quand tout fonctionne, vous pouvez utiliser `TortoisePlink` avec exactement les mêmes paramètres.

`TortoisePlink` n'a pas de documentation qui lui est propre parce que c'est juste une variante mineure de `Plink`. Renseignez-vous sur les paramètres de ligne de commande sur le [site web PuTTY](http://www.chiark.greenend.org.uk/~sgtatham/putty/) [http://www.chiark.greenend.org.uk/~sgtatham/putty/]

Pour que le mot de passe ne vous soit pas demandé à chaque fois, vous devriez utiliser un outil de mise en cache des mots de passe comme `Pageant`. Cet utilitaire est disponible sur le site internet de `PuTTY`.

Enfin, la mise en place de `SSH` sur le serveur et les clients est un processus non trivial qui est au-delà du champ d'application de ce fichier d'aide. Toutefois, vous pouvez trouver un guide dans la FAQ TortoiseSVN : [Subversion/TortoiseSVN SSH How-To](http://tortoisesvn.net/ssh_howto) [http://tortoisesvn.net/ssh\_howto].

### 4.30.5. Réglages des programmes externes



**Figure 4.59. La boîte de dialogue Configuration, page Visualisateur de différence**

Vous pouvez ici définir vos propres programmes de comparaison/fusion que TortoiseSVN devrait utiliser. Le réglage par défaut utilise TortoiseMerge qui est installé avec TortoiseSVN.

Lisez [Section 4.10.5, « Outils de différenciation/fusion externes »](#) pour une liste de quelques programmes externes de différenciation/fusion que les gens utilisent avec TortoiseSVN.

#### 4.30.5.1. Visualisateur de différences

Un programme de comparaison externe peut être utilisé pour comparer des révisions différentes de fichiers. Le programme externe devra obtenir les noms de fichier depuis la ligne de commande, avec les autres options de ligne de commande. TortoiseSVN utilise des paramètres de substitution préfixés par %. Quand il rencontre l'un d'eux, il substituera la valeur appropriée. L'ordre des paramètres dépendra du programme de comparaison que vous utilisez.

%base

Le fichier original sans vos changements

%bname

Le titre de la fenêtre pour le fichier de base

%mine

Votre propre fichier, avec vos changements

%yname

Le titre de la fenêtre pour votre fichier

Les titres de fenêtre ne sont pas des noms de fichier purs. TortoiseSVN traite cela comme un nom à afficher et crée les noms en conséquence. Si par exemple si vous faites une comparaison entre un fichier dans la révision 123 avec un fichier de votre copie de travail, les noms seront nom de fichier : révision 123 et nom de fichier : copie de travail

For example, with ExamDiff Pro:

```
C:\Path-To\ExamDiff.exe %base %mine --left_display_name:%bname
--right_display_name:%yname
```

or with KDiff3:

```
C:\Path-To\kdiff3.exe %base %mine --L1 %bname --L2 %yname
```

or with WinMerge:

```
C:\Path-To\WinMerge.exe -e -ub -dl %bname -dr %yname %base %mine
```

or with Araxis:

```
C:\Path-To\compare.exe /max /wait /title1:%bname /title2:%yname
%base %mine
```

Si vous utilisez la propriété `svn:keywords` pour développer des mots-clés et en particulier la révision d'un fichier, alors il peut y avoir des différences entre les fichiers qui est purement due à la valeur actuelle du mot-clé. De même si vous utilisez `svn:eol-style = native` le fichier BASE aura des fins de ligne LF tandis que votre fichier aura des fins de ligne CR-LF. TortoiseSVN cachera normalement ces différences automatiquement en analysant syntaxiquement d'abord le fichier BASE pour étendre les mots-clés et les fins de ligne avant de faire l'opération de comparaison. Cependant, cela peut prendre du temps avec de gros fichiers. Si **Convertir les fichiers lors d'une comparaison avec la BASE** est décoché alors TortoiseSVN sautera le prétraitement des fichiers.

Vous pouvez aussi indiquer un autre outil de diff pour les propriétés Subversion. Dans la mesure où ce sont de courtes chaînes de caractères, il est légitime de vouloir une visionneuse plus compacte.

Si vous avez configuré un outil de comparaison alternatif, vous pouvez accéder à TortoiseMerge *et* à l'outil tiers à partir des menus contextuels. Menu contextuel → Voir les différences utilise l'outil de comparaison primaire, et **Shift** + Menu contextuel → Voir les différences utilise l'outil de comparaison secondaire.

#### 4.30.5.2. Outil de fusion

Un programme de fusion externe utilisé pour résoudre les fichiers en conflit. La substitution de paramètre est utilisée de la même manière qu'avec le programme de comparaison.

`%base`  
le fichier original sans vos changements ou ceux des autres

`%bname`  
Le titre de la fenêtre pour le fichier de base

`%mine`  
votre propre fichier, avec vos changements

`%yname`  
Le titre de la fenêtre pour votre fichier

`%theirs`  
le fichier tel qu'il est dans le référentiel

`%tname`  
Le titre de la fenêtre pour le fichier dans le référentiel

`%merged`  
le fichier en conflit, le résultat de l'opération de fusion

%mname

Le titre de la fenêtre pour le fichier fusionné

Par exemple, avec Perforce Merge:

```
C:\Chemin-De\P4Merge.exe %base %theirs %mine %merged
```

ou avec KDiff3:

```
C:\Chemin-De\kdiff3.exe %base %mine %theirs -o %merged
--L1 %bname --L2 %yname --L3 %tname
```

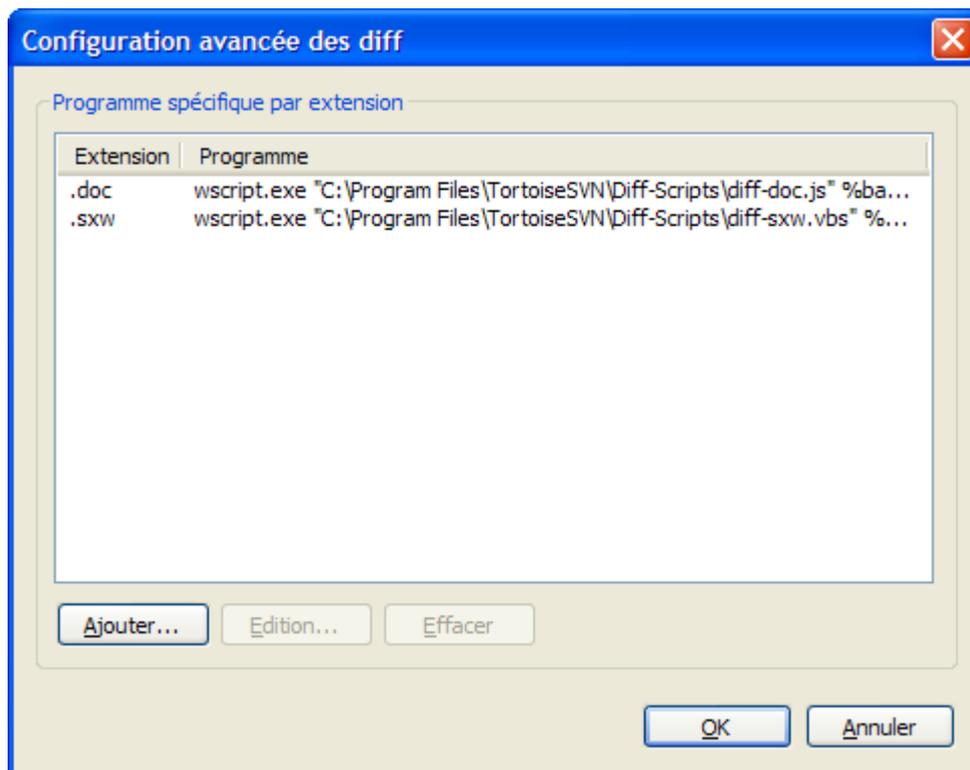
ou avec Araxis:

```
C:\Chemin-De\compare.exe /max /wait /3 /title1:%tname /title2:%bname
/title3:%yname %theirs %base %mine %merged /a2
```

ou avec WinMerge (2.8 or later):

```
C:\Chemin-De\WinMerge.exe %merged
```

#### 4.30.5.3. Réglages avancés de comparaison/fusion



**Figure 4.60. La boîte de dialogue Configuration, Boîte de dialogue Comparaison/fusion avancée**

Dans les réglages avancés, vous pouvez définir un programme de comparaison et de fusion différent pour chaque extension de fichier. Par exemple, vous pourriez associer Photoshop comme programme de « comparaison » pour les fichiers .jpg ;-) Vous pouvez aussi associer la propriété `svn:mime-type` à un programme de comparaison ou de fusion.

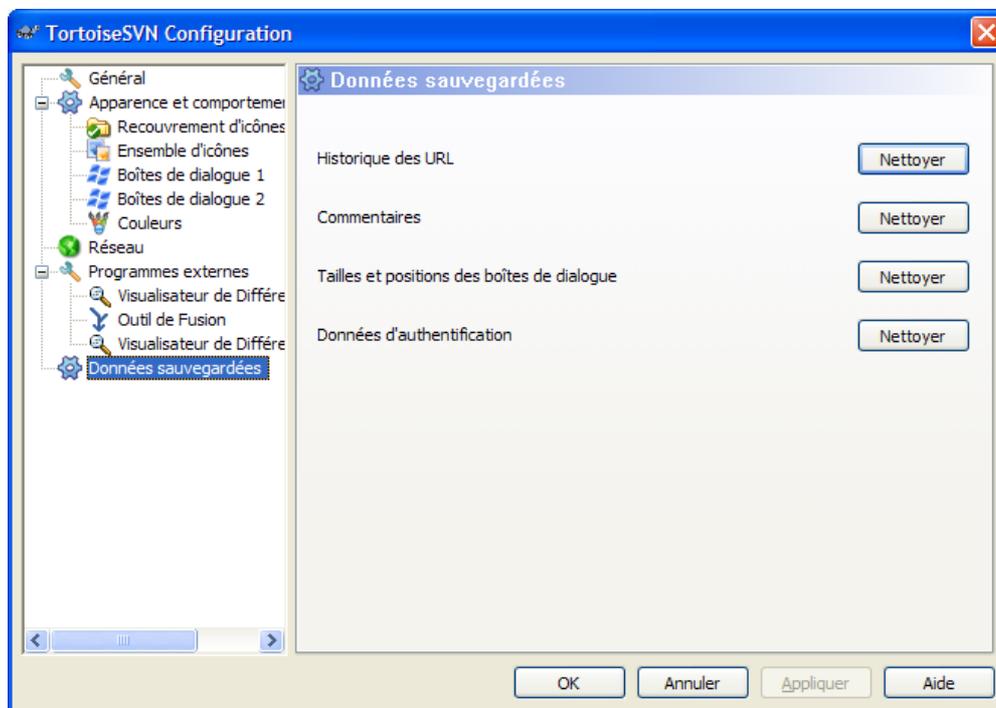
Pour associer en utilisant une extension de fichier, vous devez spécifier l'extension. Utilisez `.bmp` pour décrire les fichiers bitmap de Windows. Pour associer en utilisant la propriété `svn:mime-type`, spécifier le type mime, en incluant un slash, par exemple `text/xml`.

#### 4.30.5.4. Visualisateur de différences unifiées

Un programme de visualisation pour les fichiers de différences unifiées (fichiers patch). Aucun paramètre n'est exigé. L'option **Défaut** consiste à vérifier l'association des fichiers pour les fichiers `.diff`, et ensuite pour les fichiers `.txt`. Si vous n'avez pas de visualisateur pour les fichiers `.diff`, vous arriverez très probablement sur le Bloc-notes.

Le programme Bloc-notes original de Windows ne se comporte pas bien sur les fichiers qui n'ont pas de fins de ligne standards CR-LF. Puisque la plupart des fichiers de différences unifiées ont des fins de ligne LF pures, ils n'apparaissent pas bien dans le Bloc-notes. Cependant, vous pouvez télécharger un remplaçant du Bloc-notes gratuit *Notepad2* [<http://www.flos-freeware.ch/notepad2.html>] qui montre non seulement les fins de ligne correctement, mais aussi met en couleurs les lignes ajoutées et supprimées.

#### 4.30.6. Configuration des données sauvegardées



**Figure 4.61. La boîte de dialogue Configuration, Page Données sauvegardées**

Pour votre convenance, TortoiseSVN enregistre les réglages que vous utilisez et se souvient où vous avez été récemment. Si vous voulez nettoyer ce cache de données, vous pouvez le faire ici.

##### Historique des URL

Chaque fois que vous extrayez une copie de travail, fusionnez des changements ou utilisez l'explorateur de référentiel, TortoiseSVN tient un rapport des URLs récemment utilisées et les propose dans une boîte déroulante. Cette liste est parfois encombrée par des URLs périmées donc il est utile de la nettoyer périodiquement.

Si vous souhaitez supprimer un seul élément de l'une des zones de liste déroulante vous pouvez le faire sur place. Il suffit de cliquer sur la flèche pour afficher le contenu de la liste déroulante, déplacer la souris sur l'élément que vous souhaitez supprimer et appuyer sur **Maj+Suppr**.

#### Messages de log (fenêtre d'édition)

TortoiseSVN stocke les commentaires récents de livraison que vous saisissez. Ceux-ci sont stockés par référentiel, donc si vous avez accès à beaucoup de référentiels, cette liste peut devenir assez longue.

#### Messages de log (Montrer la fenêtre de log)

TortoiseSVN met en cache les messages de log, récupérés par la boîte de dialogue Montrer les logs, pour faire gagner du temps pour la prochaine fois où vous afficherez le log. Si quelqu'un d'autre édite un message de log et que ce message est déjà mis en cache, vous ne verrez pas les changements jusqu'à ce que vous vidiez le cache. La mise en cache du log est activable via l'onglet Log Cache.

#### Tailles et positions des boîtes de dialogue

Plusieurs boîtes de dialogue se souviennent de la taille et de la position de l'écran utilisées en dernier.

#### Données d'authentification

Quand vous vous authentifiez avec un serveur Subversion, le nom de l'utilisateur et le mot de passe sont mis en cache localement pour que vous n'ayez pas à les entrer à nouveau. Vous pouvez vouloir effacer cela pour des raisons de sécurité ou parce que voulez accéder au référentiel sous un autre nom d'utilisateur ... est-ce que John sait que vous utilisez son PC ?

Si vous voulez effacer des données d'authentification pour un serveur particulier seulement, lisez [Section 4.1.5, « Authentification »](#) pour des instructions sur la façon de trouver les données mises en cache.

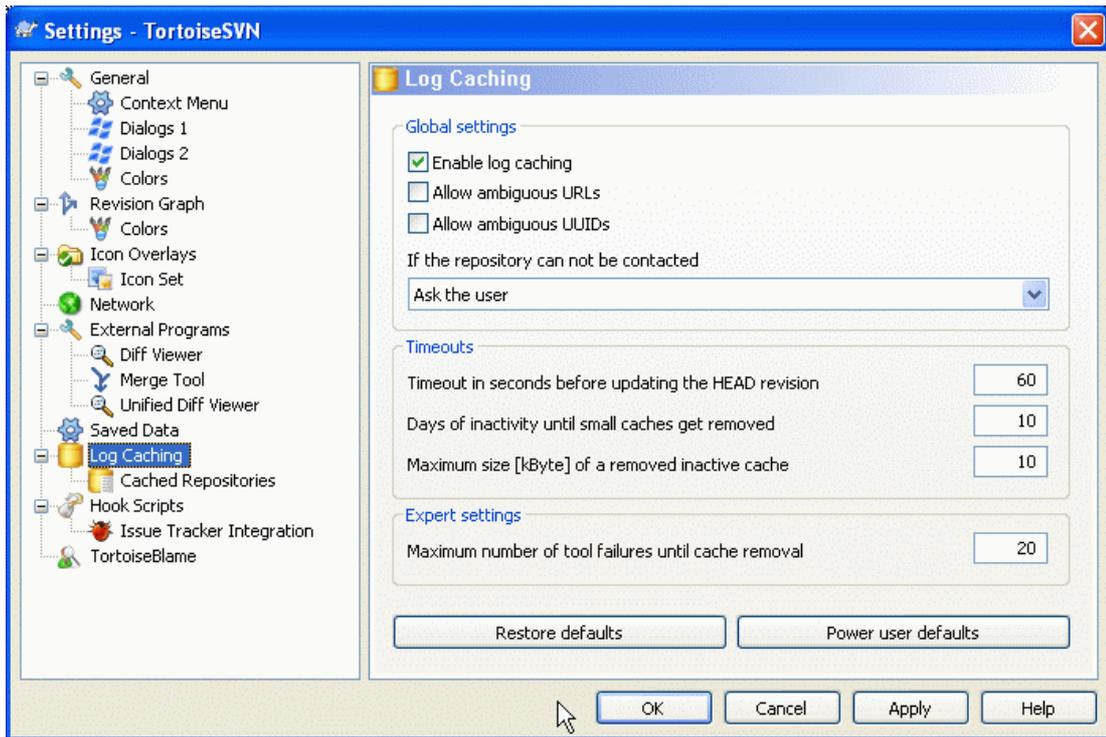
#### Log des Actions

TortoiseSVN tient un journal de tout ce qui est écrit dans ses dialogues d'avancement. Cela peut être utile lorsque, par exemple, vous souhaitez vérifier ce qui s'est passé dans une commande de mise à jour récente.

Le fichier journal est limité en longueur et quand il devient trop gros le contenu le plus ancien est supprimé. Par défaut 4000 lignes sont conservées, mais vous pouvez personnaliser ce nombre.

Depuis cet endroit vous pouvez voir le contenu du fichier de commentaires, ainsi que le vider.

### **4.30.7. Mise en Cache des messages de log**



**Figure 4.62. La boîte de dialogue de Configuration, Page de Mise en Cache des Logs**

Cette boîte de dialogue vous permet de configurer la fonctionnalité de mise en cache du journal TortoiseSVN, qui conserve une copie locale des messages de log et des chemins modifiés afin d'éviter des téléchargements consommateurs de ressources à partir du serveur. Utiliser le cache journal peut accélérer considérablement l'ouverture de la boîte de dialogue journal et le graphique de révision. Un autre avantage est que les messages du journal sont toujours accessibles en mode hors connexion.

#### Activer la mise en cache des messages de log

Permet la mise en cache chaque fois que des données log sont demandés. Si coché, les données seront récupérées à partir du cache lorsqu'elles seront disponibles, et les messages qui ne sont pas dans le cache seront récupérés sur le serveur et ajoutés au cache.

Si le cache est désactivé, les données seront toujours récupérées directement à partir du serveur et non stockées localement.

#### Permettre les URLs ambiguës

Parfois vous pouvez avoir à vous connecter à un serveur qui utilise la même URL pour tous les référentiels. Les anciennes versions de `svnbridge` devaient le faire. Si vous avez besoin d'accéder à ces référentiels, vous devez cocher cette option. Sinon, ne la cochez pas pour améliorer les performances.

#### Permettre les UUIDs ambigus

Certains services d'hébergement donnent à tous leurs dépôts le même UUID. Vous pouvez même le faire vous-même en copiant un dossier de référentiel pour en créer un nouveau. Pour toutes sortes de raisons, cela est une mauvaise idée - un UUID doit être *unique*. Toutefois, le cache de log continue de fonctionner dans un tel cas si vous cochez cette case. Si vous n'en avez pas besoin, laissez décoché pour améliorer les performances.

#### Si le référentiel n'est pas disponible

Si vous travaillez en mode hors connexion, ou si le serveur de référentiel est hors service, le cache du journal peut toujours être utilisé. Bien sûr, le cache peut ne pas être à jour, il existe alors des options pour vous permettre de choisir si cette fonction doit être utilisée.

Lorsque les données du journal sont issues du cache sans avoir contacté le serveur, la boîte de dialogue utilisant ces messages indiquera l'état hors connexion dans sa barre de titre.

Temps d'attente maximum écoulé avant la mise à jour de la révision de tête (HEAD).

Lorsque vous ouvrez la boîte de dialogue du journal, vous voudrez normalement contacter le serveur pour vérifier tous les nouveaux messages. Si le délai fixé ici est non-nul, le serveur ne sera contacté que lorsque le délai d'attente se sera écoulé depuis le dernier contact. Cela peut réduire les allers-retours avec le serveur si vous ouvrez fréquemment la boîte de dialogue et si le serveur est lent, mais les données présentées peuvent ne pas être complètement à jour. Si vous souhaitez utiliser cette fonctionnalité, nous vous suggérons d'utiliser une valeur de 300 (5 minutes) comme un compromis.

Nombre de jours d'inactivité avant que les petits caches soient supprimés

Si vous naviguez parmi un grand nombre de référentiels, vous accumulerez beaucoup de caches journal. Si vous ne les utilisez pas beaucoup, le cache ne grossira pas trop et TortoiseSVN les purgera après un délai défini par défaut. Utilisez cet élément pour contrôler la purge du cache.

Taille maximum des caches inactifs à supprimer

Les plus gros caches sont plus coûteux à rétablir, TortoiseSVN ne purge donc que les petits caches. Ajustez le seuil avec cette valeur.

Nombre maximum d'échecs de l'outil avant suppression du cache

De temps en temps quelque chose se passe mal avec la mise en cache et cause un crash. Dans ce cas le cache est normalement supprimée automatiquement pour éviter une réapparition du problème. Si vous utilisez le paramétrage de nuit le moins stable vous pouvez choisir de garder le cache de toute façon.

#### 4.30.7.1. Référentiels mis en mémoire cache

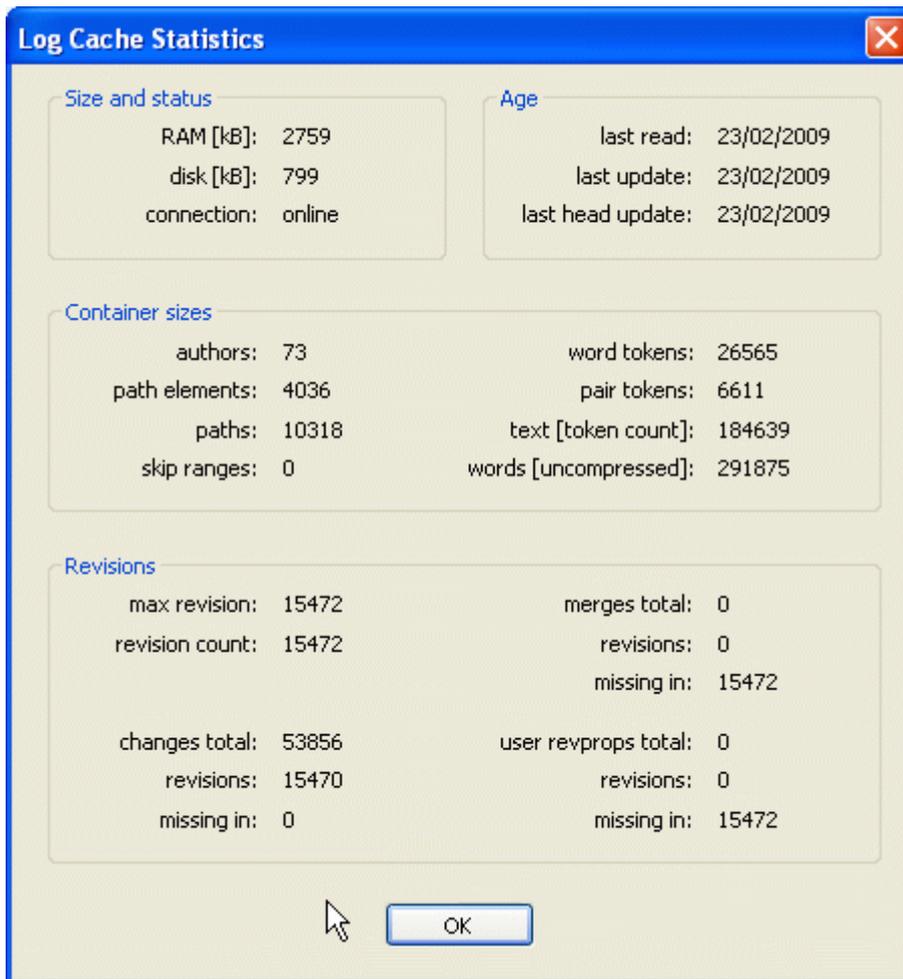
Sur cette page vous pouvez voir une liste des référentiels qui sont mis en cache localement, et l'espace utilisé pour le cache. Si vous sélectionnez l'un des référentiels, vous pourrez ensuite utiliser les boutons ci-dessous.

Cliquez sur le bouton **Mise à jour** pour rafraichir complètement la mémoire cache et remplir tous les trous. Pour les gros référentiels cette tâche peut être très coûteuse, mais utile si vous souhaitez vous déconnecter et avoir le meilleur cache possible.

Cliquez sur le bouton **Exportater** pour exporter l'intégralité du cache dans un ensemble de fichiers CSV. Cela peut être utile si vous voulez traiter les données du journal à l'aide d'un programme externe, surtout pour les développeurs.

Cliquez sur **Supprimer** pour supprimer toutes les données mises en cache pour les référentiels sélectionnés. Cela ne va pas désactiver le cache pour le référentiel, ainsi la prochaine fois que vous demanderez les données du journal, un nouveau cache sera créé.

#### 4.30.7.2. Statistiques d'Utilisation du Cache



**Figure 4.63. La Fenêtre de propriétés, Statistiques d'Utilisation de la Mémoire Cache**

Cliquer sur le bouton **Détails** pour voir les statistiques détaillées d'une mémoire cache particulière. Beaucoup des champs montrés là ont surtout un intérêt pour les développeur de TortoiseSVN, ils ne sont donc pas tous expliqués en détail.

##### RAM

La quantité de mémoire servant à ce cache

##### Disque

La quantité d'espace disque utilisé pour le cache. Les données sont compressées, l'utilisation du disque est alors généralement assez modeste.

##### Connexion

Montre si le référentiel était disponible la dernière fois que le cache a été utilisé.

##### Dernière Mise à Jour

La dernière fois que le contenu de la mémoire cache a changé.

##### Dernière mise à jour de la version de tête

La dernière fois que la révision HEAD a été demandée au serveur.

##### Auteurs

Le nombre d'auteurs différents ayant enregistré des messages dans le cache.

**Chemins**

Le nombre de chemins listés, comme vous pourriez le voir avec `svn log -v`.

**Ignorer des plages de révisions**

Les plages de révisions que nous n'avons pas récupérées, tout simplement parce qu'elles n'ont pas été demandées. Ceci est une mesure du nombre de trous dans le cache.

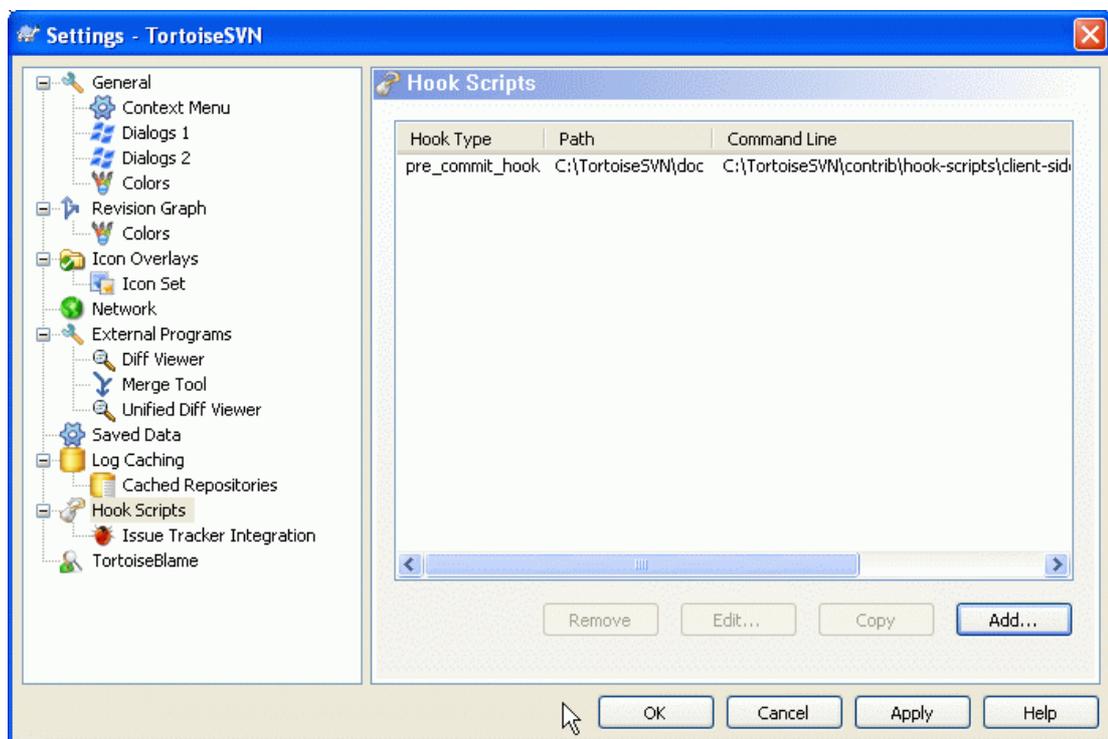
**Révision La plus Elevée**

Le numéro de version le plus élevé étant enregistré dans le cache.

**Compteur de Révision**

Le nombre de révisions stockées en mémoire cache. C'est un autre système de mesure de la mémoire cache.

**4.30.8. Scripts hook côté client**

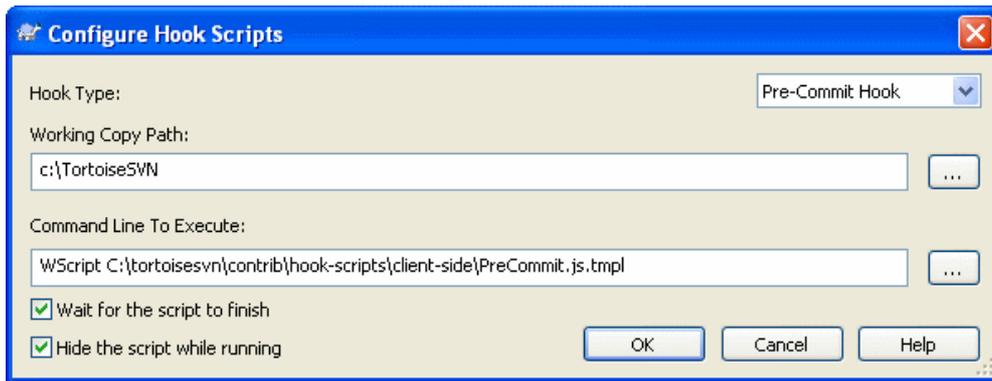


**Figure 4.64. La boîte de dialogue Configuration, page Scripts hook**

Cette boîte de dialogue vous permet de définir les scripts hook qui seront exécutés automatiquement lors de certaines actions de Subversion. Contrairement aux scripts de hook expliqué dans [Section 3.3](#), « Scripts de hook côté serveur », ces scripts sont exécutés localement sur le client.

Une application pour de tels (hooks) pourrait appeler un programme comme `SubWCRev.exe` pour mettre à jour les numéros de version après une révision, et peut-être pour déclencher une reconstruction.

Pour des raisons de sécurité et d'implémentation, les scripts de hook sont définis localement sur une machine plutôt que dans les propriétés du projet. Vous définissez ce qui se passe, en ne tenant pas compte de ce que quelqu'un d'autre pourrait livrer dans le référentiel. Bien sûr vous pouvez toujours choisir d'appeler un script qui est lui même sous contrôle de version.



**Figure 4.65. La fenêtre de paramétrage, configuration des scripts de hook**

Pour ajouter un nouveau script hook, cliquez simplement sur Ajouter et saisissez les détails.

Il existe actuellement six types de script hook disponibles

#### Start-commit

Appelé avant que la fenêtre de livraison ne s'affiche. Vous pouvez avoir besoin d'appeler ce hook pour ajouter à un fichier de version la liste des fichiers ayant été livrés et/ou le message de livraison. Néanmoins, il est important de noter qu'étant donné que le hook est appelé tôt, la liste complète des objets à livrer n'est pas disponible.

#### Pre-commit

Appelé lorsque l'utilisateur clique sur le bouton OK dans la fenêtre de livraison, et avant que la livraison ne commence. Ce hook contient une liste de tout ce qui sera livré.

#### Post-commit

Appelé après la fin de la livraison (qu'elle soit réussie ou non).

#### Start-update

Appelé avant que la fenêtre mise à jour-à-la -révision ne soit affichée.

#### Pre-update

Called before the actual Subversion update begins.

#### Post-update

Called after the update finishes (whether successful or not).

Un hook est défini pour un répertoire précis d'une copie de travail. Vous n'avez besoin que de spécifier le répertoire de plus haut niveau ; si vous faites une opération sur un sous répertoire, TortoiseSVN recherchera automatiquement dans les dossiers parents un répertoire correspondant.

Next you must specify the command line to execute, starting with the path to the hook script or executable. This could be a batch file, an executable file or any other file which has a valid windows file association, eg. a perl script.

La ligne de commande comporte plusieurs paramètres qui sont renseignés par TortoiseSVN. Les paramètres disponibles dépendent du script de hook appelé. Chaque script de hook a ses propres paramètres qui sont passés dans l'ordre suivant:

#### Start-commit

PATHMESSAGEFILECWD

#### Pre-commit

PATHDEPTHMESSAGEFILECWD

#### Post-commit

PATHDEPTHMESSAGEFILEREVISIONERRORCWD

**Start-update**

PATHCWD

**Pre-update**

PATHDEPTHREVISIONCWD

**Post-update**

PATHDEPTHREVISIONERRORCWD

La signification de chacun des paramètres est décrite ici :

**PATH**

Un chemin d'un fichier temporaire contenant tous les chemins d'où les opérations ont commencé. Il y a un chemin par ligne dans le fichier temporaire.

**DEPTH**

Profondeur dans laquelle la livraison/mise à jour est faite.

Les valeurs possibles sont :

- 2  
    svn\_depth\_unknown
- 1  
    svn\_depth\_exclude
- 0  
    svn\_depth\_empty
- 1  
    svn\_depth\_files
- 2  
    svn\_depth\_immediates
- 3  
    svn\_depth\_infinity

**MESSAGEFILE**

Le chemin d'un fichier contenant les commentaires de livraison. Le fichier est encodé en UTF-8. Après l'exécution réussie d'un script de hook de start-commit, le commentaire est relu, permettant au script de hook de le modifier.

**REVISION**

La révision du référentiel pour laquelle la mise à jour devrait être faite ou après qu'une livraison est terminée.

**ERROR**

Chemin vers un fichier contenant le message d'erreur. S'il n'y a pas d'erreur, le fichier sera vide.

**CWD**

Le répertoire de travail courant dans lequel le script est exécuté. Il est fixé dans le répertoire racine commun à tous les chemins affectés.

Notez que, bien que nous ayons donné des noms à ces paramètres pour plus de commodité, vous n'avez pas à vous référer à ces noms dans la configuration de hook. Tous les paramètres définis pour un hook particulier sont toujours transmis, que vous le souhaitiez ou non ;-)

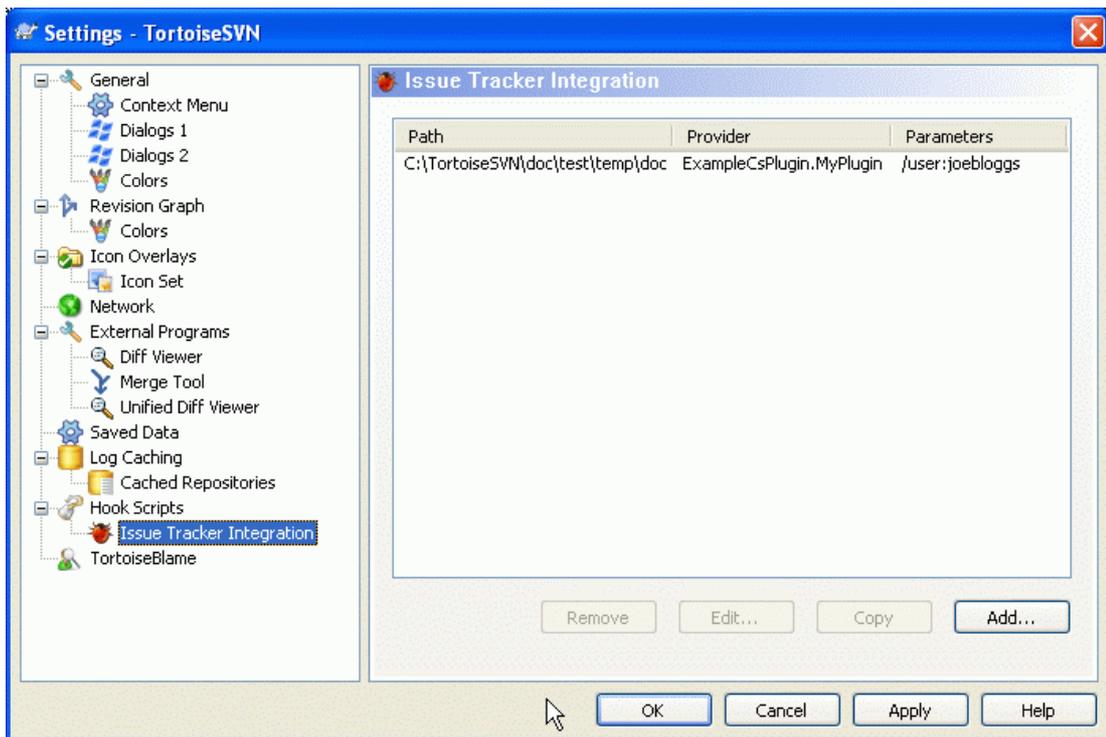
Si vous voulez que l'opération Subversion attende que le hook soit terminé, vérifiez **Attendez que le script se termine**.

Normalement vous souhaitez masquer les boîtes DOS lorsque le script s'exécute, **Masquer le script lors de l'exécution** est donc coché par défaut.

Des exemples de script hook client sont donnés dans le dossier `contrib` dans le [Référentiel TortoiseSVN](#) [<http://tortoisesvn.googlecode.com/svn/trunk/contrib/hook-scripts>]. (Section 3, « **TortoiseSVN est gratuit !** » explique comment accéder au référentiel).

#### 4.30.8.1. Intégration d'un gestionnaire d'incidents

TortoiseSVN peut utiliser un plugin COM pour interroger le gestionnaire d'incidents dans la boîte de dialogue de livraison. L'utilisation de tels plugins est décrite à [Section 4.28.2, « Récupérer des informations depuis le gestionnaire d'incidents »](#). Si votre administrateur système vous a fourni un plugin, que vous avez déjà installé et enregistré, c'est le lieu où préciser comment l'intégrer à votre copie de travail.



**Figure 4.66. La Fenêtre de Propriétés, Page d'Intégration d'un Gestionnaire d'Incidents**

Cliquez sur **Ajouter...** pour utiliser le plugin avec une copie de travail particulière. Ici vous pouvez spécifier le chemin de la copie de travail, choisir le plugin à utiliser à partir d'une liste déroulante de tous les plugins enregistrés de gestion d'incidents, et tous les paramètres à transmettre. Les paramètres sont spécifiques au plugin, mais peuvent inclure votre nom d'utilisateur du gestionnaire d'incidents, afin que le plugin puisse récupérer les incidents qui vous sont assignés.

If you want all users to use the same COM plugin for your project, you can specify the plugin also with the properties `bugtraq:provideruuid` and `bugtraq:providerparams`.

##### `bugtraq:provideruuid`

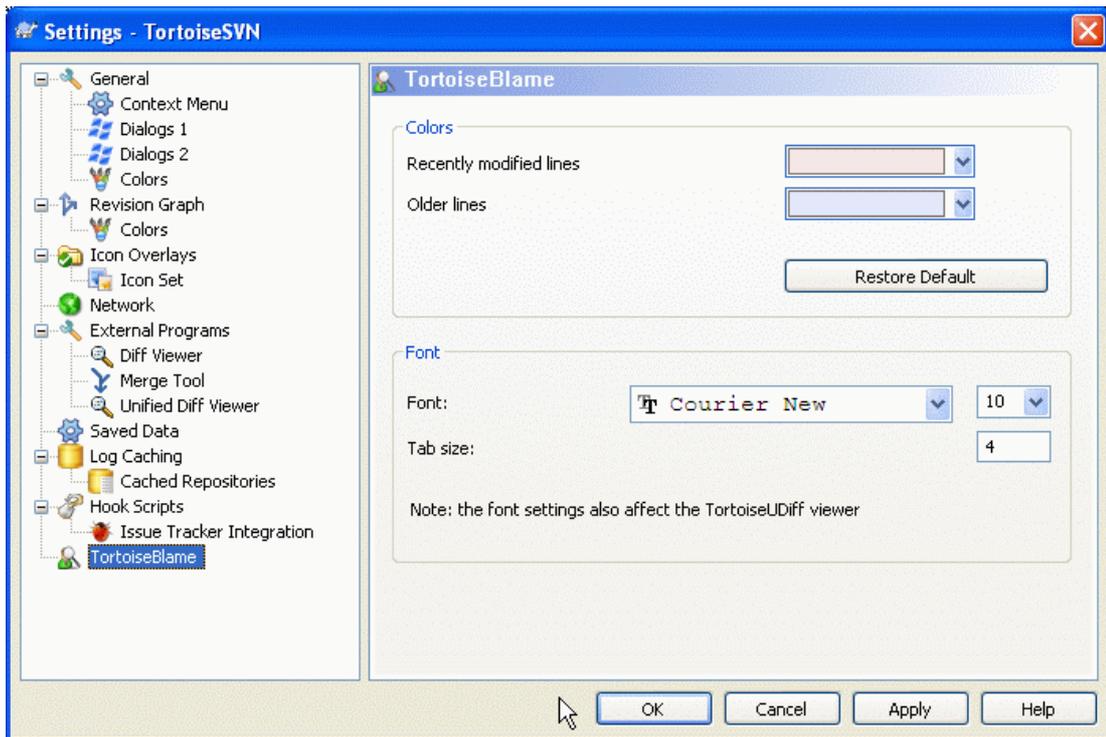
Cette propriété spécifie le COM UUID du `IBugtraqProvider`, par exemple `{91974081-2DC7-4FB1-B3BE-0DE1C8D6CE4E}`. (Cet exemple est l'UUID de [Gurtle bugtraq provider](#) [<http://code.google.com/p/gurtle/>], qui est un fournisseur pour le gestionnaire d'incidents [Google Code](#) [<http://code.google.com/hosting/>]).

##### `bugtraq:providerparams`

Cette propriété spécifie les paramètres envoyés à `IBugTraqProvider`.

Please check the documentation of your `IBugtraqProvider` plugin to find out what to specify in these two properties.

### 4.30.9. Configuration de TortoiseBlame



**Figure 4.67.** La boîte de dialogue de configuration, page de bannissement.

Les paramètres utilisés par TortoiseBlame sont contrôlés à partir du menu contextuel principal, et non directement par TortoiseBlame.

#### Couleurs

TortoiseBlame utilise la couleur de fond pour indiquer l'âge des lignes. Vous spécifiez les couleurs extrêmes en choisissant une couleur pour la révision la plus récente et une autre pour la révision la plus ancienne. TortoiseBlame applique une interpolation linéaire entre ces deux couleurs afin de colorer chaque ligne en fonction de son numéro de révision.

#### Police

Vous pouvez choisir la police utilisée pour afficher le texte et la taille à utiliser. Cela vaut tant pour le contenu du fichier, que pour l'auteur et les informations de révision figurant dans le volet de gauche.

#### Tabulations

Définit combien d'espaces utiliser à la place d'une tabulation dans le fichier.

### 4.30.10. Réglages dans le registre

Quelques-uns des réglages rarement utilisés ne sont disponibles qu'en éditant directement le registre. Il n'est nul besoin de préciser que vous ne devriez éditer le registre que si vous savez ce que vous faites.

#### Configuration

You can specify a different location for the Subversion configuration file using registry location `HKCU\Software\TortoiseSVN\ConfigDir`. This will affect all TortoiseSVN operations.

#### Mettre l'icône de la barre des tâches en cache

To add a cache tray icon for the TSVNCache program, create a `DWORD` key with a value of 1 at `HKCU\Software\TortoiseSVN\CacheTrayIcon`. This is really only useful for developers as it allows you to terminate the program gracefully.

#### Debug

To show the command line parameters passed from the shell extension to TortoiseProc.exe create a DWORD key with a value of 1 at HKCU\Software\TortoiseSVN\Debug.

#### Icones du Menu contextuel

This can be useful if you use something other than the windows explorer or if you get problems with the context menu displaying correctly. create a DWORD key with a value of 0 at HKCU\Software\TortoiseSVN>ShowContextMenuIcons if you don't want TortoiseSVN to not show icons for the shell context menu items. Set this value to 1 to show the icons again.

#### Bloquer l'état du recouvrement d'icônes

If you don't want the explorer to update the status overlays while another TortoiseSVN command is running (e.g. Update, Commit, ...) then create a DWORD key with a value of 1 at HKCU\Software\TortoiseSVN\BlockStatus.

#### Update Check URL

HKCU\Software\TortoiseSVN\UpdateCheckURL contains the URL from which TortoiseSVN tries to download a text file to find out if there are updates available. You can also set this under HKLM instead of HKCU if you want, but HKCU overwrites the setting in HKLM. This might be useful for company admins who don't want their users to update TortoiseSVN until they approve it.

#### Nom des fichiers sans extension dans la liste de complétion automatique

The auto-completion list shown in the commit message editor displays the names of files listed for commit. To also include these names with extensions removed, create a DWORD key with a value of 1 at HKCU\Software\TortoiseSVN\AutocompleteRemovesExtensions.

#### Colonnes de l'explorateur partout

The extra columns the TortoiseSVN adds to the details view in Windows Explorer are normally only active in a working copy. If you want those to be accessible everywhere, not just in working copies, create a DWORD key with a value of 1 at HKCU\Software\TortoiseSVN\ColumnsEveryWhere.

#### Séparateur des commentaires de fusion

When you merge revisions from another branch, and merge tracking information is available, the log messages from the revisions you merge will be collected to make up a commit log message. A pre-defined string is used to separate the individual log messages of the merged revisions. If you prefer, you can create a SZ key at HKCU\Software\TortoiseSVN\MergeLogSeparator containing a separator string of your choice.

#### Toujours bannir les modifications faites avec TortoiseMerge

TortoiseSVN allows you to assign external diff viewer. Most such viewers, however, are not suited for change blaming ([Section 4.23.2, « Annoter les différences »](#)), so you might wish to fall back to TortoiseMerge in this case. To do so, create a DWORD key with a value of 1 at HKCU\Software\TortoiseSVN\DiffBlamesWithTortoiseMerge.

#### Mise en valeur de la version courante des répertoires dans la fenêtre de log.

The log dialog highlights the current working copy revision when the log is shown for a file. To do the same thing for a folder requires a working copy crawl, which is the default action, but it can be a slow operation for large working copies. If you want to change the operation of this feature you must create a DWORD registry key at HKCU\Software\TortoiseSVN\RecursiveLogRev. A value of 0 disables the feature (no highlighting for folders), a value of 1 (default) will fetch the status recursively (find the highest revision in the working copy tree), and a value of 2 will check the revision of the selected folder itself, but will not check any child items.

#### Faire échouer la récupération si un élément du même nom existe déjà

By default, if you checkout a working copy over an existing unversioned folder structure, as you might do after import, then any existing which differ from the repository content will be left unchanged and marked as modified. When you come to commit, it is your local copy which will then be sent back to the repository. Some people would prefer the checkout to fail if the

existing content differs, so that if two people add the same file the second person's version does not overwrite the original version by mistake. If you want to force checkouts to fail in this instance you must create a DWORD registry key with value 0 at HKCU\Software\TortoiseSVN\AllowUnversionedObstruction.

#### 4.30.11. Dossiers de travail de Subversion

VS.NET quand il est utilisé avec des projets web ne peut pas manipuler les répertoires `.svn` que Subversion utilise pour stocker son information interne. Ce n'est pas un bug de Subversion. Le bug se trouve dans VS.NET et les extensions frontpage qu'il utilise.

Notez que le bug a été corrigé à partir de VS2005.

À partir de la version 1.3.0 de Subversion et de TortoiseSVN, vous pouvez définir la variable d'environnement `SVN_ASP_DOT_NET_HACK`. Si cette variable est définie, Subversion utilisera alors des dossiers `_svn` au lieu des dossiers `.svn`. Vous devez redémarrer votre shell pour que cette variable d'environnement entre en vigueur. Normalement cela revient à redémarrer votre PC. Pour que ce soit plus facile, vous pouvez maintenant faire cela depuis la page des réglages généraux en utilisant une simple case à cocher - référez-vous à [Section 4.30.1, « Configuration générale »](#).

Pour plus d'informations et d'autres façons d'éviter ce problème en premier lieu, regardez l'article à ce sujet dans notre [FAQ](http://tortoisesvn.net/aspdotnethack) [http://tortoisesvn.net/aspdotnethack].

#### 4.31. Étape Finale

##### Faites une donation!

Bien que TortoiseSVN et TortoiseMerge soient gratuits, vous pouvez supporter les développeurs en envoyant des patches et en jouant un rôle actif dans le développement. Vous pouvez aussi aider à nous réconforter pendant les heures interminables que nous passons devant nos ordinateurs.

Pendant que nous travaillons sur TortoiseSVN, nous aimons bien écouter de la musique. Et puisque nous passons beaucoup d'heures sur le projet, nous avons besoin de *beaucoup* de musique. Donc nous avons créé quelques listes de voeux avec nos CD et DVD préférés : <http://tortoisesvn.tigris.org/donate.html>. Regardez aussi la liste des personnes qui ont contribué au projet en envoyant des patches ou des traductions.

---

# Chapitre 5. Le programme SubWCRev

SubWCRev est un programme console Windows qui peut être utilisé pour lire le statut d'une copie de travail Subversion et exécuter facultativement la substitution de mots-clé dans un fichier modèle. C'est souvent utilisé comme une partie du processus de génération comme un moyen d'incorporer l'information de la copie de travail dans l'objet que vous générez. Typiquement cela pourrait être utilisé pour inclure le numéro de révision dans une boîte d'« À propos ».

## 5.1. La ligne de commande SubWCRev

SubWCRev lit le statut Subversion de tous les fichiers d'une copie de travail, en excluant par défaut les références externes. Il enregistre le plus haut numéro de révision livrée trouvé et la date de cette révision. Il enregistre aussi s'il y a des modifications locales dans la copie de travail, ou des révisions mélangées. Le numéro de révision, la plage de révisions mises à jour et le statut de modification sont affichés sur stdout.

SubWCRev.exe est appelé de la ligne de commande ou d'un script et est contrôlé en utilisant les paramètres de ligne de commande.

```
SubWCRev CheminCopieTravail [VersionSrcFichier VersionDstFichier] [-nmdfe]
```

CheminCopieTravail est le chemin de la copie de travail étant vérifiée. Vous pouvez seulement utiliser SubWCRev sur les copies de travail, et pas directement sur le référentiel. Le chemin peut être absolu ou relatif au répertoire de travail courant.

Si vous voulez que SubWCRev exécute la substitution de mots-clé, pour que les champs comme la révision du référentiel et l'URL soient sauvegardées dans un fichier texte, vous devez fournir un fichier modèle `FichierVersionSrc` et un fichier de sortie `FichierVersionDst` qui contient la version substituée du modèle.

Il y a un certain nombre de commutateurs facultatifs qui affectent la façon dont fonctionne SubWCRev. Si vous en utilisez plus d'un, ils doivent être spécifiés comme un groupe simple, par exemple `-nm`, et non `-n -m`.

Aller sur...	Description
-n	Si ce commutateur est renseigné, SubWCRev se fermera avec <code>ERRORLEVEL 7</code> si la copie de travail contient des modifications locales. Cela peut être utile pour empêcher une compilation incluant des changements non livrés.
-m	Si ce commutateur est renseigné, SubWCRev se fermera avec <code>ERRORLEVEL 8</code> si la copie de travail contient des révisions mélangées. Cela peut être utile pour empêcher la compilation d'une version partiellement mise à jour.
-d	Si ce commutateur est donné, SubWCRev sortira avec <code>ERRORLEVEL 9</code> si le fichier de destination existe déjà.
-f	Si ce commutateur est donné, SubWCRev inclura la dernière révision changée des dossiers. Le comportement par défaut consiste à utiliser seulement les fichiers lors de l'obtenant des numéros de révision.
-e	Si ce commutateur est donné, SubWCRev examinera les répertoires qui sont inclus avec <code>svn:externals</code> , mais seulement s'ils sont du même référentiel. Le comportement par défaut est d'ignorer les références externes.
-x	Si ce commutateur est donné, SubWCRev renverra les numéros de révision en hexadécimal.
-X	Si ce commutateur est donné, SubWCRev renverra les numéros de révision en hexadécimal, préfixés de '0X'.

**Tableau 5.1. Liste des commutateurs de ligne de commande disponibles**

## 5.2. Substitution de mot-clés

Si un fichier source et un fichier de destination sont fournis, SubWCRev copie la source à la destination, en exécutant la substitution de mots-clé comme suit :

Mot-clé	Description
\$WCREV\$	Remplacé par la plus haute révision livrée dans la copie de travail.
\$WCDATE\$	Remplacé par la date/heure de livraison de la plus haute révision. Par défaut, le format international est utilisé: yyyy-mm-dd hh:mm:ss. Alternativement, vous pouvez spécifier un format personnalisé qui sera utilisé avec <code>strftime()</code> , par exemple: \$WCDATE=%a %b %d %I:%M:%S %p\$. Pour une liste des caractères de formatage disponibles, consultez <a href="http://www.cppreference.com/stddate/strftime.html">Aide en ligne</a> [http://www.cppreference.com/stddate/strftime.html].
\$WCNOW\$	Remplacé par la date/heure actuelle du système. Cela peut être utile pour indiquer l'heure de compilation. Le format de l'heure est décrit comme ceci \$WCDATE\$.
\$WCRANGES\$	Remplacé par la plage de révisions mises à jour dans la copie de travail. Si la copie de travail est dans un état cohérent, ce sera une seule révision. Si la copie de travail contient des révisions mélangées, soit parce qu'elle est périmée, soit à cause d'une mise à jour à la révision délibérée, donc la plage sera affichée dans la forme 100:200
\$WCMIXED\$	\$WCMIXED?VTexte:FTexte\$ est remplacé par TText s'il y a des révisions avec des mises à jour mélangées, ou FText sinon.
\$WCMODS\$	\$WCMODS?VTexte:FTexte\$ est remplacé par TText s'il y a des modifications locales, ou FText sinon.
\$WCURL\$	Remplacé par l'URL du référentiel du chemin de la copie de travail passée à SubWCRev.
\$WCINSVN\$	\$WCINSVN?TText:FText\$ est remplacé par TText si l'élément est versionné, ou FText sinon.
\$WCNEEDSLOCK\$	\$WCNEEDSLOCK?TText:FText\$ est remplacé par TText si l'élément à la propriété <code>svn:needs-lock</code> activée, ou FText sinon.
\$WCISLOCKED\$	\$WCISLOCKED?TText:FText\$ est remplacé par TText si l'élément est verrouillé, ou FText sinon.
\$WCLOCKDATE\$	Remplacé par la date de verrouillage. La mise en forme de l'heure peut être utilisée comme décrit pour \$WCDATE\$.
\$WCLOCKOWNER\$	Remplacer par le nom du propriétaire du verrou
\$WCLOCKCOMMENT\$	Remplacé par le commentaire du verrou

**Tableau 5.2. Liste des commutateurs de ligne de commande disponibles**



### Astuce

Some of these keywords apply to single files rather than to an entire working copy, so it only makes sense to use these when SubWCRev is called to scan a single file. This applies to \$WCINSVN\$, \$WCNEEDSLOCK\$, \$WCISLOCKED\$, \$WCLOCKDATE\$, \$WCLOCKOWNER\$ and \$WCLOCKCOMMENT\$.

## 5.3. Exemple de mot-clé

L'exemple ci-dessous montre comment les mots-clés dans un fichier modèle sont substitués dans le fichier de résultat.

```
// Fichier de test pour SubWCRev: testfile.tpl

char *Revision = "$WCREV$";
char *Modified = "$WCMODS?Modified:Not modified$";
char *Date     = "$WCDATE$";
char *Range    = "$WCRANGE$";
char *Mixed    = "$WCMIXED?Mixed revision WC:Not mixed$";
char *URL      = "$WCURL$";

#if $WCMODS?1:0$
#error Source is modified
#endif

// Fin du fichier

Après l'exécution de SubWCRev.exe path\to\workingcopy testfile.tpl
testfile.txt, le fichier de sortie testfile.txt doit être de la forme :

// Fichier de test pour SubWCRev: testfile.txt

char *Revision = "3701";
char *Modified = "Modified";
char *Date     = "2005/06/15 11:15:12";
char *Range    = "3699:3701";
char *Mixed    = "Mixed revision WC";
char *URL      = "http://nom.de.domaine.du.projet.org/svn/trunk/src";

#if 1
#error Source is modified
#endif

// End of file
```



### Astuce

Un fichier comme ceci sera inclus dans la compilation, ainsi il faut s'attendre à ce qu'il versionné. Assurez-vous de versionner le fichier de modèle, et pas le fichier généré, sinon chaque fois que vous régénérer le fichier de version, vous aurez besoin de livrer la modification, ce qui au final signifie que le fichier de version doit être mise à jour.

## 5.4. Interface COM

Si vous avez besoin d'accéder à l'information sur les révisions depuis d'autres programmes que Subversion, vous pouvez utiliser l'objet COM SubWCRev comme interface. L'objet à créer est SubWCRev.object, et les méthodes suivantes sont supportées :

Méthode	Description
.GetWCInfo	Cette méthode parcourt la copie de travail en regroupant les informations de révision. Naturellement, vous devez l'appeler avant d'utiliser les autres méthodes pour accéder à l'information. Le premier paramètre est le chemin. Le deuxième paramètre doit être vrai si vous souhaitez inclure les révisions de dossier. Équivalent à la commande -f. Le troisième paramètre doit être vrai si vous voulez inclure svn:externals. Équivalent à la commande -e.

Méthode	Description
.Revision	Le numéro de version le plus élevé de la copie de travail. Equivalent à \$WCREV\$
.Date	Date/heure de la livraison de la dernière révision. Equivalent à \$WCDATE\$
.Author	L'auteur de la version de livraison la plus élevée, c'est à dire, la dernière personne à avoir fait une livraison.
.MinRev	Le numéro de version le moins élevé, comme montré dans \$WCRANGE\$.
.MaxRev	Le numéro de version le plus élevé, comme montré dans \$WCRANGE\$.
.HasModifications	Vrais s'il y a des modifications locales.
.Url	Remplacé par l'URL du référentiel du chemin de la copie de travail passée à GetWCInfo. Equivalent à \$WCURL\$.
.IsSvnItem	Vrai si l'élément est versionné
.NeedsLocking	Vrai si la propriété svn:needs-lock est activée pour l'élément.
.IsLocked	Vrai si l'élément est verrouillé
.LockCreationDate	Chaîne de caractère représentant la date à laquelle le verrou a été créé, ou une chaîne vide si l'élément n'est pas verrouillé.
.LockOwner	Chaîne de caractère représentant le propriétaire du verrou ou une chaîne vide si l'élément n'est pas verrouillé.
.LockComment	Le message renseigné au moment du verrouillage.

### Tableau 5.3. Les méthodes COM/automation sont supportées

Les exemples suivants montrent comment l'interface devrait être utilisée.

```
// testCOM.js - javascript file
// test script for the SubWCRev COM/Automation-object

filesystem = new ActiveXObject("Scripting.FileSystemObject");

revObject1 = new ActiveXObject("SubWCRev.object");
revObject2 = new ActiveXObject("SubWCRev.object");
revObject3 = new ActiveXObject("SubWCRev.object");
revObject4 = new ActiveXObject("SubWCRev.object");

revObject1.GetWCInfo(
    filesystem.GetAbsolutePathName("."), 1, 1);
revObject2.GetWCInfo(
    filesystem.GetAbsolutePathName(".."), 1, 1);
revObject3.GetWCInfo(
    filesystem.GetAbsolutePathName("SubWCRev.cpp"), 1, 1);
revObject4.GetWCInfo(
    filesystem.GetAbsolutePathName("../.."), 1, 1);

wcInfoString1 = "Revision = " + revObject1.Revision +
    "\nMin Revision = " + revObject1.MinRev +
    "\nMax Revision = " + revObject1.MaxRev +
    "\nDate = " + revObject1.Date +
    "\nURL = " + revObject1.Url + "\nAuthor = " +
    revObject1.Author + "\nHasMods = " +
```

```
revObject1.HasModifications + "\nIsSvnItem = " +
revObject1.IsSvnItem + "\nNeedsLocking = " +
revObject1.NeedsLocking + "\nIsLocked = " +
revObject1.IsLocked + "\nLockCreationDate = " +
revObject1.LockCreationDate + "\nLockOwner = " +
revObject1.LockOwner + "\nLockComment = " +
revObject1.LockComment;
wcInfoString2 = "Revision = " + revObject2.Revision +
"\nMin Revision = " + revObject2.MinRev +
"\nMax Revision = " + revObject2.MaxRev +
"\nDate = " + revObject2.Date +
"\nURL = " + revObject2.Url + "\nAuthor = " +
revObject2.Author + "\nHasMods = " +
revObject2.HasModifications + "\nIsSvnItem = " +
revObject2.IsSvnItem + "\nNeedsLocking = " +
revObject2.NeedsLocking + "\nIsLocked = " +
revObject2.IsLocked + "\nLockCreationDate = " +
revObject2.LockCreationDate + "\nLockOwner = " +
revObject2.LockOwner + "\nLockComment = " +
revObject2.LockComment;
wcInfoString3 = "Revision = " + revObject3.Revision +
"\nMin Revision = " + revObject3.MinRev +
"\nMax Revision = " + revObject3.MaxRev +
"\nDate = " + revObject3.Date +
"\nURL = " + revObject3.Url + "\nAuthor = " +
revObject3.Author + "\nHasMods = " +
revObject3.HasModifications + "\nIsSvnItem = " +
revObject3.IsSvnItem + "\nNeedsLocking = " +
revObject3.NeedsLocking + "\nIsLocked = " +
revObject3.IsLocked + "\nLockCreationDate = " +
revObject3.LockCreationDate + "\nLockOwner = " +
revObject3.LockOwner + "\nLockComment = " +
revObject3.LockComment;
wcInfoString4 = "Revision = " + revObject4.Revision +
"\nMin Revision = " + revObject4.MinRev +
"\nMax Revision = " + revObject4.MaxRev +
"\nDate = " + revObject4.Date +
"\nURL = " + revObject4.Url + "\nAuthor = " +
revObject4.Author + "\nHasMods = " +
revObject4.HasModifications + "\nIsSvnItem = " +
revObject4.IsSvnItem + "\nNeedsLocking = " +
revObject4.NeedsLocking + "\nIsLocked = " +
revObject4.IsLocked + "\nLockCreationDate = " +
revObject4.LockCreationDate + "\nLockOwner = " +
revObject4.LockOwner + "\nLockComment = " +
revObject4.LockComment;

WScript.Echo(wcInfoString1);
WScript.Echo(wcInfoString2);
WScript.Echo(wcInfoString3);
WScript.Echo(wcInfoString4);
```

---

# Chapitre 6. IBugtraqProvider interface

Pour obtenir une intégration plus fine avec les gestionnaires d'incident que par la simple utilisation des propriétés `bugtraq`, TortoiseSVN peut utiliser des plugins COM. Avec ces plugins, il est possible de récupérer des informations directement à partir du gestionnaire d'incidents, d'interagir avec l'utilisateur et de fournir des informations en retour à TortoiseSVN à propos des incidents en cours, de vérifier les messages log entrés par l'utilisateur et même d'exécuter des actions après une livraison réussie comme, par exemple, clore un incident.

Nous ne pouvons pas vous indiquer comment développer un objet COM dans votre langage préféré, mais des exemples de greffons en C++/ATL et C# sont disponibles dans notre référentiel dans le répertoire `contrib/issue-tracker-plugins`. Vous pouvez également y trouver les fichiers à inclure pour compiler votre greffon. (Section 3, « [TortoiseSVN est gratuit !](#) » explique comment accéder au référentiel).

## 6.1. L'interface de IBugtraqProvider

TortoiseSVN 1.5 can use plugins which implement the IBugtraqProvider interface. The interface provides a few methods which plugins can use to interact with the issue tracker.

```
HRESULT ValidateParameters (  
    // Fenêtre parent pour toutes les interfaces devant être  
    // affichées pendant la validation.  
    [in] HWND hParentWnd,  
  
    // La chaîne en paramètre doit être validée.  
    [in] BSTR parameters,  
  
    // La chaîne est elle valide ?  
    [out, retval] VARIANT_BOOL *valid  
);
```

This method is called from the settings dialog where the user can add and configure the plugin. The `parameters` string can be used by a plugin to get additional required information, e.g., the URL to the issue tracker, login information, etc. The plugin should verify the `parameters` string and show an error dialog if the string is not valid. The `hParentWnd` parameter should be used for any dialog the plugin shows as the parent window. The plugin must return `TRUE` if the validation of the `parameters` string is successful. If the plugin returns `FALSE`, the settings dialog won't allow the user to add the plugin to a working copy path.

```
HRESULT GetLinkText (  
    // Fenêtre parent pour toutes les interfaces (d'erreur) devant être affiché.  
    [in] HWND hParentWnd,  
  
    // La chaîne en paramètre, au cas où vous devriez communiquer avec votre  
    // service web (i.e.) pour savoir qu'est ce qu'un texte correct.  
    [in] BSTR parameters,  
  
    // Quel texte voulez vous afficher ?  
    // Utilisez le thread local courant.  
    [out, retval] BSTR *linkText  
);
```

The plugin can provide a string here which is used in the TortoiseSVN commit dialog for the button which invokes the plugin, e.g., "Choose issue" or "Select ticket". Make sure the string is not too long,

otherwise it might not fit into the button. If the method returns an error (e.g., E\_NOTIMPL), a default text is used for the button.

```

HRESULT GetCommitMessage (
    // Fenêtre principale de l'interface utilisateur de votre FAI.
    [in] HWND hParentWnd,

    // Paramètres pour votre FAI.
    [in] BSTR parameters,
    [in] BSTR commonRoot,
    [in] SAFEARRAY(BSTR) pathList,

    // The texte déjà présent dans le message de livraison.
    // Votre FAI doit inclure ce texte dans le nouveau message,
    // où c'est approprié.
    [in] BSTR originalMessage,

    // Le nouveau texte pour le message de livraison.
    // Ceci remplace le message original.
    [out, retval] BSTR *newMessage
);

```

This is the main method of the plugin. This method is called from the TortoiseSVN commit dialog when the user clicks on the plugin button. The `parameters` string is the string the user has to enter in the settings dialog when he configures the plugin. Usually a plugin would use this to find the URL of the issue tracker and/or login information or more. The `commonRoot` string contains the parent path of all items selected to bring up the commit dialog. Note that this is *not* the root path of all items which the user has selected in the commit dialog. The `pathList` parameter contains an array of paths (as strings) which the user has selected for the commit. The `originalMessage` parameter contains the text entered in the log message box in the commit dialog. If the user has not yet entered any text, this string will be empty. The `newMessage` return string is copied into the log message edit box in the commit dialog, replacing whatever is already there. If a plugin does not modify the `originalMessage` string, it must return the same string again here, otherwise any text the user has entered will be lost.

## 6.2. L'interface de IBugtraqProvider2

Dans TortoiseSVN 1.6 une nouvelle interface a été ajoutée, augmentant le potentiel des greffons. Cet interface IBugtraqProvider2 hérite de IBugtraqProvider.

```

HRESULT GetCommitMessage2 (
    // Fenêtre principale de l'interface utilisateur de votre FAI.
    [in] HWND hParentWnd,

    // Paramètres pour votre FAI.
    [in] BSTR parameters,
    // L'URL commune de livraison
    [in] BSTR commonURL,
    [in] BSTR commonRoot,
    [in] SAFEARRAY(BSTR) pathList,

    // Le texte déjà présent dans le message de livraison.
    // Votre FAI doit inclure ce texte dans le nouveau message,
    // où c'est approprié.
    [in] BSTR originalMessage,

    // Vous pouvez assigner des propriétés de révision personnalisées à la livraison
    // en configurant les deux prochains paramètres.

```

```

// note: les deux tableaux doivent avoir la même longueur.
//      Chaque propriété doit avoir une valeur!

// Le contenu du champ bugID (si affiché)
[in] BSTR bugID,

// Le contenu modifié du champ bugID
[out] BSTR * bugIDOut,

// La liste des noms de propriété de révision.
[out] SAFEARRAY(BSTR) * revPropNames,

// La liste des valeurs de propriété de révision.
[out] SAFEARRAY(BSTR) * revPropValues,

// Le nouveau texte du message de livraison.
// Ceci remplace le message original
[out, retval] BSTR * newMessage
);

```

This method is called from the TortoiseSVN commit dialog when the user clicks on the plugin button. This method is called instead of `GetCommitMessage()`. Please refer to the documentation for `GetCommitMessage` for the parameters that are also used there. The parameter `commonURL` is the parent URL of all items selected to bring up the commit dialog. This is basically the URL of the `commonRoot` path. The parameter `bugID` contains the content of the bug-ID field (if it is shown, configured with the property `bugtraq:message`). The return parameter `bugIDOut` is used to fill the bug-ID field when the method returns. The `revPropNames` and `revPropValues` return parameters can contain name/value pairs for revision properties that the commit should set. A plugin must make sure that both arrays have the same size on return! Each property name in `revPropNames` must also have a corresponding value in `revPropValues`. If no revision properties are to be set, the plugin must return empty arrays.

```

HRESULT CheckCommit (
    [in] HWND hParentWnd,
    [in] BSTR parameters,
    [in] BSTR commonURL,
    [in] BSTR commonRoot,
    [in] SAFEARRAY(BSTR) pathList,
    [in] BSTR commitMessage,
    [out, retval] BSTR * errorMessage
);

```

This method is called right before the commit dialog is closed and the commit begins. A plugin can use this method to validate the selected files/folders for the commit and/or the commit message entered by the user. The parameters are the same as for `GetCommitMessage2()`, with the difference that `commonURL` is now the common URL of all *checked* items, and `commonRoot` the root path of all checked items. The return parameter `errorMessage` must either contain an error message which TortoiseSVN shows to the user or be empty for the commit to start. If an error message is returned, TortoiseSVN shows the error string in a dialog and keeps the commit dialog open so the user can correct whatever is wrong. A plugin should therefore return an error string which informs the user *what* is wrong and how to correct it.

```

HRESULT OnCommitFinished (
    // Fenêtre principale pour n'importe quelle interface utilisateur (d'erreur) qu
    [in] HWND hParentWnd,

    // La racine commune de tous les chemins qui ont été livrés.

```

```

[in] BSTR commonRoot,

// Tous les chemins qui ont été livrés.
[in] SAFEARRAY(BSTR) pathList,

// Le texte déjà présent dans le message de livraison.
[in] BSTR logMessage,

// La révision de la livraison.
[in] ULONG revision,

// Une erreur à montrer à l'utilisateur si cette fonction
// renvoie autre chose que S_OK
[out, retval] BSTR * error
);

```

This method is called after a successful commit. A plugin can use this method to e.g., close the selected issue or add information about the commit to the issue. The parameters are the same as for `GetCommitMessage2`.

```

HRESULT HasOptions(
    // Si le fournisseur fournit des options
    [out, retval] VARIANT_BOOL *ret
);

```

This method is called from the settings dialog where the user can configure the plugins. If a plugin provides its own configuration dialog with `ShowOptionsDialog`, it must return `TRUE` here, otherwise it must return `FALSE`.

```

HRESULT ShowOptionsDialog(
    // Fenêtre principale pour la boîte de dialogue des options
    [in] HWND hParentWnd,

    // Paramètres pour votre FAI.
    [in] BSTR parameters,

    // La chaîne des paramètres
    [out, retval] BSTR * newparameters
);

```

This method is called from the settings dialog when the user clicks on the "Options" button that is shown if `HasOptions` returns `TRUE`. A plugin can show an options dialog to make it easier for the user to configure the plugin. The `parameters` string contains the plugin parameters string that is already set/entered. The `newparameters` return parameter must contain the parameters string which the plugin constructed from the info it gathered in its options dialog. That `parameters` string is passed to all other `IBugtraqProvider` and `IBugtraqProvider2` methods.

---

# Annexe A. Foire aux questions (FAQ)

Parce que TortoiseSVN est développé en permanence, il est parfois difficile de garder la documentation complètement à jour. Nous maintenons une *FAQ interactive en ligne* [<http://tortoisesvn.tigris.org/faq.html>] qui contient une sélection des questions les plus souvent posées sur les mailing lists de TortoiseSVN <dev@tortoisesvn.tigris.org> et <users@tortoisesvn.tigris.org>.

Nous maintenons aussi un projet *Gestionnaire d'incident* [<http://issues.tortoisesvn.net>] qui vous indique certaines des choses que nous avons sur notre liste À faire, et les bugs qui ont déjà été corrigés. Si vous pensez avoir trouvé un bug, ou si vous voulez demander une nouvelle fonctionnalité, vérifiez d'abord ici pour savoir si quelqu'un d'autre ne l'a pas fait avant vous.

Le meilleur endroit pour poser une question à laquelle vous n'avez pas trouvé de réponse est l'une des listes de diffusion. <users@tortoisesvn.tigris.org> est celle à utiliser pour les questions relatives à TortoiseSVN. Si vous voulez aider au développement de TortoiseSVN, vous devriez alors participer aux débats sur <dev@tortoisesvn.tigris.org>.

---

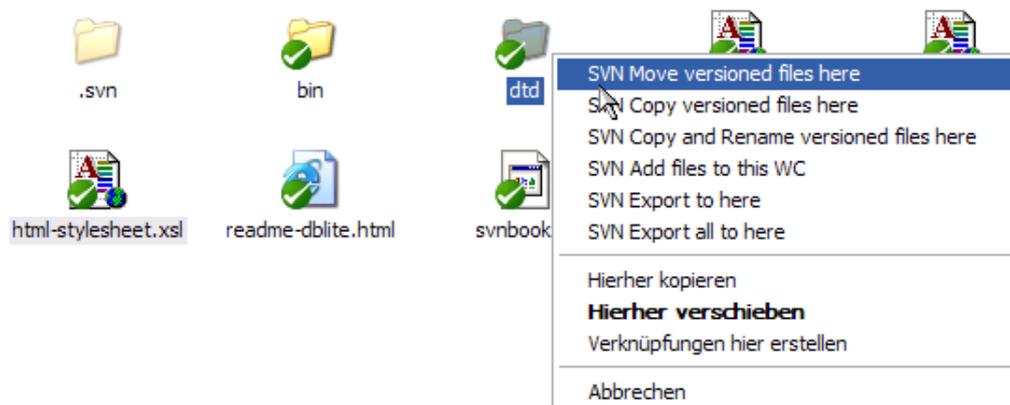
# Annexe B. Comment faire pour...

Cette annexe contient les solutions aux problèmes/questions que vous pourriez avoir en utilisant TortoiseSVN.

## B.1. Déplacer/copier beaucoup de fichiers en une fois

Déplacer/copier de simples fichiers peut être fait en utilisant TortoiseSVN → Renommer.... Mais si vous voulez déplacer/copier beaucoup de fichiers, cette façon est bien trop lente et demande trop de travail.

La façon recommandée est de glisser avec le bouton droit les fichiers au nouvel emplacement. Faites simplement un clic droit sur les fichiers que vous voulez déplacer/copier sans relâcher le bouton de souris. Glissez alors les fichiers au nouvel emplacement et relâchez le bouton de souris. Un menu contextuel apparaîtra où vous pouvez soit choisir Menu contextuel → SVN Copier ici, soit Menu contextuel → SVN Déplacer ici.



## B.2. Forcer les utilisateurs à entrer un commentaire

Il y a deux façons d'empêcher les utilisateurs de livrer avec un commentaire vide. L'une est spécifique à TortoiseSVN, l'autre fonctionne pour tous les clients Subversion, mais exige l'accès au serveur directement.

### B.2.1. Script hook sur le serveur

Si vous avez un accès direct au serveur du référentiel, vous pouvez installer un script hook pre-commit qui rejette toutes les livraisons avec des commentaires vides ou trop courts.

Dans le dossier du référentiel sur le serveur, il y a un sous-dossier hooks qui contient quelques exemples de scripts hook que vous pouvez utiliser. Le fichier `pre-commit.tmpl` contient un script type qui rejettera les livraisons si aucun commentaire n'est fourni, ou si le commentaire est trop court. Le fichier contient aussi des commentaires sur la façon d'installer/utiliser ce script. Suivez juste les instructions de ce fichier.

Cette méthode est celle recommandée si vos utilisateurs utilisent aussi d'autres clients Subversion que TortoiseSVN. L'inconvénient réside dans le fait que la livraison est rejetée par le serveur et donc les utilisateurs obtiendront un message d'erreur. Le client ne peut pas savoir avant la livraison qu'elle sera rejetée. Si vous voulez que TortoiseSVN ait le bouton OK désactivé jusqu'à ce que le commentaire soit assez long alors veuillez utiliser la méthode décrite ci-dessous.

## B.2.2. Propriétés de projet

TortoiseSVN utilise des propriétés pour contrôler certaines de ses fonctionnalités. Une de ces propriétés est la propriété `tsvn:minlogsize`.

Si vous définissez cette propriété sur un dossier, alors TortoiseSVN désactivera le bouton OK dans toutes les boîtes de dialogues de livraison jusqu'à ce que l'utilisateur ait entré un commentaire avec au moins la longueur indiquée dans la propriété.

Pour des informations détaillées sur ces propriétés de projet, veuillez vous référer à [Section 4.17, « Configuration des projets »](#)

## B.3. Mettre à jour les fichiers sélectionnés à partir du référentiel

Normalement, vous mettez à jour votre copie de travail en utilisant TortoiseSVN → Mettre à jour. Mais si vous voulez seulement récupérer les nouveaux fichiers qu'un collègue a ajoutés sans fusionner les changements d'autres fichiers dans même temps, vous avez besoin d'une approche différente.

Utilisez TortoiseSVN → Vérifier les modifications. Et cliquez sur Vérifier le référentiel pour voir ce qui a changé dans le référentiel. Sélectionnez les fichiers que vous voulez mettre à jour localement, utilisez ensuite le menu contextuel pour ne mettre à jour que ces fichiers.

## B.4. Annuler des révisions dans le référentiel

### B.4.1. Utiliser la boîte de dialogue du journal de révision

La façon la plus facile d'annuler les changements d'une seule révision, ou d'une plage de révisions, est d'utiliser la boîte de dialogue du journal de révision. C'est aussi la méthode à utiliser si vous voulez renoncer aux changements récents et faire d'une révision précédente la nouvelle révision HEAD.

1. Sélectionnez le fichier ou le dossier pour lequel vous voulez annuler les changements. Si vous voulez annuler tous les changements, cela devrait être le dossier au niveau supérieur.
2. Sélectionnez TortoiseSVN → Voir le journal pour afficher une liste des révisions. Vous pouvez avoir à utiliser Afficher tout ou 100 suivants pour afficher les révisions qui vous intéressent.
3. Sélectionnez la révision que vous voulez annuler. Si vous voulez annuler une plage de révisions, sélectionnez la première et maintenez la touche **Maj** en choisissant la dernière. Notez que pour des révisions multiples, la plage doit être continue sans écarts. Faites un clic droit sur les révisions choisies, puis sélectionnez Menu contextuel → Annuler les changements de ces révisions.
4. Ou si vous voulez faire d'une révision précédente la nouvelle révision HEAD, faites un clic droit sur les révisions sélectionnées, puis sélectionnez Menu contextuel → Revenir à cette révision. Cela annulera *tous* les changements après la révision choisie.

Vous avez annuler les changements dans votre copie de travail. Vérifiez les résultats, puis livrez les changements.

### B.4.2. Utiliser la boîte de dialogue fusionner

Pour annuler une plus grande plage de révisions, vous pouvez utiliser la boîte de dialogue Fusionner. La méthode précédente utilise la fusion en coulisses ; cette méthode l'utilise explicitement.

1. Dans votre copie de travail, sélectionnez TortoiseSVN → Fusionner.

2. Dans le champ **De** : entrez l'URL complète du dossier de la branche ou du tag contenant les changements que vous voulez annuler dans votre copie de travail. Cela devrait être l'URL par défaut.
3. Dans le champ **De la révision** entrez le numéro de la révision à laquelle vous êtes actuellement. Si vous êtes sûr qu'il n'y a personne d'autre faisant des changements, vous pouvez utiliser la révision HEAD.
4. assurez-vous que la case **Utiliser "Depuis :"** URL est cochée.
5. Dans le champ **Vers la révision**, saisissez le numéro de révision vers lequel vous souhaitez revenir, à savoir celui *précédant* la première révision à annuler.
6. Cliquez sur **OK** pour terminer la fusion.

Vous avez annuler les changements dans votre copie de travail. Vérifiez les résultats, puis livrez les changements.

### B.4.3. Utiliser `svndumpfilter`

Puisque TortoiseSVN ne perd jamais de données, vos révisions « annulées » existent toujours comme révisions intermédiaires dans le référentiel. Seule la révision HEAD a été changée à un état précédent. Si vous voulez faire que les révisions disparaissent complètement de votre référentiel, en effaçant toute trace de leur existence, vous devez utiliser des mesures plus extrêmes. À moins d'avoir une bonne raison pour le faire, ce n'est *pas recommandé*. Une raison possible serait que quelqu'un a livré un document confidentiel à un référentiel public.

La seule façon de supprimer des données du référentiel est d'utiliser l'outil de ligne de commande de Subversion `svnadmin`. Vous pouvez trouver une description de son fonctionnement dans le [Repository Maintenance](http://svnbook.red-bean.com/en/1.5/svn.reposadmin.maint.html) [http://svnbook.red-bean.com/en/1.5/svn.reposadmin.maint.html].

## B.5. Compare deux révisions d'un fichier ou d'un répertoire

Si vous voulez comparer deux révisions dans l'historique d'un fichier, par exemple les révisions 100 et 200 du même fichier, utilisez simplement TortoiseSVN → Voir le journal pour lister l'historique des révisions pour ce fichier. Choisissez les deux révisions que vous voulez comparer puis utilisez Menu contextuel → Comparer les révisions.

Si vous voulez comparer le même fichier dans deux arborescences différentes, par exemple dans la version trunk et dans une branche, vous pouvez utiliser l'explorateur de référentiel pour ouvrir les deux arborescences, sélectionnez le fichier dans les deux emplacements, puis utilisez Menu contextuel → Comparer les révisions.

Si vous voulez comparer deux arborescences pour voir ce qui a changé, par exemple le tronc et une version tagguée, vous pouvez utiliser TortoiseSVN → Graphique de révision. Sélectionnez les deux noeuds à comparer, utilisez ensuite Menu contextuel → Comparer les révisions HEAD. Cela montrera une liste des fichiers modifiés et vous pouvez alors choisir des fichiers individuellement pour voir les changements en détail. Vous pouvez également exporter une arborescence contenant tous les fichiers modifiés ou simplement une liste de tous les fichiers modifiés. Lisez [Section 4.10.3, « Comparer des répertoires »](#) pour plus d'information. Autrement utilisez Menu contextuel → Comparaison unifiée des révisions HEAD pour voir un résumé de toutes les différences, avec un contexte minimal.

## B.6. Inclure un sous-projet commun

Parfois vous voudrez inclure un autre projet dans votre copie de travail, peut-être du code de bibliothèque. Vous ne voulez pas faire un duplicata de ce code dans votre référentiel parce qu'alors vous perdriez la connexion avec le code original (et maintenu). Ou peut-être vous avez plusieurs projets qui partagent le code fondamental. Il y a au moins 3 façons de gérer cela.

### B.6.1. Utiliser svn:externals

Définit la propriété `svn:externals` sur un dossier de votre projet. Cette propriété consiste en une ou plusieurs lignes; chaque ligne comporte le nom d'un sous-dossier que vous voulez utiliser comme dossier d'extraction pour du code commun et l'URL du référentiel que vous voulez extraire là. Pour des détails plus complets, référez-vous à [Section 4.18, « Eléments externes »](#).

Livre le nouveau dossier. Maintenant quand vous mettez à jour, Subversion récupérera une copie de ce projet de son référentiel vers votre copie de travail. Les sous-dossiers seront créés automatiquement au besoin. Chaque fois que vous mettez à jour votre copie de travail principale, vous recevrez aussi la dernière version de tous les projets externes.

If the external project is in the same repository, any changes you make there there will be included in the commit list when you commit your main project.

Si le projet externe est dans un référentiel différent, les changements que vous faites au projet externe seront signalés quand vous livrez le projet principal, mais vous devez livrer ces changements externes séparément.

Parmi les trois méthodes décrites, c'est la seule qui n'a pas besoin d'installation côté client. Une fois que les projets externes sont spécifiés dans les propriétés du dossier, tous les clients recevront les dossiers remplis lors de la mise à jour.

### B.6.2. Utiliser une copie de travail nichée

Créez un nouveau dossier dans votre projet pour contenir le code commun, mais ne l'ajoutez pas à Subversion

Sélectionnez TortoiseSVN → Extraire pour le nouveau dossier et extrayez une copie du code commun. Vous avez maintenant une copie de travail séparée emboîtée dans votre copie de travail principale.

Les deux copies de travail sont indépendantes. Quand vous livrez des changements au parent, les changements à la CdT emboîtée sont ignorés. De même quand vous mettez à jour le parent, la CdT emboîtée n'est pas mise à jour.

### B.6.3. Utiliser un emplacement relatif

Si vous utilisez le même code fondamental commun dans plusieurs projets et vous ne voulez pas en garder plusieurs copies de travail pour chaque projet qui l'utilise, vous pouvez juste l'extraire à un emplacement séparé qui est lié à tous les autres projets qui l'utilisent. Par exemple :

```
C:\Projets\Proj1
C:\Projets\Proj2
C:\Projets\Proj3
C:\Projets\Commun
```

et faites référence au code commun en utilisant un chemin relatif, par exemple `..\..\Commun\DSPcore`.

Si vos projets sont dispersés dans des emplacements sans rapport, vous pouvez utiliser une variante, qui consiste à mettre le code commun dans un emplacement et utilisez la substitution de lettre de disque pour affecter cet emplacement à quelque chose que vous pouvez coder en dur dans vos projets, par exemple Extrayez le code commun dans `D:\Documents\Framework` ou `C:\Documents and Settings\{login}\My Documents\framework` puis utilisez

```
SUBST X: "D:\Documents\framework"
```

pour créer l'affectation de disque utilisé dans votre code source. Votre code peut alors utiliser des emplacements absolus.

```
#include "X:\superio\superio.h"
```

Cette méthode ne fonctionnera que dans un environnement tout PC et vous devrez documenter les affectations de disque requises pour que votre équipe sache où se trouvent ces fichiers mystérieux. Cette méthode est strictement utilisée dans des environnements de développement fermés et n'est pas recommandée pour une utilisation générale.

## B.7. Créer un raccourci vers un référentiel

Si vous devez fréquemment ouvrir l'explorateur de référentiel à un emplacement particulier, vous pouvez créer un raccourci sur le bureau en utilisant l'interface d'automatisation de TortoiseProc. Créez simplement un nouveau raccourci et définissez la cible :

```
TortoiseProc.exe /command:repobrowser /path:"url/to/repository"
```

. Vous devez bien sûr inclure la véritable URL du référentiel.

## B.8. Ignorer les fichiers déjà versionnés

Si vous avez accidentellement ajouté des fichiers qui devraient avoir été ignorés, comment les retirez-vous du contrôle de version sans les perdre ? Peut-être que vous avez votre propre fichier de configuration d'IDE qui ne fait pas partie du projet, mais qui vous a pris beaucoup de temps à configurer juste comme vous l'aimez.

Si vous n'avez pas encore livré l'ajout, tout ce que vous avez alors à faire est d'utiliser TortoiseSVN → Revenir en arrière... pour annuler l'ajout. Vous devriez alors ajouter les fichiers à la liste des ignorés pour qu'ils ne soient plus ajoutés plus tard par erreur à nouveau.

Si les fichiers sont déjà dans le référentiel, vous devez faire un peu plus de travail.

1. Gardez la touche **Shift** appuyée pour afficher le menu contextuel étendu et utilisez TortoiseSVN → Supprimer (garder en local) pour marquer le fichier/répertoire comme étant destiné à être supprimé du référentiel sans perdre la copie locale.
2. TortoiseSVN → Livraison le répertoire parent.
3. Ajoutez le fichier/dossier à la liste des ignorés pour que vous n'ayez plus le problème.

## B.9. Retirer une copie de travail du contrôle de version

Si vous souhaitez retirer tous les répertoires `.svn` de votre version de travail vous pouvez tout simplement utiliser la commande export. Lisez [Section 4.26.1](#), « Retirer une copie de travail du contrôle de version » pour plus d'informations sur cette commande.

## B.10. Retirer une copie de travail

Si vous avez une copie de travail dont vous n'avez plus besoin, comment pouvez-vous vous en débarrasser proprement? Facile - il suffit de le supprimer dans l'Explorateur Windows! Les copies de travail sont des entités privées locales, et elles sont autonomes.

---

# Annexe C. Trucs Utiles Pour Les Administrateurs

Cette annexe contient les solutions aux problèmes/questions que vous pourriez avoir quand vous êtes responsable du déploiement de TortoiseSVN sur plusieurs ordinateurs client.

## C.1. Déployer TortoiseSVN via les stratégies de groupe

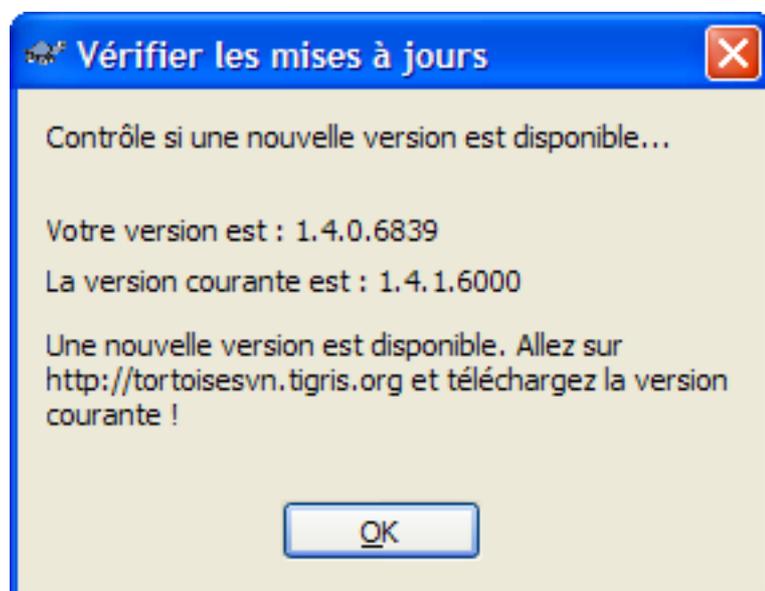
L'installateur TortoiseSVN est un fichier msi, ce qui signifie que vous ne devriez avoir aucun problème pour ajouter ce fichier msi à la stratégie de groupe de votre contrôleur de domaine.

Un bon guide sur la façon de faire cela peut être trouvé dans l'article de la base de connaissance 314934 de Microsoft : <http://support.microsoft.com/?kbid=314934>.

Les versions 1.3.0 et supérieures de TortoiseSVN doivent être installées dans *Configuration ordinateur* et pas dans *Configuration utilisateur*. C'est parce que ces versions ont besoin des nouvelles dlls CRT et MFC, qui peuvent être seulement déployés *par ordinateur* et non *par utilisateur*. Si vous devez vraiment installer TortoiseSVN sur une base par utilisateur, alors vous devez d'abord installer la version 8 du package MFC et CRT de Microsoft sur chaque ordinateur où vous voulez installer TortoiseSVN par utilisateur.

## C.2. Rediriger la vérification de mise à niveau

TortoiseSVN vérifie s'il y a une nouvelle version disponible tous les deux ou trois jours. S'il y a une version plus récente disponible, une boîte de dialogue s'affiche pour en informer l'utilisateur.



**Figure C.1. La boîte de dialogue Mettre à jour**

Si vous êtes responsable de beaucoup d'utilisateurs dans votre domaine, vous pourriez vouloir que vos utilisateurs n'utilisent que les versions que vous avez approuvées et qu'ils n'installent pas toujours la dernière version. Vous ne voulez pas probablement pas que la boîte de dialogue de mise à niveau s'affiche pour que vos utilisateurs n'aillent pas mettre à niveau immédiatement.

Les versions 1.4.0 et supérieures de TortoiseSVN vous permettent de rediriger cette vérification de mise à niveau vers votre serveur intranet. Vous pouvez définir la clé de base de registre HKCU\Software\TortoiseSVN\UpdateCheckURL (valeur chaîne) à une URL pointant sur un fichier texte dans votre intranet. Ce fichier texte doit avoir le format suivant :

1.4.1.6000

Un nouvelle version de TortoiseSVN est disponible au téléchargement !  
<http://192.168.2.1/downloads/TortoiseSVN-1.4.1.6000-svn-1.4.0.msi>

. La première ligne dans ce fichier est la chaîne de version. Vous devez vous assurer qu'elle corresponde à la chaîne de version exacte du package d'installation de TortoiseSVN. La deuxième ligne est un texte personnalisé, affiché dans la boîte de dialogue de mise à niveau. Vous pouvez écrire ici ce que vous voulez. Notez juste que l'espace dans la boîte de dialogue de mise à niveau est limité. De trop longs messages seront tronqués ! La troisième ligne est l'URL vers le nouveau package d'installation. Cette URL est ouverte quand l'utilisateur clique sur le label du message personnalisée dans la boîte de dialogue de mise à niveau. Vous pouvez aussi diriger simplement l'utilisateur vers une page Web plutôt que directement vers le fichier msi. L'URL est ouverte avec le navigateur Internet par défaut, donc si vous spécifiez une page Web, cette page est ouverte et montrée à l'utilisateur. Si vous spécifiez le package MSI, le navigateur demandera à l'utilisateur de sauvegarder le fichier MSI localement.

### C.3. Mettre la variable d'environnement SVN\_ASP\_DOT\_NET\_HACK

As of version 1.4.0 and later, the TortoiseSVN installer doesn't provide the user with the option to set the SVN\_ASP\_DOT\_NET\_HACK environment variable anymore, since that caused many problems and confusions with users which always install *everything* no matter if they know what it is for.

Mais cette option est seulement cachée pour l'utilisateur. Vous pouvez encore forcer l'installateur de TortoiseSVN à mettre cette variable d'environnement en mettant la propriété ASPDOTNETHACK à TRUE. Par exemple, vous pouvez démarrer l'installateur comme ceci :

```
msiexec /i TortoiseSVN-1.4.0.msi ASPDOTNETHACK=TRUE
```

### C.4. Désactiver les entrées du menu contextuel

Depuis la version 1.5.0 et ultérieure, TortoiseSVN vous permet de désactiver (en fait, masquer) des entrées du menu contextuel. Comme il s'agit d'une caractéristique qui ne doit pas être utilisé à la légère, mais seulement s'il y a une raison impérieuse, il n'y a pas d'interface graphique pour cela et ça doit être fait directement dans le Registre. Cela peut être utilisé pour désactiver certaines commandes pour les utilisateurs qui ne devraient pas les utiliser. Mais veuillez noter que les entrées du menu contextuel sont seulement cachées dans *l'explorateur*, et que les commandes sont toujours disponibles par d'autres moyens, par exemple la ligne de commande ou même d'autres boîtes de dialogue dans TortoiseSVN lui-même!

Les clés de registre qui contiennent les informations pour savoir quels menus contextuels afficher sont HKEY\_CURRENT\_USER\Software\TortoiseSVN\ContextMenuEntriesMaskLow et HKEY\_CURRENT\_USER\Software\TortoiseSVN\ContextMenuEntriesMaskHigh.

Chacune de ces entrées de Registre est une valeur DWORD, chaque bit correspondant à une entrée de menu spécifique. Un bit activé signifie que l'entrée de menu correspondante est désactivée.

Valeur	Entrée du menu
0x0000000000000001	Extraire
0x0000000000000002	Mettre à jour

Valeur	Entrée du menu
0x0000000000000004	Livrer
0x0000000000000008	Ajouter
0x0000000000000010	Revenir en arrière
0x0000000000000020	Nettoyer
0x0000000000000040	Résoudre
0x0000000000000080	Aller sur...
0x0000000000000100	Importer
0x0000000000000200	Exporter
0x0000000000000400	Créer un référentiel ici
0x0000000000000800	Branche/Etiquette
0x0000000000001000	Fusionner
0x0000000000002000	Supprimer
0x0000000000004000	Renommer
0x0000000000008000	Mettre à jour à la révision
0x0000000000010000	Voir les différences
0x0000000000020000	Voir le journal
0x0000000000040000	Éditer les conflits
0x0000000000080000	Relocaliser
0x0000000000100000	Vérifier les modifications
0x0000000000200000	Ignorer
0x0000000000400000	Explorateur de référentiel
0x0000000000800000	Annoter
0x0000000001000000	Créer un patch
0x0000000002000000	Appliquer un patch
0x0000000004000000	Graphique de révision
0x0000000008000000	Verrouiller
0x0000000010000000	Relâcher un verrou
0x0000000020000000	Propriétés
0x0000000040000000	Comparer avec l'URL
0x0000000080000000	Supprimer les éléments non versionnés
0x2000000000000000	Réglages
0x4000000000000000	Aide
0x8000000000000000	À propos

**Tableau C.1. Entrées du menu et leurs valeurs**

Exemple: pour désactiver les entrées de menu « Déplacer », « Supprimer les éléments non versionnés » et « Paramètres », ajouter les valeurs attribuées aux entrées suivantes:

```

0x0000000000008000
+ 0x0000000080000000
+ 0x2000000000000000

```

= 0x2000000080080000

La valeur la plus basse de DWORD (0x80080000) doit alors être stockée dans HKEY\_CURRENT\_USER\Software\TortoiseSVN\ContextMenuEntriesMaskLow, la plus élevée (0x20000000) dans HKEY\_CURRENT\_USER\Software\TortoiseSVN\ContextMenuEntriesMaskHigh.

Pour réactiver les entrées du menu, supprimer simplement les deux clés de registre.

---

# Annexe D. Automatiser TortoiseSVN

Puisque toutes les commandes pour TortoiseSVN sont contrôlées par des paramètres de ligne de commande, vous pouvez l'automatiser avec des scripts batch ou démarrer des commandes spécifiques et des boîtes de dialogues depuis d'autres programmes (par exemple votre éditeur de texte favori).



## Important

Souvenez-vous que TortoiseSVN est un client GUI et ce guide d'automatisation vous montre comment faire les boîtes de dialogues de TortoiseSVN apparaissent pour collecter les entrées utilisateur. Si vous voulez écrire un script qui n'exige aucune entrée, vous devriez utiliser le client en ligne de commande Subversion officiel à la place.

## D.1. Commandes de TortoiseSVN

Le programme GUI de TortoiseSVN s'appelle `TortoiseProc.exe`. Toutes les commandes sont spécifiées avec le paramètre `/command:abcd` où `abcd` est le nom de la commande requise. La plupart de ces commandes ont besoin d'au moins un argument de chemin, que l'on donne avec `/path:"un\chemin"`. Dans la table suivante, la commande fait référence au paramètre `/command:abcd` et le chemin se réfère au paramètre `/path:"un\chemin"`.

Puisque certaines des commandes peuvent prendre une liste de chemins cibles (par exemple livrer plusieurs fichiers spécifiques) le paramètre `/path` peut prendre plusieurs chemins, séparés par un caractère `*`.

TortoiseSVN utilise des fichiers temporaires pour passer des arguments multiples entre l'extension du shell et le programme principal. À partir de la version 1.5.0 et supérieures, le paramètre `/notempfile` est obsolète et il est inutile de l'ajouter désormais.

D'habitude, la barre de progression utilisée pour les livraisons, les mises à jour et beaucoup d'autres commandes reste ouverte après que la commande ait fini et jusqu'à ce que l'utilisateur appuie sur le bouton OK. Ce comportement peut être modifié en cochant l'option correspondante dans la boîte de dialogue de configuration. Mais ainsi la barre de progression se fermera à la fin de l'opération, que vous ayez lancé la commande depuis un fichier batch ou depuis le menu contextuel TortoiseSVN.

Pour spécifier un emplacement différent du fichier de configuration, utilisez le paramètre `/configdir:"chemin\vers\répertoire\de\conf"`. Cela remplacera le chemin par défaut, y compris tous les paramètres de la base de registre.

To close the progress dialog at the end of a command automatically without using the permanent setting you can pass the `/closeonend` parameter.

- `/closeonend:0` ne ferme pas la boîte de dialogue automatiquement
- `/closeonend:1` ferme automatiquement s'il n'y a pas d'erreurs
- `/closeonend:2` ferme automatiquement s'il n'y a pas d'erreurs ni de conflits
- `/closeonend:2` ferme automatiquement s'il n'y a pas d'erreurs, de conflits ni de fusions
- `/closeonend:4` ferme automatiquement s'il n'y a pas d'erreurs, de conflits ni de fusions pour les opérations locales

Le tableau ci-dessous liste toutes les commandes qui peuvent être accessibles en utilisant la ligne de commande `TortoiseProc.exe`. Comme décrit ci-dessus, celles-ci devraient être de la forme `/command:abcd`. Dans le tableau, le préfixe `/command` est omis pour économiser de la place.

Commande	Description
:about	Affiche la boîte de dialogue d'À propos. Elle s'affiche aussi si aucune commande n'est fournie.
:log	Opens the log dialog. The <code>/path</code> specifies the file or folder for which the log should be shown. Three additional options can be set: <code>/startrev:xxx</code> , <code>/endrev:xxx</code> and <code>/strict</code>
:checkout	Ouvre la boîte de dialogue d'extraction. Le <code>/path</code> spécifie le répertoire cible et le <code>/url</code> spécifie l'URL à extraire.
:import	Ouvre la boîte de dialogue d'Importation. Le <code>/path</code> spécifie le répertoire contenant les données à importer.
:update	Updates the working copy in <code>/path</code> to HEAD. If the option <code>/rev</code> is given then a dialog is shown to ask the user to which revision the update should go. To avoid the dialog specify a revision number <code>/rev:1234</code> . Other options are <code>/nonrecursive</code> and <code>/ignoreexternals</code> .
:commit	Ouvre la boîte de dialogue de livraison. Le <code>/path</code> spécifie le répertoire cible ou la liste des fichiers à livrer. Vous pouvez aussi spécifier le commutateur <code>/logmsg</code> pour passer un commentaire prédéterminé à la boîte de dialogue de livraison. Ou, si vous ne voulez pas passer le commentaire via la ligne de commande, utilisez <code>/logmsgfile:chemin</code> , où <code>chemin</code> pointe sur un fichier contenant le commentaire. Pour préremplir le champ d'ID de bug (dans le cas où vous avez défini correctement la propriété d'intégration aux traqueurs de bug), vous pouvez utiliser le <code>/bugid: "l'id du bug ici"</code> .
:add	Ajoute les fichiers de <code>/path</code> au contrôle de version.
:revert	Annule les modifications locales d'une copie de travail. Le <code>/path</code> indique quels éléments annuler.
:cleanup	Nettoie les opérations interrompues ou annulées et déverrouille la copie de travail dans <code>/path</code> .
:resolve	Marque un fichier en conflit indiqué dans <code>/path</code> comme résolu. Si <code>/noquestion</code> est donné, alors la résolution est faite sans demander d'abord à l'utilisateur si cela doit être vraiment fait.
:repopulate	Crée un référentiel dans <code>/path</code>
:switch	Ouvre la fenêtre Aller sur. Le <code>/path</code> spécifie le répertoire cible.
:export	Exporte dans un autre répertoire la copie de travail située dans <code>/path</code> . Si le <code>/path</code> pointe vers un répertoire non versionné, une boîte de dialogue demandera une URL à laquelle exporter le dossier dans <code>/path</code> .
:merge	Ouvre la fenêtre de fusion. Le <code>/path</code> spécifie le répertoire cible. Pour fusionner une plage de révisions, les options disponibles sont les suivantes : <code>/fromurl:URL</code> , <code>/revrange:string</code> . Pour fusionner deux arborescences du référentiel les options sont les suivantes : <code>/fromurl:URL</code> , <code>/tourl:URL</code> , <code>/fromrev:xxx</code> et <code>/torev:xxx</code> . Elles pré-remplissent les champs correspondants dans la fenêtre de fusion.
:mergeall	Ouvre la fenêtre fusionner tout. Le <code>/path</code> spécifie le répertoire cible.
:copy	Ouvre la fenêtre des branches/tags. Le <code>/path</code> est la copie de travail de laquelle faire une branche ou un tag. Et le <code>/url</code> est l'URL de destination. Vous pouvez également spécifier le <code>/logmsg</code> pour passer un message prédéfini à la fenêtre des branches/tags. Ou si vous voulez passer le message en ligne de commande, utilisez <code>/logmsgfile:chemin</code> , où <code>chemin</code> pointe sur un fichier contenant le message.
:settings	Ouvre la boîte de dialogue de configuration.

Commande	Description
:remove	Supprime les fichiers dans <code>/path</code> du contrôle de version.
:rename	Renomme le fichier dans <code>/path</code> . Le nouveau nom pour le fichier est demandé par une boîte de dialogue. Pour éviter la question concernant le renommage de fichiers similaires en une étape, passez <code>/noquestion</code> .
:diff	Starts the external diff program specified in the TortoiseSVN settings. The <code>/path</code> specifies the first file. If the option <code>/path2</code> is set, then the diff program is started with those two files. If <code>/path2</code> is omitted, then the diff is done between the file in <code>/path</code> and its BASE. To explicitly set the revision numbers use <code>/startrev:xxx</code> and <code>/endrev:xxx</code> . If <code>/blame</code> is set and <code>/path2</code> is not set, then the diff is done by first blaming the files with the given revisions.
:showcompare	<p>Selon les URL et les versions à comparer, ceci affiche soit un contenu unifié des différences (si l'option <code>unifié</code> est utilisée), soit une boîte de dialogue avec une liste de fichiers qui ont changé, soit, si les URL pointent vers les fichiers de démarrage, lance le visualisateur de différence pour ces deux fichiers.</p> <p>Les options <code>url1</code>, <code>url2</code>, <code>revision1</code> et <code>revision2</code> doivent être précisées. Les options <code>pegrevision</code>, <code>ignoreancestry</code>, <code>blame</code> et <code>unified</code> sont facultatives.</p>
:conflicteditor	Démarre l'éditeur de conflit indiqué dans la configuration de TortoiseSVN avec les fichiers corrects pour le fichier en conflit dans <code>/path</code> .
:relocate	Ouvre la boîte de dialogue Relocaliser. Le <code>/path</code> spécifie le chemin de la copie de travail à relocaliser.
:help	Ouvre le fichier d'aide.
:repostatus	Ouvre la fenêtre vérifier les modifications. Le <code>/path</code> spécifie le répertoire de la copie de travail.
:repobrowser	Ouvre la fenêtre de l'explorateur de référentiel, pointant sur l'URL de la copie de travail donnée dans <code>/path</code> ou <code>/path</code> pointe directement sur une URL. Une option supplémentaire <code>/rev:xxx</code> peut être utilisée pour spécifier la révision devant être affichée. Si le <code>/rev:xxx</code> n'est pas renseigné, la version de tête est utilisée par défaut. Si <code>/path</code> pointe sur une URL, le <code>/projectpropertiespath:path/to/wc</code> spécifie le chemin à partir duquel il faut lire et utiliser les propriétés du projet.
:ignore	Ajoute toutes les cibles dans <code>/path</code> à la liste des éléments ignorés, c'est-à-dire ajoute le <code>svn:ignore</code> à ces fichiers.
:blame	<p>Ouvre la fenêtre de bannissement pour le fichier spécifier dans <code>/path</code>.</p> <p>Si les options <code>/startrev</code> et <code>/endrev</code> sont précisées, la fenêtre permettant de spécifier la plage de révisions à bannir n'est pas affichée, ces valeurs des révisions sont utilisées à la place.</p> <p>Si cette option est renseignée <code>/line:nnn</code>, TortoiseBlame ouvrira en affichant la nième ligne.</p> <p>Les options <code>/ignoreeol</code>, <code>/ignorespaces</code> et <code>/ignoreallspaces</code> sont également supportées.</p>
:cat	Enregistre un fichier depuis une URL ou depuis un chemin de la copie de travail donné dans <code>/path</code> à l'emplacement donné dans <code>/savepath:chemin</code> . La révision est donnée dans <code>/revision:xxx</code> . Cela peut être utilisé pour obtenir un fichier avec une révision spécifique.

Commande	Description
:createpatch	Crée un patch pour le chemin donné dans /path.
:revisiongraph	Montre le graphique de révision pour le chemin donné dans /path.
:lock	Verrouille un ou tous les fichiers dans un répertoire donné dans /path. La boîte de dialogue 'Verrouiller' s'affiche afin de permettre à l'utilisateur d'entrer un commentaire pour le verrou.
:unlock	Déverrouille un fichier ou tous les fichiers d'un répertoire donné dans /path.
:rebuildiconcache	Reconstruit le cache d'icône Windows. Utilisez-le seulement dans le cas où les icônes Windows sont corrompues. Un effet secondaire à cela (qui ne peut être évité) est que les icônes sur le bureau sont réarrangées. Pour supprimer la fenêtre d'information, passez /noquestion.
:properties	Affiche la boîte de dialogue de propriétés pour le chemin donné dans /path.

### Tableau D.1. Liste des commandes et des options disponibles

Exemples (qui devraient être saisis sur une ligne):

```
TortoiseProc.exe /command:commit /path:"c:\svn_ct\fichier1.txt*c:"
                /logmsg:"message de log de test" /closeonend
```

```
TortoiseProc.exe /command:update /path:"c:\svn_ct\" /closeonend
```

```
TortoiseProc.exe /command:log /path:"c:\svn_ct\fichier1.txt"
                /startrev:50 /endrev:60 /closeonend
```

## D.2. Commandes de TortoiseIDiff

L'outil de comparaison d'images a quelques options en ligne de commande que vous pouvez utiliser pour contrôler la façon dont l'outil démarre. Le programme est appelé TortoiseIDiff.exe.

Le tableau suivant fait la liste des options pouvant être passées en ligne de commande à l'outil de comparaison d'images.

Option	Description
:left	Chemin du fichier affiché à gauche.
:lefttitle	Une chaîne de titre. Cette chaîne est utilisée dans le titre de la vue image au lieu du chemin complet du fichier image.
:right	Chemin du fichier affiché à droite.
:righttitle	Une chaîne de titre. Cette chaîne est utilisée dans le titre de la vue image au lieu du chemin complet du fichier image.
:overlay	Si spécifié, l'outil de comparaison d'images bascule vers le mode d'icône de recouvrement (alpha).
:fit	Si spécifié, l'outil de différenciation d'image fait concorder les deux images ensemble.
:showinfo	Affiche la boîte d'informations sur l'image

### Tableau D.2. Liste des options disponibles

Exemple (qui doit tenir sur une seule ligne):

```
TortoiseIDiff.exe /left:"c:\images\img1.jpg" /lefttitle:"image 1"  
/right:"c:\images\img2.jpg" /righttitle:"image 2"  
/fit /overlay
```

---

# Annexe E. Référence croisée de l'interface en ligne de commande

Parfois ce manuel se réfère à la documentation principale de Subversion, qui décrit Subversion en termes d'Interface de Ligne de Commande (ILC). Pour vous aider à comprendre ce que fait TortoiseSVN en coulisses, nous avons compilé une liste montrant les commandes ILC équivalentes pour chacune des opérations GUI de TortoiseSVN.

## Note

Bien qu'il y ait des équivalents ILC à ce que fait TortoiseSVN, souvenez-vous que TortoiseSVN ne fait *pas* appel à l'ILC mais utilise la bibliothèque de Subversion directement.

Si vous pensez avoir trouvé un bug dans TortoiseSVN, nous pouvons vous demander d'essayer de le reproduire en utilisant l'ILC, pour que nous puissions distinguer les incidents de TortoiseSVN de ceux de Subversion. Cette référence vous dit quelle commande essayer.

## E.1. Conventions et règles de base

In the descriptions which follow, the URL for a repository location is shown simply as URL, and an example might be `http://tortoisesvn.googlecode.com/svn/trunk`. The working copy path is shown simply as PATH, and an example might be `C:\TortoiseSVN\trunk`.



## Important

Parce que TortoiseSVN est une extension du shell Windows, il n'est pas capable d'utiliser la notion d'un répertoire de travail courant. Tous les chemins de la copie de travail doivent être donnés en utilisant le chemin absolu, pas le chemin relatif.

Certains éléments sont facultatifs et ceux-ci sont souvent contrôlés par des cases à cocher ou des boutons radio dans TortoiseSVN. Ces options sont affichées dans des [crochets] dans les définitions de ligne de commande.

## E.2. Commandes de TortoiseSVN

### E.2.1. Extraire

```
svn checkout [-N] [--ignore-externals] [-r rev] URL PATH
```

Si Extraire seulement le répertoire de tête est coché, utilisez le commutateur `-N`.

Si Omettre les références externes est coché, utilisez le commutateur `--ignore-externals`.

Si vous extrayez une révision spécifique, spécifiez cela après l'URL en utilisant le commutateur `-r`.

### E.2.2. Mettre à jour

```
svn info URL_de_la_CdT
svn update [-r rev] CHEMIN
```

Mettre à jour plusieurs éléments n'est pas actuellement une opération atomique dans Subversion. Donc TortoiseSVN trouve d'abord la révision HEAD du référentiel et met ensuite à jour tous les éléments à ce numéro de révision particulier pour éviter de créer une copie de travail avec des révisions mélangées.

Si un seul élément est sélectionné à mettre à jour ou si les éléments choisis ne sont pas tous du même référentiel, TortoiseSVN met simplement à jour à HEAD.

Aucune option de ligne de commande n'est utilisée ici. **Mettre à jour à la révision** met aussi en oeuvre la commande de mise à jour, mais offre plus d'options.

### E.2.3. Mettre à jour à la révision

```
svn info URL_of_WC
svn update [-r rev] [-N] [--ignore-externals] PATH
```

Si Mettre à jour seulement le répertoire de tête est coché, utilisez le commutateur `-N`.

Si Omettre les références externes est coché, utilisez le commutateur `--ignore-externals`.

### E.2.4. Livrer

Dans TortoiseSVN, la boîte de dialogue livrer utilise plusieurs commandes Subversion. La première étape est une vérification de statut qui détermine les éléments de votre copie de travail qui peuvent potentiellement être livrés. Vous pouvez passer en revue la liste, comparer les fichiers avec la BASE et les éléments que vous voulez inclure dans la livraison.

```
svn status -v CHEMIN
```

Si Afficher les fichiers non versionnés est coché, TortoiseSVN affichera aussi tous les fichiers et tous les dossiers non versionnés dans la hiérarchie de la copie de travail, en prenant en compte les règles d'exclusion. Cette fonctionnalité particulière n'a aucun équivalent direct dans Subversion, puisque la commande `svn status` ne parcourt pas les dossiers non versionnés.

Si vous sélectionnez des fichiers ou des dossiers non versionnés, ces éléments seront d'abord ajoutés à votre copie de travail.

```
svn add CHEMIN...
```

Quand vous cliquez sur OK, la livraison Subversion se produit. Si vous avez laissé toutes les cases de sélection de fichier dans leur état par défaut, TortoiseSVN utilise une seule livraison récursive de la copie de travail. Si vous désélectionnez quelques fichiers, alors une livraison non récursive (`-N`) doit être utilisée et chaque chemin doit être spécifié individuellement sur la ligne de commande de livraison.

```
svn commit -m "LogMessage" [-N] [--no-unlock] PATH...
```

Commentaire représente ici le contenu de la boîte de saisie du commentaire. Cela peut être vide.

Si Garder les verrous est coché, utilisez le commutateur `--no-unlock`.

### E.2.5. Voir les différences

```
svn diff CHEMIN
```

Si vous utilisez Voir les différences depuis le menu contextuel principal, vous comparez un fichier modifié avec sa version de BASE. La sortie de la commande de l'ILC ci-dessus fait la même chose et génère

une sortie au format unified-diff. Cependant, ce n'est pas ce qu'utilise TortoiseSVN. TortoiseSVN utilise TortoiseMerge (ou le programme de comparaison de votre choix) pour afficher les différences entre des fichiers purement texte, donc il n'y a aucun équivalent dans l'ILC.

Vous pouvez aussi comparer 2 fichiers en utilisant TortoiseSVN, qu'ils soient sous contrôle de version ou non. TortoiseSVN alimente simplement les deux fichiers dans le programme de comparaison choisi et laisse rechercher où se trouvent les différences.

### E.2.6. Voir le journal

```
svn log -v -r 0:N --limit 100 [--stop-on-copy] CHEMIN  
ou  
svn log -v -r M:N [--stop-on-copy] CHEMIN
```

Par défaut, TortoiseSVN essaye de récupérer 100 commentaires en utilisant la méthode `--limit`. Si les réglages indiquent d'utiliser de vieilles API, alors la deuxième forme est utilisée pour aller chercher les commentaires de 100 révisions du référentiel.

Si Arrêt à la copie/renommage est coché, utilisez le commutateur `--stop-on-copy`.

### E.2.7. Vérifier les modifications

```
svn status -v CHEMIN  
ou  
svn status -u -v CHEMIN
```

La vérification initiale du statut ne regarde que votre copie de travail. Si vous cliquez sur **Vérifier le référentiel** alors le référentiel est aussi vérifié pour voir quels fichiers seraient changés par une mise à jour, ce qui exige le commutateur `-u`.

Si **Afficher les fichiers non versionnés** est coché, TortoiseSVN affichera aussi tous les fichiers et tous les dossiers non versionnés dans la hiérarchie de la copie de travail, en prenant en compte les règles d'exclusion. Cette fonctionnalité particulière n'a aucun équivalent direct dans Subversion, puisque la commande `svn status` ne parcourt pas les dossiers non versionnés.

### E.2.8. Graphique de révision

Le graphique de révision est une fonctionnalité de TortoiseSVN uniquement. Il n'y a pas d'équivalent pour le client en ligne de commande.

Ce que fait TortoiseSVN est

```
svn info URL_de_la_CdT  
svn log -v URL
```

où l'URL est la *racine* du référentiel et analyse ensuite les données renvoyées.

### E.2.9. Explorateur de référentiel

```
svn info URL_de_la_CdT  
svn list [-r rev] -v URL
```

Vous pouvez utiliser `svn info` pour déterminer la racine du référentiel, qui est le niveau supérieur affiché dans l'explorateur de référentiel. Vous ne pouvez pas naviguer vers le haut au-dessus de ce niveau. Aussi, cette commande renvoie toute l'information de verrouillage affichée dans l'explorateur de référentiel.

L'appel `svn list` listera le contenu d'un répertoire, à l'URL et la révision données.

### E.2.10. Éditer les conflits

Cette commande n'a aucun équivalent en ILC. Elle appelle TortoiseMerge ou un outil externe de comparaison/fusion à 3 vues pour regarder les fichiers impliqués dans le conflit et déterminer quelles lignes utiliser.

### E.2.11. Résolu

```
svn resolved CHEMIN
```

### E.2.12. Renommer

```
svn rename CHEMIN_COURANT NOUVEAU_CHEMIN
```

### E.2.13. Supprimer

```
svn delete CHEMIN
```

### E.2.14. Revenir en arrière

```
svn status -v CHEMIN
```

La première étape est un contrôle de statut qui détermine les éléments de votre copie de travail qui peuvent être potentiellement annulés. Vous pouvez examiner la liste, comparer les fichiers contre la BASE et choisir les éléments que vous voulez inclure dans le retour en arrière.

Quand vous cliquez sur OK, le retour en arrière de Subversion se produit. Si vous avez laissé toutes les cases de sélection de fichier dans leur état par défaut, TortoiseSVN utilise un seul retour en arrière récursif (-R) de la copie de travail. Si vous désélectionnez quelques fichiers, alors chaque chemin doit être spécifié individuellement sur la ligne de commande de retour en arrière.

```
svn revert [-R] CHEMIN...
```

### E.2.15. Nettoyer

```
svn cleanup CHEMIN
```

### E.2.16. Obtenir un verrou

```
svn status -v CHEMIN
```

La première étape est une vérification de statut qui détermine les fichiers de votre copie de travail qui peuvent être potentiellement verrouillés. Vous pouvez choisir les éléments que vous voulez verrouiller.

```
svn lock -m "Commentaire du verrou" [--force] CHEMIN...
```

Commentaire du verrou représente ici le contenu de la boîte de saisie de commentaire de verrou. Il peut être vide.

Si Voler les verrous est coché, utilisez le commutateur `--force`.

## E.2.17. Relâcher un verrou

```
svn unlock CHEMIN
```

## E.2.18. Branche/Étiquette

```
svn copy -m "Commentaire" URL URL
ou
svn copy -m "Commentaire" URL@rev URL@rev
ou
svn copy -m "Commentaire" CHEMIN URL
```

La boîte de dialogue de Branche/Étiquette exécute une copie vers le référentiel. Il y a 3 boutons radio d'options :

- Révision HEAD dans le référentiel
- Révision spécifique dans le référentiel
- Copie de travail

qui correspondent aux 3 variantes de ligne de commande ci-dessus.

Commentaire représente ici le contenu de la boîte de saisie du commentaire. Cela peut être vide.

## E.2.19. Aller sur...

```
svn info URL_de_la_CdT
svn switch [-r rev] URL CHEMIN
```

## E.2.20. Fusionner

```
svn merge [--dry-run] URL_Depuis@revN URL_Vers@revM CHEMIN
```

Le bouton Fusion de test exécute la même fusion mais avec l'option `--dry-run`.

```
svn diff URL_Depuis@revN URL_Vers@revM
```

Le Diff unifiée affiche l'opération de comparaison qui sera utilisée pour faire la fusion.

## E.2.21. Exporter

```
svn export [-r rev] [--ignore-externals] URL CHEMIN_Export
```

Cette forme est utilisée lors d'un accès depuis un dossier non versionné et le dossier est utilisé comme destination.

L'exportation d'une copie de travail dans un emplacement différent est fait sans utiliser la bibliothèque de Subversion, donc il n'existe aucun équivalent en ligne de commande correspondant.

Ce que fait TortoiseSVN est une copie de tous les fichiers vers le nouvel emplacement lors de l'affichage de la progression de l'opération. Les fichiers/dossiers non versionnés peuvent être aussi exportés facultativement .

Dans les deux cas, si Omettre les références externes est coché, utilisez le commutateur `--ignore-externals`.

### E.2.22. Relocaliser

```
svn switch --relocate URL_Depuis URL_Vers
```

### E.2.23. Créer un référentiel ici

```
svnadmin create --fs-type fsfs PATH
```

### E.2.24. Ajouter

```
svn add CHEMIN...
```

Si vous avez choisi un dossier, TortoiseSVN le parcourt d'abord récursivement pour les éléments qui peuvent être ajoutés.

### E.2.25. Importer

```
svn import -m Commentaire CHEMIN URL
```

Commentaire représente ici le contenu de la boîte de saisie du commentaire. Cela peut être vide.

### E.2.26. Annoter

```
svn blame -r N:M -v CHEMIN
```

```
svn log -r N:M CHEMIN
```

Si vous utilisez TortoiseBlame pour voir les informations de bannissement, le fichier de log est également requis pour afficher les messages de log dans une info-bulle. Si vous voyez le bannissement comme un fichier texte, cette information n'est pas exigée.

### E.2.27. Ajouter à la liste des ignorés

```
svn propget svn:ignore CHEMIN > FichierTemp  
{éditez le nouvel élément à ignorer dans FichierTemp}  
svn propset svn:ignore -F FichierTemp CHEMIN
```

Parce que la propriété `svn:ignore` est souvent composée de plusieurs lignes, elle est montrée ici comme étant modifiée via un fichier texte plutôt que directement en ligne de commande.

### E.2.28. Créer un patch

```
svn diff CHEMIN > fichier_patch
```

TortoiseSVN crée un patch dans le format de différences unifiées en comparant la copie de travail avec sa version de BASE.

### E.2.29. Appliquer un patch

Appliquer un patch est une activité difficile à moins que le patch et la copie de travail ne soient à la même révision. Heureusement pour vous, vous pouvez utiliser TortoiseMerge, qui n'a aucun équivalent direct dans Subversion.

---

# Annexe F. Détails de l'implémentation

Cette annexe contient plus de détails concernant l'implémentation de quelques fonctionnalités de TortoiseSVN.

## F.1. Recouvrement d'icônes

Chaque fichier et dossier a une valeur de statut Subversion tel que rapporté par la bibliothèque de Subversion. Dans le client de ligne de commande, elles sont représentées par des codes d'une seule lettre, mais dans TortoiseSVN elles sont représentées graphiquement en utilisant les icônes d'avant-plan. Parce que le nombre d'avant-plan est très limité, chacun d'eux peut représenter une ou plusieurs valeurs de statut.



L'icône de recouvrement *En Conflit* est utilisée pour représenter un état en `conflict`, là où une mise à jour génère des conflits entre la version locale et la version du référentiel. Elle est aussi utilisée pour un état bloqué, qui peut se produire quand une opération ne se termine pas correctement.



L'icône de recouvrement *Modifié* représente un état modifié, i.e. lorsque vous avez fait des modifications, l'état `fusionné` se produit lorsque les versions du référentiel ont changé et qu'elles ont été intégrées à la version locale, et l'état `remplacé` se produit lorsqu'un fichier a été supprimé et remplacé par un autre ayant le même nom mais dont le contenu est différent.



L'icône de recouvrement *Supprimé* représente un état supprimé, i.e. lorsqu'un élément a été marqué comme étant à supprimer, ou un état manquant, i.e. lorsqu'un élément n'est pas présent en local. Naturellement un élément qui manque ne peut avoir lui-même d'icône de recouvrement, mais le répertoire le contenant le peut.



L'overlay indique juste qu'un fichier ou un dossier a été ajouté au contrôle de version.



L'icône de recouvrement *Dans Subversion* est utilisé pour représenter un élément qui est dans un état normal, ou un élément sous contrôle de version dont l'état n'est pas encore connu. TortoiseSVN fonctionne avec un système de mise en cache en arrière plan pour récupérer les états, les mises à jours des icônes de recouvrement peuvent donc prendre quelques secondes.



L'icône de recouvrement *A Besoin d'être Verrouillé* est utilisé pour indiquer que la propriété `svn:needs-lock` est activée pour un fichier. Pour les copies de travail créées avec les versions 1.4.0 et ultérieures de Subversion, l'état de la propriété `svn:needs-lock` est mis en cache localement par Subversion et cette valeur est utilisée pour activer ou non l'icône de recouvrement. Pour les copies de travail créées avec des versions antérieures, TortoiseSVN utilise cette icône lorsque le fichier est en lecture seule. Notez que Subversion met à jour automatiquement le format des copies de travail lorsque vous faites une mise à jour, de ce fait la mise en cache de l'état de la propriété `svn:needs-lock` ne devrait pas se produire à moins que le fichier lui-même ne soit pas à jour.



L'icône de recouvrement *Verrouillé* est utilisée lorsque le fichier est verrouillé dans la copie de travail.



L'icône de recouvrement *Ignoré* indique qu'un élément est dans un état ignoré, soit car il satisfait une condition globale (global pattern) soit car il satisfait une condition du dossier parent. Cette icône de recouvrement est optionnelle.



L'icône de recouvrement *non versionné* est utilisé pour représenter un élément étant dans l'état non versionné. C'est à dire un élément situé dans un répertoire sous contrôle de version, mais qui n'est pas lui-même sous contrôle de version. Cette icône de recouvrement est optionnelle.

Si un élément a le statut de subversion aucun (l'élément n'est pas dans une copie de travail), aucune icône de recouvrement n'est montrée. Si vous avez choisi de désactiver les icônes de recouvrement *ignoré* et *non versionné* alors aucune icône de recouvrement ne sera montré pour ces fichiers aussi.

Un élément ne peut avoir qu'une seule valeur de statut Subversion. Par exemple, un fichier peut être modifié localement et il pourrait être marqué pour suppression dans le même temps. Subversion renvoie une valeur de statut unique - dans ce cas *supprimé*. Ces priorités sont définies au sein de Subversion lui-même.

Lorsque TortoiseSVN affiche le statut récursivement (le réglage par défaut), sur chaque dossier est affichée une icône reflétant son propre statut et celui de tous ses enfants. Pour afficher une icône qui *résumeemphasis> l'ensemble, nous utilisons l'ordre de prioriten conflit prenant la priorit*

En fait, vous pouvez constater que toutes ces icônes ne sont pas utilisées sur votre système. C'est dû au fait que le nombre de recouvrements permis par Windows est limité à 15. Windows en utilise 4 et les 11 restantes peuvent être utilisés par d'autres applications. S'il n'a pas assez d'emplacements libres, TortoiseSVN essaie d'être un « bon citoyen (TM) » et limite son utilisation des recouvrements pour laisser une chance aux autres applications.

- Normal, Modifié et En conflit sont toujours chargés et visibles.
- Supprimé est chargé si possible, mais devient Modifié s'il n'y a pas assez de connecteurs.
- Lecture seule est chargé si possible, mais redevient Normal s'il n'y a pas assez de connecteurs.
- Verrouillé est seulement chargé s'il y a moins de 13 recouvrements déjà chargés. Il devient Normal s'il n'y a pas assez d'emplacements.
- Ajouté est seulement chargé s'il y a moins de 14 recouvrements déjà chargés. Il devient Modifié s'il n'y a pas assez d'emplacements.

---

# Annexe G. Sécuriser Svnserve grâce à SSH

Cette section fournit un guide pas-à-pas pour configurer Subversion et TortoiseSVN pour utiliser le protocole `svn+ssh`. Si vous utilisez déjà une authentification SSH pour vous connecter à votre serveur, alors vous pouvez passer cette section et trouver plus de détails dans le livre Subversion. Si vous n'utilisez pas encore SSH mais aimeriez le faire pour protéger votre installation Subversion, ce guide donne une méthode simple qui n'implique pas la création d'un compte utilisateur SSH pour chaque utilisateur subversion présent sur le serveur.

Dans cette implémentation nous créons un unique compte utilisateur SSH pour tous les utilisateurs subversion et utilisons différentes clés d'authentification pour les différencier des véritables utilisateurs Subversion.

Dans cette annexe, nous supposons que vous avez déjà les outils subversion installés et que vous avez créé un dépôt comme expliqué ailleurs dans ce manuel. Notez bien que vous ne devriez *pas* démarrer svnserve en tant que service ou démon lorsqu'il est utilisé avec SSH.

Much of the information here comes from a tutorial provided by Marc Logemann, which can be found at [www.logemann.org](http://www.logemann.org) [<http://www.logemann.org/2007/03/13/subversion-tortoisesvn-ssh-howto/>] Additional information on setting up a Windows server was provided by Thorsten Müller. Thanks guys!

## G.1. Configurer un Serveur Liunx

Vous devez avoir SSH activé sur votre serveur, et dans ce cas nous supposons que vous utilisez OpenSSH. Dans la plupart des distributions, il sera déjà installé. Pour le savoir, tapez :

```
ps xa | grep sshd
```

et cherchez les processus ssh.

Un détail à relever est que si vous compilez Subversion depuis le code source sans préciser de paramètres au `./configure`, Subversion créé un répertoire `bin` dans `/usr/local` et y place les binaires. Si vous désirez utiliser le mode de tunnelage avec SSH, vous devez avoir conscience que l'utilisateur se connectant via SSH a besoin d'exécuter le programme `svnserve` et quelques autres binaires. Pour cette raison, placez `/usr/local/bin` dans la variable `PATH` ou créez les liens symboliques de vos binaires dans le répertoire `/usr/sbin`, ou dans tout autre répertoire communément inclus dans le `PATH`.

Vérifiez que tout est en ordre, connectez vous avec l'identifiant de l'utilisateur cible avec SSH et tapez :

```
which svnserve
```

Cette commande devrait vous indiquer si `svnserve` est accessible.

Crée un nouvel utilisateur que nous utiliserons pour accéder au dépôt svn:

```
useradd -m svnuser
```

Assurez-vous de donner à cette utilisateur tous les droits d'accès au dépôt.

## G.2. Configurer un Serveur Windows

Installer le démon Cygwin SSH comme décrit ici: <http://pigtail.net/LRP/printsrv/cygwin-sshd.html>

Créer un nouveau compte d'utilisateur Windows `svnuser` que nous utiliserons pour accéder au référentiel. Soyez sûr de donner à cet utilisateur les droits d'accès au référentiel.

S'il n'y a pas encore de fichier de mot de passe, créez-en un depuis une console Cygwin grâce à cette commande :

```
mkpasswd -l > /etc/passwd
```

### G.3. Client SSH à utiliser avec TortoiseSVN

Récupérez les outils dont nous avons besoin pour l'utilisation de SSH sur le client Windows à partir de ce site: <http://www.chiark.greenend.org.uk/~sgtatham/putty/> Il suffit d'aller à la section téléchargement et de prendre Putty, Plink, Pageant et Puttygen.

### G.4. Création des certificats OpenSSH

L'étape suivante consiste à créer une paire de clés pour l'authentification. Il ya deux façons de créer des clés. La première consiste à créer les clés avec PuTTYgen sur le client, télécharger la clé publique sur votre serveur et utiliser la clé privée avec PuTTY. L'autre est de créer la paire de clés avec l'outil OpenSSH `ssh-keygen`, télécharger la clé privée sur votre client et convertir la clé privée sous la forme d'une clé privée PuTTY.

#### G.4.1. Créer des clés en utilisant `ssh-keygen`

Connectez-vous sur le serveur en tant que `root` ou `svnuser` et tapez:

```
ssh-keygen -b 1024 -t dsa -N passphrase -f keyfile
```

en remplaçant par un véritable mot de passe (que vous seul(e) connaissez) et le fichier clé. Nous venons de créer une clé SSH2 DSA de 1024 bits. Si vous tapez

```
ls -l keyfile*
```

vous verrez deux fichiers, `keyfile` et `keyfile.pub`. Comme vous pouvez le deviner, le fichier `.pub` est le fichier de clé publique, l'autre est le privé.

Ajoute la clé publique à ceux du dossier `ssh` dans le répertoire racine de `svnuser`:

```
cat keyfile.pub >> /home/svnuser/.ssh/authorized_keys
```

Pour utiliser la clé privée que nous avons générée, nous devons la convertir dans un format `putty`. C'est parce que le format du fichier de la clé privée n'est pas spécifié par un organisme de normalisation. Après avoir téléchargé le fichier de clé privée à votre PC client, démarrez PuTTYgen et utilisez **Conversions** → **Importer les clés**. Accédez à votre fichier `keyfile` que vous avez obtenu à partir du serveur avec le mot de passe que vous avez utilisé lors de la création de la clé. Enfin cliquez sur **Enregistrer la clé privée** et enregistrez le fichier comme `keyfile.PPK`.

#### G.4.2. Créer des clés en utilisant PuTTYgen

Utilisez PuTTYgen afin de générer une paire de clé public/privée, et sauvegardez les. Copiez la clé public sur le serveur et ajoutez là à celles déjà présentes dans le dossier `.ssh`, sous le dossier dossier personnel `svnuser`:

```
cat keyfile.pub >> /home/svnuser/.ssh/authorized_keys
```

## G.5. Tester en utilisant PuTTY

Afin de tester la connectivité, nous utiliserons PuTTY. Démarrez le programme et à partir de l'onglet **Session**, spécifiez comme nom d'hôte le nom ou l'adresse de votre serveur, SSH comme protocole, et sauvegardez la session en tant que `SvnConnection` ou tout autre nom qui vous convient. À partir de l'onglet **SSH**, choisissez SSH version 2 comme protocole par défaut. À partir de l'onglet **Auth**, saisissez le chemin complet vers le fichier de clé privé `.PPK` que vous avez convertis précédemment. Retournez à l'onglet **Sessions** et cliquez sur le bouton **Sauvegarde**. Vous devriez maintenant voir `SvnConnection` dans la liste de vos sessions sauvegardées.

Cliquez sur **Ouvrir** et vous devriez voir une invite au format telnet. Utilisez `svnuser` comme nom d'utilisateur et si tout se passe comme prévu, vous devriez être authentifié sans qu'un mot de passe ne vous soit demandé.

Vous pouvez avoir à éditer `/etc/sshd_config` sur le serveur. Effectuez l'édition dans le fichier en accord avec ce qui suit et redémarrez le service SSH par la suite.

```
PubkeyAuthentication yes
PasswordAuthentication no
PermitEmptyPasswords no
ChallengeResponseAuthentication no
```

## G.6. Tester SSH avec TortoiseSVN

Jusqu'à maintenant, nous avons seulement vérifié que l'authentification fonctionne sous SSH. Maintenant, nous devons nous assurer que notre connexion SSH nous permet d'exécuter `svnserve`. Sur le serveur, modifiez `/home/svnuser/.ssh/authorized_keys` de la façon suivante afin de permettre à plusieurs auteurs subversion d'utiliser le même compte, `svnuser`. Veuillez noter que tous les auteurs subversion utilisent le même identifiant utilisateur, mais une clé d'authentification différente. Vous devez ainsi ajouter une ligne pour chaque auteur.

Note: Nous avons ici une seule ligne très longue et non coupée.

```
command="svnserve -t -r <ReposRootPath> --tunnel-user=<author> ",
      no-port-forwarding,no-agent-forwarding,no-X11-forwarding,
      no-pty ssh-rsa <PublicKey> <Comment>
```

Plusieurs paramètres sont à changer, dépendamment de votre configuration.

`<ReposRootPath>` should be replaced with the path to the directory containing your repositories. This avoids the need to specify full server paths within URLs. Note that you must use forward slashes even on a Windows server, e.g. `c:/svn/reposroot`. In the examples below we assume that you have a repository folder within the repository root called `repos`.

`<author>` doit être remplacé par l'auteur svn associé à l'envoi au dépôt. Cela permet également à `svnserve` d'utiliser ses propres droits d'accès au fichier `svnserve.conf`.

`<PublicKey>` devrait être remplacé avec la clé publique que vous avez généré plus tôt.

`<Comment>` peut être n'importe quel commentaire de votre choix, mais il est utile pour faire correspondre le nom d'un auteur svn avec son vrai nom.

Effectuez un clic droit de la souris sur n'importe quel dossier de l'explorateur Windows et choisissez **TortoiseSVN** → **Explorateur de référentiel**. Vous serez invités à saisir une URL, utilisez le format suivant:

```
svn+ssh://svnuser@SvnConnection/repos
```

Que signifie cette URL? Le nom du schéma est `svn+ssh` et permet d'indiquer à TortoiseSVN comment traiter les requêtes au serveur. Après le double slash, vous indiquez à l'utilisateur de s'authentifier au serveur, dans notre cas il s'agit de `svnuser`. Le nom de la session PuTTY doit être fournis après le `@`. Le nom de cette session contient tout les détails, comme où trouver la clé privée, l'adresse IP de notre serveur, ou le DNS. Finalement, nous devons fournir le chemin d'accès vers le référentiel, relatif au référentiel racine sur le serveur, tel que spécifié dans le fichier `authorized_keys`.

Cliquez sur le bouton OK et vous devriez par la suite être en mesure de naviguer dans le contenu du référentiel. Vous avez maintenant un tunnel SSH établis avec TortoiseSVN.

Notez que par défaut, TortoiseSVN utilise sa propre version de Plink pour établir la connexion. Cette stratégie permet d'empêcher l'affichage d'une fenêtre à chaque tentative d'authentification. Cependant, cela à pour conséquence que lorsqu'une erreur se produit, elle ne peut pas être affichée. Si vous recevez l'erreur « Impossible d'écrire sur la sortie standard », vous pouvez essayer de configurer Plink comme le client sous les paramètres réseaux de TortoiseSVN. Cela vous permettra de voir la vrai erreur générée par Plink.

## G.7. Variantes de Configuration SSH

Une manière de simplifier l'URL dans TortoiseSVN est de définir l'utilisateur à l'intérieur de la session PuTTY. Pour cela vous devez charger votre session prédéfinie `ConnexionSVN` dans PuTTY et dans l'onglet `Connexion` définir l' `utilisateur de la connexion automatique` sur le nom d'utilisateur, par exemple `svnuser`. Sauvegardez votre session PuTTY comme précédemment et essayez l'URL suivante dans TortoiseSVN:

```
svn+ssh://ConnexionSvn/repos
```

Cette fois-ci nous fournissons uniquement la `ConnexionSVN` de la session PuTTY au client SSH que TortoiseSVN utilise (`TortoisePlink.exe`). Ce client va contrôler tous les détails nécessaires de la session.

Au moment d'écrire PuTTY ne vérifie pas toutes les configurations sauvegardées, alors si vous avez de multiples configurations avec le même nom de serveur, PuTTY prendra la première qui correspond. De même, si vous éditez la configuration par défaut et que vous la sauvegardez, le nom d'utilisateur de la connexion automatique n'est *pas* sauvegardé.

Beaucoup de gens aiment à utiliser Pageant pour stocker toutes leurs clés. Comme il est possible de stocker une clé dans une session PuTTY, vous n'avez pas toujours besoin de Pageant. Cependant, imaginez que vous vouliez stocker des clés pour plusieurs serveurs différents; dans ce cas il vous faudra éditer la session PuTTY encore et encore, en fonction du serveur avec lequel vous essayez de vous connecter. Dans cette situation Pageant prend tout son sens, car PuTTY, Plink, TortoisePlink ou n'importe quel outil basé sur PuTTY, lorsqu'il essaie de se connecter à un serveur SSH, vérifie toutes les clés privées que Pageant détient pour initier la connexion.

Pour cette tâche, exécutez simplement Pageant et ajoutez la clé privée. Celle-ci devrait être la même clé privée que vous avez définie dans la session PuTTY au-dessus. Si vous utilisez Pageant pour le stockage de clés privées, vous pouvez effacer la référence au fichier de la clé privée dans votre session PuTTY sauvegardée. Vous pouvez ajouter plus de clés pour d'autres serveurs, ou bien entendu d'autres utilisateurs.

Si vous ne voulez pas répéter cette manipulation après chaque redémarrage du client, vous devez mettre Pageant dans la liste des programmes à exécuter au démarrage de Windows. Vous pouvez ajouter les clés avec leurs chemin complet comme arguments de `Pageant.exe`

La dernière façon de se connecter à un serveur SSH est tout simplement d'utiliser cette URL dans TortoiseSVN:

```
svn+ssh://svnuser@100.101.102.103/referentiel
```

```
svn+ssh://svnuser@mondomaine.com/referentiel
```

Comme vous pouvez le voir, nous n'utilisons pas une session PuTTY mais une adresse IP (ou un nom de domaine) comme cible. L'utilisateur est également stipulé, cela dit vous pourriez vous demander comment est trouvée la clé privée. Parce que TortoisePlink.exe n'est qu'une version modifiée de Plink, l'outil standard de la suite PuTTY, TortoiseSVN essaiera toutes les clés stockées dans Pageant.

Si vous utilisez cette dernière méthode, assurez-vous de ne pas avoir un nom d'utilisateur par défaut configuré dans PuTTY. Nous avons reçu pour ce cas de figure des signalements de bogues entraînant des coupures de connexion dans PuTTY. Pour supprimer l'utilisateur par défaut, effacez simplement `HKEY_CURRENT_USER\Software\SimonTatham\Putty\Sessions\Default %20Settings\HostName`

---

# Glossaire

Ajouter	Une commande Subversion utilisée pour ajouter un fichier ou un répertoire à votre copie de travail. Les nouveaux éléments sont ajoutés au référentiel à la livraison.
Aller sur...	De même que « Mettre à jour à la révision » change la fenêtre de temps d'une copie de travail pour regarder un point différent dans l'histoire, « Aller sur... » modifie la fenêtre d'espace d'une copie de travail pour qu'elle pointe vers un endroit différent du référentiel. C'est particulièrement utile quand vous travaillez sur le tronc et les branches où seuls quelques fichiers diffèrent. Vous pouvez alors commuter votre copie de travail entre les deux et seuls les fichiers modifiés seront transférés.
Annoter	Cette commande n'est utilisée que pour les fichiers texte et elle annote chaque ligne pour montrer la révision du référentiel à laquelle elle a été changée et l'auteur du changement. Notre implémentation graphique s'appelle TortoiseBlame et il montre aussi la date de dernière livraison et son commentaire au survol du numéro de révision avec la souris.
BDB	Berkeley DB. Une base de données bien testée pour les référentiels, qui ne peut pas être utilisée sur les partages réseaux. Par défaut pour les référentiels pré 1.2.
Branche	Un terme fréquemment utilisé dans les système de contrôle de révisions pour décrire ce qu'il se passe quand le développement se scinde à un point particulier et suit 2 chemins séparés. Vous pouvez créer une branche en dehors de la ligne de développement principale pour développer une nouvelle fonctionnalité sans rendre la ligne principale instable. Ou vous pouvez faire une branche avec une version stable sur laquelle vous ne ferez que des corrections de bugs, tandis que les nouveaux développements seront faits sur la branche instable. Dans Subversion, une branche est implémentée comme une « copie bon marché ».
Conflit	Quand les changements du référentiel sont fusionnés avec les changements locaux, ces changements se produisent parfois sur les mêmes lignes. Dans ce cas, Subversion ne peut pas décider automatiquement quelle version utilisée et le fichier est dit en conflit. Vous devez éditer le fichier manuellement et résoudre le conflit avant de pouvoir livrer d'autres modifications.
Copie de travail	C'est votre « bac à sable » local, l'endroit où vous travaillez sur les fichiers versionnés et il réside normalement sur votre disque dur local. Vous créez une copie de travail en faisant une « Extraction » du référentiel et vous remettez vos modifications dans le référentiel en faisant une « Livraison ».
Copier	Dans un référentiel Subversion, vous pouvez créer une copie d'un unique fichier ou de toute une arborescence. Celles-ci sont implémentées comme des « copies bon marché » qui fonctionnent comme des liens vers l'original dans le sens où elles ne prennent quasiment pas de place. Faire une copie préserve l'historique de l'élément dans la copie, donc vous pouvez suivre les changements effectués avant que la copie n'ait été faite.

Exporter	Cette commande crée une copie d'un répertoire versionné, comme une copie de travail, mais sans les répertoires locaux .svn.
Extraire	Une commande Subversion qui crée une copie de travail locale dans un répertoire vide en téléchargeant les fichiers versionnés depuis le référentiel.
FSFS	Un système de gestion de fichier de Subversion pour référentiels. Peut être utilisé pour les partages réseaux. Utilisé par défaut pour les référentiels à partir de la version 1.2.
Fusionner	<p>Le procédé par lequel les modifications du référentiel sont ajoutées dans votre copie de travail sans perturber les changements que vous avez déjà faits localement. Parfois ces changements ne peuvent pas être réconciliés automatiquement et la copie de travail est dite en conflit.</p> <p>La fusion se produit automatiquement lors de la mise à jour de votre copie de travail. Vous pouvez aussi fusionner des changements spécifiques depuis une autre branche en utilisant la commande Fusionner de TortoiseSVN.</p>
GPO	Group policy object (Stratégie de groupe)
Historique	Afficher l'historique des révisions d'un fichier ou d'un répertoire. Aussi appelé le « Journal ».
Importer	La commande Subversion pour importer une hiérarchie de dossiers complète dans le référentiel en une seule révision.
Journal	Afficher l'historique des révision d'un fichier ou d'un répertoire. Aussi connu comme l'« Historique ».
Livrer	Cette commande Subversion est utilisée pour renvoyer les changements de votre copie de travail locale au référentiel, en créant une nouvelle révision du référentiel.
Mettre à jour	Cette commande Subversion récupère les derniers changements depuis le référentiel vers votre copie de travail, en fusionnant les modifications faites par d'autres avec les modifications locales dans la copie de travail.
Nettoyer	Pour citer le manuel Subversion : « Nettoie récursivement la copie de travail, en supprimant les verrous et en reprenant les opérations non terminées. Si vous obtenez une erreur copie de travail verrouillée, exécutez cette commande pour supprimer les verrous périmés et rendre votre copie de travail utilisable à nouveau. » Notez que dans ce contexte, les « verrous » font référence aux verrouillages du système de fichiers local et non au verrouillage du référentiel.
Patch	Si la copie de travail n'a que des modifications sur des fichiers texte, il est possible d'utiliser la commande Subversion Voir les différences pour générer un unique fichier résumant ces changements dans le format de différences unifiées. Un fichier de ce type est souvent connu comme un « Patch » et il peut envoyé par email à quelqu'un d'autre (ou à une mailing list) et appliqué à une autre copie de travail. Une personne sans un accès pour livrer peut effectuer ces changements et soumettre un fichier patch à une personne autorisée à livrer pour qu'elle l'applique. Ou si vous n'êtes pas sûr d'un

---

	<p>changement, vous pouvez soumettre un patch aux autres pour qu'ils l'examinent.</p>
Propriété	<p>En plus de versionner vos répertoires et vos fichiers, Subversion vous permet d'ajouter des métadonnées versionnées - connues comme des « propriétés » - à chacun de vos fichiers et répertoires versionnés. Chaque propriété possède un nom et une valeur, plutôt qu'une clé de registre. Subversion dispose de quelques propriétés spéciales qu'il utilise en interne, comme <code>svn:eol-style</code>. TortoiseSVN en a aussi, tel que <code>tsvn:logminsize</code>. Vous pouvez ajouter vos propres propriétés avec des noms et des valeurs de votre choix.</p>
Propriété de révision (revprop)	<p>Comme les fichiers peuvent avoir des propriétés, chaque révision du référentiel le peut aussi. Quelques revprops spéciales sont ajoutées automatiquement quand la révision est créée, à savoir : <code>svn:date</code> <code>svn:author</code> <code>svn:log</code> qui représentent respectivement la date de livraison, le livreur et le commentaire. Ces propriétés peuvent être édités mais elles ne sont pas versionnées, donc les changements sont permanents et ne peuvent pas être annulés.</p>
Référentiel	<p>Un référentiel est un endroit central où sont stockées et entretenues les données. Un référentiel peut être un endroit où se trouvent plusieurs bases de données ou des fichiers pour la distribution sur un réseau, ou un référentiel peut être un emplacement directement accessible à l'utilisateur sans devoir traverser de réseaux.</p>
Relocaliser	<p>Si votre référentiel bouge, peut-être parce que vous l'avez déplacé dans un autre répertoire de votre serveur, ou le nom de domaine du serveur a changé, vous devez « relocaliser » votre copie de travail pour que ces URLs du référentiel pointent vers le nouvel emplacement.</p> <p>Note : vous ne devriez utiliser cette commande que si votre copie de travail fait référence au même emplacement dans le même référentiel, mais le référentiel a lui-même bougé. Dans d'autres circonstances, vous avez probablement besoin de la commande « Aller sur... » à la place.</p>
Résoudre	<p>Quand les fichiers d'une copie de travail sont laissés dans un état conflictuel suivant une fusion, ces conflits doivent être traités par une personne en utilisant un éditeur (ou peut-être TortoiseMerge). Ce procédé est connu comme « Résoudre les conflits ». Quand cela est fait, vous pouvez marquer les fichiers en conflit comme étant résolus, ce qui vous permet de les livrer.</p>
Revenir en arrière	<p>Subversion conserve une copie « primitive » locale de chaque fichier tel qu'il était lors de la dernière mise à jour de votre copie de travail. Si vous avez des changements et que vous décidez de les annuler, vous pouvez utiliser la commande « Revenir en arrière » pour revenir à la copie primitive.</p>
Révision	<p>Chaque fois que vous livrez un jeu de modifications, vous créez une nouvelle « révision » dans le référentiel. Chaque révision représente l'état de l'arborescence du référentiel à un certain point de son histoire. Si vous voulez revenir dans le temps, vous pouvez examiner le référentiel tel qu'il était à la révision N.</p> <p>Dans un autre sens, une révision peut faire référence à un jeu de modifications qui ont été faites quand la révision a été créée.</p>

---

Revision BASE	La révision de base courante d'un fichier ou d'un répertoire dans votre <i>copie de travail</i> . C'est la révision à laquelle se trouvait le fichier ou le répertoire, à la dernière extraction, mise à jour ou livraison. La révision BASE est normalement différente de la révision HEAD.
Révision HEAD	La dernière révision d'un fichier ou d'un répertoire dans le <i>référentiel</i> .
Supprimer	Quand vous supprimez un élément versionné (et que vous livrez le changement), l'élément n'existe plus dans le référentiel après la révision livrée. Mais il existe bien sûr toujours dans les révisions précédentes du référentiel, donc vous pouvez toujours y avoir accès. Si besoin, vous pouvez copier un élément supprimé et le « ressusciter » complètement avec son historique.
SVN	Une abbréviation pour Subversion fréquemment utilisée.  Le nom du protocole personnalisé de Subversion utilisé par le serveur de référentiel « svnserv ».
Verrouiller	Quand vous retirez un verrou sur un élément versionné, vous le marquez comme non livrable dans le référentiel, sauf dans la copie de travail où il a été déverrouillé.
Voir les différences	Très utile quand vous voulez voir exactement quels changements ont été faits.

---

# Index

## Symboles

'Copie de travail' non versionnée, 128

## A

Accès, 16  
actions côte serveur, 120  
Afficher les modifications, 52  
ajouter, 82  
Ajouter des fichiers au référentiel., 42  
aller sur, 103  
annoter, 117, 117  
annuler, 89, 177  
Annuler la livraison, 177  
Annuler les modifications, 177  
Apache, 26  
approuver, 117  
Authentification, 41  
auto-props, 95  
automatisation, 185, 188  
Autorisation, 30

## B

branche, 83, 101  
bug tracking, 130

## C

Changements de casse, 88  
Chemins UNC, 16  
click-droit, 37  
client en ligne de commande, 190  
Colonnes de l'explorateur, 59  
COM, 166, 171  
commentaire, 176  
commentaire de suivi de fusion, 72  
Commentaires de livraison, 64, 176  
compare des répertoires, 178  
comparer, 77  
comparer des fichiers, 178  
comparer des images, 80  
comparer des révisions, 79  
configuration, 135  
conflit, 8, 54  
Conflit dans l'arborescence, 54  
conflits de la fusion, 110  
contrôle de version, 1  
contrôleur de domaine, 31, 181  
copie, 101, 120  
copie de travail, 9  
copier des fichiers, 83  
Créer  
    Client de ligne de commande, 15  
    TortoiseSVN, 15  
Créer un référentiel, 15

Créer une copie de travail, 44

## D

déplacer, 87, 176  
déplacer des fichiers, 83  
deployer, 181  
Désactiver des fonctions, 182  
Détacher du référentiel, 180  
Développer les mot-clés, 93  
déversionner, 129, 180  
dictionnaire, 3  
différencier, 62  
Différencier, 77, 116  
Domaine Windows, 31

## E

éditer le journal/l'auteur, 73  
enlever, 86  
Entrées du menu contextuel, 182  
Envoyer les changements, 47  
Etat de la copie de travail, 58  
étiquette, 83, 101  
exclusion globale, 136  
expansion des jokers, 85  
explorateur, 1  
explorateur de référentiel, 120  
exportation, 128  
exporter les changements, 79  
externes, 98, 178  
extraction, 44  
Extraction, 47  
extraction de version, 166

## F

FAQ, 175  
fichiers spéciaux, 44  
fichiers temporaires, 42  
Fichiers/dossiers non versionnés, 84  
filtrer, 73  
fusionner, 104  
    Deux arborescences, 107  
    plage de révisions, 105  
    réintégration, 106

## G

gestionnaire d'incident, 130, 171  
gestionnaire de bugs, 130, 130  
glisser avec le bouton droit, 40  
glisser-déposer, 40, 40  
GPO, 181  
graphique, 123  
Graphique des révisions, 123  
greffon, 171

## H

historique, 64  
hooks, 19

hooks clients, 159

## I

IBugtraqProvider, 171

icônes, 58

ignorer, 84

ILC, 190

importer, 42

importer en place, 43

Index des projets, 30

installer, 3

intégration des fusions, 111

Interface COM SubWCRev, 168

## J

journal, 64

## L

L'URL a changé, 129

l'URL du référentiel a changé, 129

le serveur a été déplacé, 129, 129

lecteurs SUBST, 148

lecture seule, 112

lien, 19

lien d'extraction, 19

Lien TortoiseSVN, 19

ligne de commande, 185, 188

liste de changements, 62

liste de modifications, 178

livraison, 47

## M

Manuel de Subversion, 5

Marquer la release, 101

maximiser, 42

Mémoire Cache des messages de log, 155

menu contextuel, 37

message vide, 176

Messages de log, 64

mise à jour, 52, 177

modèle d'exclusion, 136

modifications, 60

mod\_authz\_svn, 27, 30

mot-clés, 93

msi, 181

## N

nettoyer, 91

NTLM, 31

Numéro de version dans les fichiers, 166

## O

outils de différenciation, 81

outils de fusion, 81

## P

packs de langue, 3

Partage réseau, 16

patch, 116

pattern matching, 85

Plusieurs authentications, 32

priorité des recouvrements, 196

projets ASP, 182

projets communs, 178

Projets du vendeur, 178

propriétés, 91

Propriétés de la révision, 73

Propriétés de TortoiseSVN, 96

Propriétés du projet, 96

Propriétés Subversion, 92

## R

raccourci, 180

recouvrement, 58, 196

référentiel, 5, 42

référentiels externes, 98

registre, 163

relocaliser, 129

renommer, 87, 120, 176

renommer des fichiers, 83

réorganiser, 176

répertoire .svn, 165

répertoire \_svn, 165

Résoudre, 54

retirer la mise sous contrôle de version, 180

revenir en arrière, 89, 177

révision, 12, 123

revprops, 73

## S

SASL, 24

sauvegarde, 18

Scripts de hook côté serveur, 19

scripts hook, 19, 159

serveur proxy, 149

Shell Windows, 1

Site internet, 19

sons, 135

SSL, 33

SSPI, 31

statistiques, 74

statut, 58, 60

stratégies de groupe, 181, 182

SubWCRev, 166

Suivi des fusions, 110

supprimer, 86

SVNParentPath, 28, 30

SVNPath, 28

svnserve, 20, 22

SVN\_ASP\_DOT\_NET\_HACK, 182

## T

TortoiseIDiff, 80

traductions, 3

## **V**

vérificateur d'orthographe, 3  
vérification de mise à niveau, 181  
Vérifier les mises à jour, 181  
verrouiller, 112  
version, 181  
Versionner les nouveaux fichiers, 82  
ViewVC, 134  
Visualiseur de différences unifiées, 116  
visualiseur de référentiel, 120, 134  
Voir les modifications, 58  
VS2003, 182  
Vue web, 134

## **W**

WebDAV, 26  
WebSVN, 134