

# **TortoiseSVN**

**Een Subversion client voor Windows**

**Version 1.14**

**Stefan Küng  
Lübbe Onken  
Simon Large**

---

# **TortoiseSVN: Een Subversion client voor Windows: Version 1.14**

door Stefan Küng, Lübbe Onken, en Simon Large

Vertaler: Jeroen van Amstel, Eddo Harkema, Ewout Muller, Christan Peper, Guus Raaphorst, Christophe de Wagter, Johannes Zwart

publicatie datum 2022/03/12 10:17:22 (r29368)

---

---

# Inhoudsopgave

Voorwoord .....	xi
1. Wat is TortoiseSVN? .....	xi
2. TortoiseSVN's mogelijkheden .....	xi
3. Licentie .....	xii
4. Ontwikkeling .....	xii
4.1. TortoiseSVN's Historie .....	xii
4.2. Dankwoord .....	xiii
5. Leesgids .....	xiii
6. Gebruikte terminologie in dit document .....	xiv
1. Aan de slag .....	1
1.1. Installeren van TortoiseSVN .....	1
1.1.1. Systeemvereisten .....	1
1.1.2. Installatie .....	1
1.2. Basis Beginselen .....	1
1.3. Ga voor een Proefrit .....	2
1.3.1. Aanmaken van een archief .....	2
1.3.2. Een Project importeren .....	2
1.3.3. Een lokale werkkopie (WC) ophalen .....	3
1.3.4. Wijzigingen Doorvoeren .....	4
1.3.5. Meer Bestanden Toevoegen .....	4
1.3.6. De project geschiedenis bekijken .....	5
1.3.7. Wijzigingen ongedaan maken .....	6
1.4. We gaan verder... .....	6
2. Basis Versie-Beheer Concepten .....	7
2.1. Het Archief .....	7
2.2. Versioneermodellen .....	7
2.2.1. Het Probleem van Bestand Delen .....	8
2.2.2. De Vergrendel-Wijzig-Ontgrendel Oplossing .....	8
2.2.3. De Kopieer-Wijzig-Samenvoeg Oplossing .....	9
2.2.4. Wat doet Subversion? .....	12
2.3. Subversion in Actie .....	12
2.3.1. Werkkopieën .....	12
2.3.2. Archief URLs .....	14
2.3.3. Revisies .....	14
2.3.4. Hoe Werkkopieën Het Archief Bijhouden .....	16
2.4. Samenvatting .....	16
3. Het Archief .....	18
3.1. Archief creatie .....	18
3.1.1. Een archief aanmaken via de opdrachtregel applicatie. ....	18
3.1.2. Het Archief Aanmaken Met TortoiseSVN .....	18
3.1.3. Lokale Toegang tot het Archief .....	19
3.1.4. Een archief benaderen op een Netwerk schijf .....	19
3.1.5. Archief indeling .....	19
3.2. Archief Reservekopie .....	21
3.3. Server gesitueerde hook scripts .....	21
3.4. Links voor Ophalen .....	22
3.5. Toegang hebben tot het Archief .....	22
4. Dagelijkse gebruikersgids .....	24
4.1. Algemene Features .....	24
4.1.1. Overlappende Pictogrammen .....	24
4.1.2. Contextmenus .....	24
4.1.3. Slepen en Plaatsen .....	26
4.1.4. Algemene snelkoppelingen .....	26
4.1.5. Authenticatie .....	27
4.1.6. Maximaliseren van Windows .....	28

---

4.2. Data Importeren In Een Archief .....	28
4.2.1. Importeren .....	28
4.2.2. Invoeren op zijn Plaats .....	30
4.2.3. Speciale Bestanden .....	30
4.3. Een lokale werkkopie (WC) ophalen .....	30
4.3.1. Ophaaldiepte .....	31
4.4. Je Wijzigingen Vastleggen In Het Archief .....	33
4.4.1. De Commit Dialoog .....	33
4.4.2. Werksets .....	36
4.4.3. Alleen delen van bestanden inchecken .....	36
4.4.4. Objecten van de vastleggingslijst uitsluiten .....	37
4.4.5. Vastleg Log Meldingen .....	37
4.4.6. Vastlegging voortgang .....	39
4.5. Je Werkkopie Bijwerken Met De Wijzigingen Van Anderen .....	40
4.6. Conflicten oplossen .....	42
4.6.1. Bestandsconflicten .....	42
4.6.2. Conflicten in eigenschappen .....	43
4.6.3. Mapstructuur conflicten .....	43
4.7. Statusinformatie ophalen .....	46
4.7.1. Overlappende Pictogrammen .....	46
4.7.2. Gedetailleerde status .....	48
4.7.3. Lokale en Server Status .....	49
4.7.4. Diffs bekijken .....	51
4.8. Werksets .....	51
4.9. Shelving .....	54
4.10. Logboekboodschappen Venster .....	56
4.10.1. Het Logboek Boodschappen Venster Opstarten .....	57
4.10.2. Revisie Logboek Acties .....	58
4.10.3. Extra Informatie Opvragen .....	59
4.10.4. Ophalen van uitgebreide logboek boodschappen .....	65
4.10.5. Huidige Werkkopie Revisie .....	66
4.10.6. Samenvoeg Traceer Functies .....	66
4.10.7. Logboekboodschap en Auteur Wijzigen .....	67
4.10.8. Filter op Logmeldingen .....	68
4.10.9. Statistische informatie .....	69
4.10.10. Offline Modus .....	73
4.10.11. De weergave verversen .....	73
4.11. Tonen van Verschillen .....	74
4.11.1. Bestandsverschillen .....	74
4.11.2. Opties voor regeleinden, spaties en witregels .....	75
4.11.3. Vergelijken van mappen .....	75
4.11.4. Vergelijk afbeeldingen met TortoiseIDiff .....	77
4.11.5. Diffing Office Documents .....	78
4.11.6. Externe Verschil/Samenvoeg gereedschappen .....	78
4.12. Nieuwe Bestanden En Mappen Toevoegen .....	79
4.13. Kopiëren/Verplaatsen/Hernoemen Bestanden en Mappen .....	80
4.14. Negeren bestanden en mappen .....	81
4.14.1. Patroonherkenning in negerlijsten .....	82
4.15. Verwijderen, Verplaatsen en Hernoemen .....	82
4.15.1. Verwijderen van bestanden en mappen .....	83
4.15.2. Verplaatsen van bestanden en mappen .....	84
4.15.3. Omgaan met conflicten in hoofdlettergebruik in bestandsnamen .....	84
4.15.4. Herstellen van Bestandhernoeringen .....	85
4.15.5. Verwijderen van Bestanden Zonder Versiebeheer .....	85
4.16. Wijzigingen ongedaan maken .....	85
4.17. Schonen .....	86
4.18. Project Instellingen .....	87
4.18.1. Subversion Eigenschappen .....	87

---

4.18.2. TortoiseSVN Project Eigenschappen .....	91
4.18.3. Eigenschappen bewerken .....	98
4.19. Externe Objecten .....	105
4.19.1. Externe Mappen .....	105
4.19.2. Externe Bestanden .....	108
4.19.3. Creating externals via drag and drop .....	108
4.20. Vertakken / labelen .....	108
4.20.1. Aanmaken van Tak of Label .....	108
4.20.2. Andere maniere om een tak of label aan te maken .....	111
4.20.3. Ophalen of Wisselen... ..	111
4.21. Samenvoegen .....	112
4.21.1. Een Revisie-Bereik Samenvoegen .....	113
4.21.2. Twee Verschillende Takken Samenvoegen .....	116
4.21.3. Samenvoegopties .....	117
4.21.4. De Resultaten van het Samenvoegen Bekijken .....	117
4.21.5. Samenvoegen traceren .....	118
4.21.6. Handling Conflicts after Merge .....	119
4.21.7. Onderhoud aan een Tak met Nieuwe Functies .....	122
4.22. Blokkeren .....	123
4.22.1. Hoe Het Blokkeren Werkt In Subversion .....	124
4.22.2. Blokkade verkrijgen .....	125
4.22.3. Een Blokkade Vrijgeven .....	125
4.22.4. Blokkadestatus Controleren .....	126
4.22.5. Niet-geblokkeerde bestanden als Alleen-lezen instellen .....	126
4.22.6. Hook Scripts Voor Blokkeringen .....	127
4.23. Een Patch Maken en Toepassen .....	127
4.23.1. Maken van een Patchbestand .....	127
4.23.2. Een Patchbestand Toepassen .....	129
4.24. Wie Heeft Welke Regel Gewijzigd? .....	129
4.24.1. Verklaar Bestanden .....	130
4.24.2. Verklaar Verschillen .....	132
4.25. De Archief-verkenner .....	132
4.26. Revisie schema's .....	135
4.26.1. Revisie Schema Knooppunten .....	136
4.26.2. Wijzigen van de Weergave .....	137
4.26.3. Gebruik van de grafiek .....	137
4.26.4. De weergave verversen .....	138
4.26.5. Boomstructuren inkorten .....	138
4.27. Exporteren van een Subversion Werkkopie .....	138
4.27.1. Versiebeheer van een werkkopie verwijderen .....	140
4.28. Verhuizen van een werkkopie .....	140
4.29. Integratie met Bug Tracking Systemen / Issue Trackers .....	141
4.29.1. Issue-nummers aan Logberichten toevoegen .....	142
4.29.2. Informatie Verkrijgen vanuit de Issue Tracker .....	145
4.30. Integratie met Web-gebaseerde Archief Viewers .....	146
4.31. TortoiseSVN Instellingen .....	147
4.31.1. Algemene instellingen .....	147
4.31.2. Revisie Grafiek Instellingen .....	158
4.31.3. Overlappende Pictogram Instellingen .....	160
4.31.4. Netwerk Instellingen .....	164
4.31.5. Externe Programma Instellingen .....	166
4.31.6. Bewaarde Gegevens Instellingen .....	171
4.31.7. Log Caching .....	172
4.31.8. Hook Scripts op de Client .....	175
4.31.9. TortoiseBlame Instellingen .....	181
4.31.10. TortoiseUDiff Settings .....	182
4.31.11. Exporting TSVN Settings .....	183
4.31.12. Geavanceerde Instellingen .....	183

---

4.32. Laatste stap .....	188
5. Project Monitor .....	189
5.1. Adding projects to monitor .....	189
5.2. Monitor dialog .....	190
5.2.1. Main operations .....	190
6. Het SubWCRev Programma .....	192
6.1. De SubWCRev Command Line .....	192
6.2. Trefwoordvervanging .....	194
6.3. Voorbeeld trefwoord .....	195
6.4. COM interface .....	197
7. IBugtraqProvider interface .....	200
7.1. Naming conventions .....	200
7.2. De IBugtraqProvider interface .....	200
7.3. De IBugtraqProvider2 interface .....	202
A. Veelgestelde Vragen (FAQ) .....	205
B. Hoe kan ik... .....	206
B.1. Verplaats een hoop bestanden in één keer .....	206
B.2. Forceer gebruikers een log-bericht in te typen .....	206
B.2.1. Hook-script op de server .....	206
B.2.2. Project eigenschappen .....	206
B.3. Update Geselecteerde Bestanden in het Archief .....	206
B.4. Draai Revisies Terug in het Archief (ongedaan maken) .....	207
B.4.1. Gebruik het logboek boodschappen scherm .....	207
B.4.2. Gebruik het samenvoeg scherm .....	207
B.4.3. Gebruik svndumpfilter .....	207
B.5. Vergelijk Twee Revisies van een Bestand of Map .....	208
B.6. Een veelgebruikt subproject gebruiken .....	208
B.6.1. svn:externals Gebruiken .....	208
B.6.2. Een geneste werkkopie gebruiken .....	208
B.6.3. Gebruik een relatieve lokatie .....	209
B.6.4. Add the project to the repository .....	209
B.7. Maak een snelkoppeling naar een archief .....	209
B.8. Negeer bestanden die al opgenomen zijn in het versiebeheer .....	209
B.9. Versiebeheer van een werkkopie verwijderen .....	210
B.10. Een werkkopie verwijderen .....	210
C. Handige tips voor beheerders .....	211
C.1. Verspreid TortoiseSVN via groepsbeleid .....	211
C.2. De update controle omleiden .....	211
C.3. Zetten van de SVN_ASP_DOT_NET_HACK omgevingsvariable .....	212
C.4. Opties in het contextmenu uitschakelen .....	212
D. Automatisering van TortoiseSVN .....	215
D.1. TortoiseSVN commando's .....	215
D.2. Tsvncmd URL verwerker .....	221
D.3. TortoiseIDiff Commando's .....	222
D.4. TortoiseUDiff Commands .....	223
E. Opdrachtprompt Interface Cross Reference .....	224
E.1. Conventies en Basis Regels .....	224
E.2. TortoiseSVN commando's .....	224
E.2.1. Ophalen .....	224
E.2.2. Verversen .....	224
E.2.3. Verversen naar Revisie .....	225
E.2.4. Vastleggen .....	225
E.2.5. Verschil .....	225
E.2.6. Toon logboek .....	226
E.2.7. Controleer op Wijzigingen .....	226
E.2.8. Revisie schema .....	226
E.2.9. Archief Verkenner .....	226
E.2.10. Bewerk conflicten .....	227

---

E.2.11. Opgelost .....	227
E.2.12. Hernoemen .....	227
E.2.13. Verwijderen .....	227
E.2.14. Ongedaan maken .....	227
E.2.15. Schonen .....	227
E.2.16. Plaats Blokkade .....	227
E.2.17. Blokkade Opheffen .....	228
E.2.18. Tak/Label .....	228
E.2.19. Optie .....	228
E.2.20. Samenvoegen .....	228
E.2.21. Exporteren .....	228
E.2.22. Locatie wijzigen .....	229
E.2.23. Maak Archief Hier .....	229
E.2.24. Toevoegen .....	229
E.2.25. Importeren .....	229
E.2.26. Verklaren .....	229
E.2.27. Voeg aan Negeerlijst toe .....	229
E.2.28. Maak patch .....	230
E.2.29. Patch toepassen .....	230
F. Implementatie details .....	231
F.1. Overlappende Pictogrammen .....	231
G. Taalpakketten en spellingscontrole .....	233
G.1. Taalpakketten .....	233
G.2. Spellingscontrole .....	233
Verklarende woordenlijst .....	235
Register .....	239

---

## Lijst van afbeeldingen

1.1. Het TortoiseSVN menu voor bestanden zonder versiebeheer .....	2
1.2. Het invoer venster .....	3
1.3. Bestandsverschillen Overzicht .....	4
1.4. Het logboekscherf .....	5
2.1. Een typisch Client/Server Systeem .....	7
2.2. Het probleem om te ontwijken .....	8
2.3. De Vergrendel-Wijzig-Ontgrendel Oplossing .....	9
2.4. De Kopieer-Wijzig-Samenvoeg Oplossing .....	10
2.5. ...Kopieren-Wijzigen-Samenvoegen Vervolg .....	11
2.6. Het archief bestandssysteem .....	13
2.7. Het Archief .....	15
3.1. Het TortoiseSVN menu voor bestanden zonder versiebeheer .....	18
4.1. Overlappende pictogrammen weergegeven in de verkenner .....	24
4.2. Kontekstmenu van een directory reeds onder versiebeheer .....	25
4.3. Verkenner bestandmenu van een snelkoppeling in een versiebeheerde map .....	26
4.4. Authenticatie dialoog .....	27
4.5. Het invoer venster .....	29
4.6. Het Ophalen venster .....	31
4.7. De Commit dialoog .....	34
4.8. De Vastleggen Dialoog Spellingscontrole .....	38
4.9. Het voortgangsvenster geeft de voortgang van een vastlegging weer .....	39
4.10. Voorgangsvenster van een afgeronde verversing .....	40
4.11. Overlappende pictogrammen weergegeven in de verkenner .....	46
4.12. Verkenner eigenschappen pagina, Subversion tab .....	48
4.13. Controleer op Wijzigingen .....	49
4.14. Vastleggen venster met werkset .....	53
4.15. Shelve dialog .....	55
4.16. Unshelve dialog .....	56
4.17. Het Logboek Boodschappen Venster .....	57
4.18. Het Context Menu Van Het Bovenste Gedeelte In Het Logboek Boodschappen Venster .....	59
4.19. The Code Collaborator Settings Dialog .....	62
4.20. Context menu van het bovenste deelvenster voor 2 geselecteerde revisies .....	62
4.21. Het context menu van het onderste deelvenster van het Logboekboodschappen venster .....	63
4.22. The Log Dialog Bottom Pane with Context Menu When Multiple Files Selected. ....	64
4.23. Het logboekboodschappen venster toont de revisies van samenvoegingen .....	67
4.24. Vastleggingen per Auteur Histogram .....	70
4.25. Vastleggingen-per-Auteur taartdiagram .....	71
4.26. Vastleggingen-per-datum Grafiek .....	72
4.27. Schakel naar Offline werken dialoog .....	73
4.28. De Vergelijk Revisies Dialoog .....	76
4.29. De beeld-verschil viewer .....	77
4.30. Verkenner context menu voor bestanden zonder versiebeheer .....	79
4.31. Rechts-slepen-menu voor een map met versiebeheer .....	80
4.32. Verkenner context menu voor bestanden zonder versiebeheer .....	81
4.33. Verkenner context menu voor bestanden met versiebeheer .....	83
4.34. Herstel dialoog .....	
4.35. The Cleanup dialog .....	86
4.36. Subversion kenmerken pagina .....	88
4.37. Toevoegen van eigenschappen .....	89
4.38. Property dialog for hook scripts .....	94
4.39. Property dialog boolean user types .....	94
4.40. Property dialog state user types .....	95
4.41. Property dialog single-line user types .....	96
4.42. Property dialog multi-line user types .....	97
4.43. svn:externals eigenschappen pagina .....	99



---

4.44. svn:keywords eigenschappen pagina .....	100
4.45. svn:eol-style eigenschappen pagina .....	100
4.46. tsvn:bugtraq eigenschappen pagina .....	101
4.47. Logberichtgrootte eigenschappen pagina .....	102
4.48. Taal eigenschappen pagina .....	102
4.49. svn:mime-type eigenschappen pagina .....	103
4.50. svn:needs-lock eigenschappen pagina .....	103
4.51. svn:executable eigenschappen pagina .....	103
4.52. Property dialog merge log message templates .....	104
4.53. De Tak/Label Dialoog .....	109
4.54. De Wissel Dialoog .....	112
4.55. Het Samenvoeg Hulpprogramma - Selecteer Revisie Bereik .....	114
4.56. Het Samenvoeg Hulpprogramma - Samenvoegen van Boomstructuren .....	116
4.57. The Merge Conflict Dialog .....	120
4.58. The Merge Tree Conflict Dialog .....	122
4.59. The Merge-All Dialog .....	123
4.60. Het blokkeringsscherf .....	125
4.61. Het Kijk of er updates zijn scherm .....	126
4.62. Het Maak Patch Venster .....	128
4.63. Het Verklar/Annoteren Scherm .....	130
4.64. TortoiseBlame .....	131
4.65. De Archief-verkenner .....	133
4.66. Een Revisie Schema .....	135
4.67. Het Exporteren-vanaf-een-URL scherm .....	139
4.68. De Verhuis Dialoog .....	140
4.69. The Bugtraq Properties Dialog .....	142
4.70. Voorbeeld issue tracker zoek scherm .....	146
4.71. Het Instellingen Scherm, Algemene Pagina .....	148
4.72. Het Instellingen Scherm, Context Menu Pagina .....	150
4.73. Het Instellingen Scherm, Schermen 1 Pagina .....	151
4.74. Het Instellingen Scherm, Schermen 2 Pagina .....	153
4.75. The Settings Dialog, Dialogs 3 Page .....	155
4.76. Het Instellingen Scherm, Kleuren Pagina .....	156
4.77. Het instellingen scherm, Revisie schema pagina .....	158
4.78. Het instellingen scherm, Revisieschema kleuren pagina .....	159
4.79. Het instellingen scherm, Overlappend pictogram pagina .....	160
4.80. Het instellingen scherm, Pictogramset pagina .....	163
4.81. The Settings Dialog, Icon Handlers Page .....	164
4.82. Het instellingen scherm, Netwerk pagina .....	165
4.83. Het instellingen scherm, Diff viewer pagina .....	166
4.84. Het instellingen scherm, Verschil/Samenvoeg uitgebreid scherm .....	170
4.85. Het instellingen scherm, Opgeslagen gegevens pagina .....	171
4.86. Het Instellingen Scherm, Log Cache Pagina .....	172
4.87. Het Instellingen Scherm, Log Cache Statistieken .....	174
4.88. Het Instellingen Scherm, Hook Scripts Pagina .....	176
4.89. Het Instellingen Scherm, Hook Scripts Configureren .....	177
4.90. Het Instellingen Scherm, Issue Tracker Integratie Pagina .....	180
4.91. Het Instellingen Scherm, TortoiseBlame pagina .....	181
4.92. The Settings Dialog, TortoiseUDiff Page .....	182
4.93. The Settings Dialog, Sync Page .....	183
4.94. Taskbar with default grouping .....	185
4.95. Taskbar with repository grouping .....	185
4.96. Taskbar with repository grouping .....	186
4.97. Taskbar grouping with repository color overlays .....	186
5.1. The edit project dialog of the project monitor .....	189
5.2. The main dialog of the project monitor .....	190
B.1. The TortoiseSVN right drag context menu for moving files .....	206
C.1. De commit-dialoog terwijl deze de upgrade-notificatie weergeeft .....	211

---

## Lijst van tabellen

2.1. Archief Toegang URLs .....	14
4.1. Pinned Revision .....	111
6.1. Lijst met beschikbare opdrachtregel opties .....	193
6.2. List of SubWCRev error codes .....	193
6.3. List of available keywords .....	194
6.4. COM/automatiseringsmethoden worden ondersteund .....	197
C.1. Menuopties en hun waarden .....	212
D.1. Lijst van alle beschikbare commando's en opties .....	216
D.2. Lijst van beschikbare opties. ....	222
D.3. Lijst van beschikbare opties. ....	223

---

# Voorwoord



# TortoiseSVN

Versie beheer is de kunst van het beheren van wijzigingen in informatie. Als sinds lange tijd is dit een cruciaal gereedschap voor programmeurs, die meestal kleine wijzigingen in de software maken en dan de volgende dag deze wijzigingen ongedaan maken of controleren. Stel je een team voor met zulke ontwikkelaars die tegelijkertijd werken - en misschien zelfs wel simultaan aan dezelfde bestanden! - en je kunt je voorstellen waarom een goed systeem nodig is om *de mogelijke chaos te beheersen*.

## 1. Wat is TortoiseSVN?

TortoiseSVN is a free open-source Windows client for the *Apache™ 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”.

Sommige versie beheer systemen zijn ook software configuratie management (SCM) systemen. Deze systemen zijn speciaal gemaakt om source code bomen te beheren en hebben vele kenmerken specifiek voor software ontwikkeling - zoals het automatisch begrijpen van programmeertalen, of het voorzien in gereedschappen voor het bouwen van software. Subversion echter, is niet een van deze systemen; het is een generiek systeem, dat gebruikt kan worden voor het beheren van *elke* verzameling bestanden, inclusief source code.

## 2. TortoiseSVN's mogelijkheden

Wat maakt TortoiseSVN zo'n goede Subversion client? Zie hier een korte lijst van eigenschappen.

### Shell integratie

TortoiseSVN integreert naadloos in de Windows shell (i.e. de Verkenner). Dit betekent dat je kan blijven werken met de gereedschappen waar je al bekend mee bent. En je hoeft niet steeds een andere applicatie op te starten als je de versie beheer functionaliteit nodig hebt!

En daarbij komt nog dat je de Windows Verkenner niet per sé hoeft te gebruiken. TortoiseSVN's context menus werken in vele andere bestandbeheer programma's, alsook in de Bestand/Open dialoog, welke in de meeste standaard Windows applicaties aanwezig is. Je moet hierbij wel bedenken dat TortoiseSVN in eerste instantie ontwikkeld is als uitbreiding op de Windows Verkenner. Daardoor is het mogelijk dat in andere applicaties de integratie niet zo compleet is en bijvoorbeeld de overlappende pictogrammen niet zichtbaar zijn.

### Overlappende pictogrammen

De status van elk bestand en folder onder versiebeheer wordt weergegeven met een klein overlappend pictogram. Op die manier kun je direct zien wat de status is van je werkkopie.

### Grafische gebruikersinterface

Wanneer je de wijzigingen aan bestanden of mappen in een lijst laat zien, kun je op een revisie klikken om het commentaar voor die vastlegging te laten zien. Je kan ook een lijst met gewijzigde bestanden te zien krijgen - je hoeft alleen maar op een bestand te dubbel-klikken om precies te zien wat er veranderd is.

De vastleggen dialoog toont een lijst met alle onderdelen die toegevoegd zullen worden aan een vastlegging en elk # item heeft een selectievakje waarmee je kunt kiezen welke je wilt toevoegen. Voor het geval je een nieuw bestand vergeten bent toe te voegen, kunnen bestanden zonder versiebeheer ook getoond worden.

#### Gemakkelijke toegang tot Subversion commando's

Alle Subversion commando's zijn beschikbaar via het context menu van de Verkenner. TortoiseSVN voegt zijn eigen submenu hierin.

Aangezien TortoiseSVN een Subversion client is, willen we je ook een aantal kenmerken van Subversion zelf laten zien.

#### Map versiebeheer

CVS houdt alleen de geschiedenis van individuele bestanden bij, maar Subversion implementeert een “virtueel” geversioneerde filesysteem, dat wijzigingen in de gehele folderstructuur bijhoudt. Bestanden en mappen staan beide onder versiebeheer. Het resultaat is dat er echte commando's zijn om bestanden en mappen te **verplaatsen** en te **kopiëren**.

#### Atomische commits

Een enkele vastleggen-actie gaat of helemaal, of helemaal niet in het archief. Hierdoor kunnen ontwikkelaars wijzigingen maken en vastleggen als logische brokken.

#### Geversioneerde metadata

Elk bestand en elke map hebben een onzichtbare set van “properties”. Je kunt elk willekeurig sleutel/waarde paar bedenken dat maar je wilt, en opslaan. Eigenschappen worden net als bestandinhoud in de tijd geversioneerd.

#### Keuze van netwerkklagen

Subversion heeft een abstracte notie wat betreft toegang tot het archief, wat het gemakkelijk maakt voor mensen om een nieuw netwerkmechanisme te implementeren. Subversions “geavanceerde” netwerk server is een module voor een Apache webserver, die een afgeleide van HTTP genaamd WebDAV/DeltaV spreekt. Dit biedt Subversion een groot voordeel als het gaat om stabiliteit en interoperabiliteit, en voorziet in verschillende belangrijke eigenschappen, die ook nog eens gratis zijn: zoals onder andere authenticatie, autorisatie, netwerkdatacompressie, en bladeren in het archief. Een kleinere op zichzelfstaand Subversion serverproces is ook beschikbaar. Deze server gebruikt een eigen netwerkprotocol, die gemakkelijk over ssh getunneld kan worden.

#### Consistente data-afhandeling

Subversion drukt bestandsverschillen uit door middel van een binair verschilalgoritme, dat identiek werkt voor tekstbestanden (mens-leesbaar) en binaire bestanden (mens-onleesbaar). Beide typen bestanden worden op dezelfde gecomprimeerde manier in het archief opgeslagen en verschillen worden over het netwerk in beide richtingen gezonden.

#### Efficiënt aftakken en labelen

De kosten van aftakken en labelen hoeft niet proportioneel met de projectgrootte te zijn. Subversion maakt branches en tags door simpelweg het project te kopiëren, op een manier die lijkt op een hard-link. Op deze manier nemen deze operaties een zeer kleine, constante tijd in beslag en erg weinig ruimte in de repository.

## 3. Licentie

TortoiseSVN is een Open Source project ontwikkeld onder de GNU General Public License (GPL). Het is gratis te downloaden en gebruiken, zowel persoonlijk als commercieel, op een ongelimiteerd aantal PC's.

Although most people just download the installer, you also have full read access to the source code of this program. You can browse it on this link <https://osdn.net/projects/tortoisesvn/scm/svn/>. The current development line is located under `/trunk/`, and the released versions are located under `/tags/`.

## 4. Ontwikkeling

Zowel TortoiseSVN als Subversion zijn ontwikkeld door een groep mensen, die aan deze projecten werkt. Ze komen uit verschillende landen van de hele wereld en ontwikkelen samen deze mooie programma's.

### 4.1. TortoiseSVN's Historie

Tim Kemp vond in 2002 dat Subversion een zeer goed versie beheer systeem was, maar dat het tekort schoot op gebied van de GUI. Het idee om een Subversion client met Windows shell te integreren, kwam voort uit

de soortgelijke integratie voor CVS genaamd TortoiseCVS. Tim bestudeerde de source code van TortoiseCVS en gebruikte deze als basis voor TortoiseSVN. Hij startte vervolgens het project, registreerde het domein `tortoisesvn.org` en plaatste de source code online.

Gedurende diezelfde tijd zocht Stefan Küng een goed en gratis versie beheer systeem en vond zowel Subversion als de source code van TortoiseSVN. Aangezien TortoiseSVN nog niet klaar voor gebruik was, sloot hij zich bij het project aan en begon te programmeren. Al snel herschreef hij het meeste van de bestaande code en begon met het toevoegen van commando's en eigenschappen tot aan het punt waar niets van de originele code nog bestond.

Hoe stabiel Subversion werd, hoe meer gebruikers het aantrok; gebruikers die ook TortoiseSVN als Subversion client gingen gebruiken. De gebruikersgroep groeide snel en groeit nog steeds iedere dag. Op dat moment bood Lübke Onken aan om te helpen met leuke icons en een logo voor TortoiseSVN. Hij zorgt ook voor de website en beheert de vertalingen.

With time, other version control systems all got their own Tortoise client which caused a problem with the icon overlays in Explorer: the number of such overlays is limited and even one Tortoise client can easily exceed that limit. That's when Stefan Küng implemented the TortoiseOverlays component which allows all Tortoise clients to use the same icon overlays. Now all open source Tortoise clients and even some non-Tortoise clients use that shared component.

## 4.2. Dankwoord

Tim Kemp

voor het opstarten van het TortoiseSVN project

Stefan Küng

voor het harde werk om TortoiseSVN te maken zoals het nu is en zijn leiderschap over het project

Lübke Onken

voor de mooie icons, het logo, het zoeken naar defecten, het vertalen en het beheer van de vertalingen

Simon Large

voor het onderhouden van de documentatie

Stefan Fuhrmann

voor de log cache en de revisie grafiek

Het Subversion Boek

voor de geweldige introductie in Subversion en diens hoofdstuk 2 dat we hier gekopieerd hebben

Het Tigris Stijl project

voor sommige stijlen die hergebruikt zijn in dit document

Onze medewerkers

voor de patches, defect rapportages en nieuwe ideeën, en voor het helpen van anderen door vragen in onze mailinglist te beantwoorden

Onze donateurs

voor de vele uren plezier met de muziek die zij ons stuurden

## 5. Leesgids

Dit boek is geschreven voor computer intellectuelen, die Subversion willen gebruiken om hun data te beheren, maar niet graag opdracht-prompts gebruiken. Aangezien TortoiseSVN een windows shell-extensie is, wordt verondersteld dat de gebruiker bekend is met de windows verkenner en weet hoe deze te gebruiken.

Dit **Voorwoord** legt uit wat TortoiseSVN is, geeft een beknopte toelichting over het TortoiseSVN project en de gemeenschap die hieraan werkt, alsmede de licentievoorwaarden voor gebruik en verspreiding.

**Hoofdstuk 1, *Aan de slag*** legt uit hoe je TortoiseSVN op je PC installeert en hoe je het daarna direct kan gaan gebruiken.

In **Hoofdstuk 2, *Basis Versie-Beheer Concepten*** geven we een korte introductie van het *Subversion* revisie beheer systeem, dat ten grondslag ligt aan TortoiseSVN. Dit is geleend van de documentatie van het Subversion project en licht de verschillende benaderingen van versie beheer toe, alsook hoe Subversion werkt.

The chapter on **Hoofdstuk 3, *Het Archief*** explains how to set up a local repository, which is useful for testing Subversion and TortoiseSVN using a single PC. It also explains a bit about repository administration which is also relevant to repositories located on a server.

**Hoofdstuk 4, *Dagelijkse gebruikersgids*** is de belangrijkste sectie, omdat hierin de voornaamste functies van TortoiseSVN alsmede hoe ze te gebruiken, worden uitgelegd. Het is in de vorm van een handleiding, die begint met het ophalen van een werkkopie, het wijzigen ervan, het vastleggen van je wijzigingen, enz. Het vervolgt dan met meer geavanceerde thema's.

De **Hoofdstuk 5, *Project Monitor*** laat zien hoe je je Subversion projecten kan monitoren. Hierdoor mis je geen belangrijke commits van andere team leden.

**Hoofdstuk 6, *Het SubWCRev Programma*** is een afzonderlijk programma samengevoegd met TortoiseSVN, welke informatie uit je werkkopie kan halen en in een bestand kan schrijven. Het is bruikbaar om bouw informatie in je project te voegen.

De **Bijlage B, *Hoe kan ik...*** sectie beantwoordt algemene vragen over het uitvoeren van taken, die niet expliciet op andere plekken worden behandeld.

De **Bijlage D, *Automatisering van TortoiseSVN*** toont hoe de TortoiseSVN GUI dialogen vanaf de opdracht prompt kunnen worden aangeroepen. Dit is bruikbaar bij scripten waarbij wel gebruikersinteractie nodig is.

De **Bijlage E, *Opdracht prompt Interface Cross Reference*** correleren de TortoiseSVN commando's met hun equivalenten in de Subversion commando regel `svn.exe`.

## 6. Gebruikte terminologie in dit document

Om het lezen van de documentatie te vergemakkelijken worden de namen van de schermen en menu's van TortoiseSVN opgemaakt in een ander font. Bijvoorbeeld de Log Dialog.

Een menu keuze wordt weergegeven door een pijl. TortoiseSVN → Show Log betekent: selecteer *Show Log* van het *TortoiseSVN* context menu.

Als een lokaal context menu in een van de TortoiseSVN dialogs verschijnt, wordt het als volgt weergegeven: Context Menu → Save As ...

Gebruikers interface knoppen worden als volgt weergegeven: Druk OK om door te gaan.

Gebruikersacties worden met een vetgedrukt lettertype weergegeven. **Alt+A**: druk op de **Alt**-toets op je toetsenbord en terwijl je deze ingedrukt houdt, druk je de **A**-toets ook in. Rechts-slepen: druk de rechtermuis knop in en terwijl je deze ingedrukt houdt, *sleep* je de items naar de nieuwe lokatie.

Systeemuitvoer en toetsenbordinput wordt eveneens weergegeven met een ander font.



### Belangrijk

Belangrijke opmerkingen zijn gemarkeerd met een pictogram.



### Tip

Tips die je leven gemakkelijker maken



---

## **Let op**

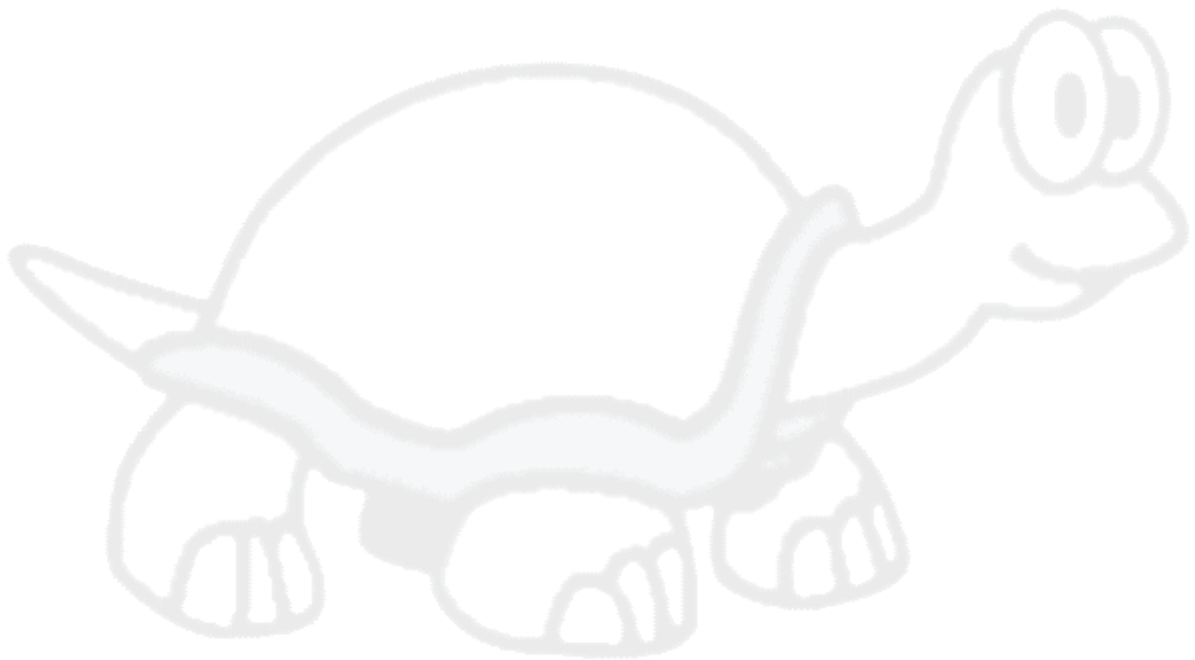
Locaties waar je voorzichtig moet zijn bij wat je doet



---

## **Waarschuwing**

Hier moet extreme voorzorg genomen worden. Er is gevaar voor datacorruptie of andere vervelende dingen als deze waarschuwingen worden genegeerd.



---

# Hoofdstuk 1. Aan de slag

Deze sectie is bedoeld voor mensen die willen weten waar TortoiseSVN over gaat en het eens uit willen proberen. Er wordt uitgelegd hoe je TortoiseSVN kunt installeren, een lokaal archief moet aanmaken en helpt je met de meest gebruikte acties.

## 1.1. Installeren van TortoiseSVN

### 1.1.1. Systeemvereisten

TortoiseSVN runs on Windows Vista or higher and is available in both 32-bit and 64-bit flavours. The installer for 64-bit Windows also includes the 32-bit extension parts. Which means you don't need to install the 32-bit version separately to get the TortoiseSVN context menu and overlays in 32-bit applications.

Ondersteuning voor Windows 98, Windows ME en Windows NT4 is weggehaald in versie 1.2.0, Windows 2000 en XP t/m SP2 ondersteuning is weggehaald in 1.7.0. Ondersteuning voor Windows XP met SP3 is weggehaald in 1.9.0. Je kunt nog steeds oudere versies downloaden installeren indien je deze nodig hebt.

### 1.1.2. Installatie

TortoiseSVN wordt geleverd met een gemakkelijk te gebruiken installatieprogramma. Dubbelklik op het installatiebestand en volg de instructies op. Het installatieprogramma zorgt voor de rest. Denk er aan dat na installatie een herstart van de computer noodzakelijk is.



#### Belangrijk

Je hebt Beheerdersrechten nodig om TortoiseSVN te mogen installeren. Het installatie programma zal om deze rechten vragen indien nodig.

Taalpakketten zijn beschikbaar om de gebruikers interface in veel verschillende talen in te stellen. Bekijk [Bijlage G, Taalpakketten en spellingscontrole](#) voor meer informatie.

If you encounter any problems during or after installing TortoiseSVN please refer to our online FAQ at <https://tortoisesvn.net/faq.html>.

## 1.2. Basis Beginselen

Voordat we aan de slag gaan met het werken met echte bestanden, is het goed om een overzicht te hebben over hoe Subversion werkt en welke terminologie gebruikt wordt.

### Het Archief

Subversion gebruikt een centrale database waarin al jouw bestanden met versiebeheer en de complete geschiedenis worden opgeslagen. Deze database wordt aangeduid met het *archief*. Het archief bevindt zich normaliter op een bestandserver met de Subversion server applicatie. Deze applicaties verstuurt op aanvraag de inhoud aan Subversion gebruikersapplicaties (zoals TortoiseSVN). Als je maar één ding archiveert, archiveer dan je archief, omdat dat je bron kopie is van al je data.

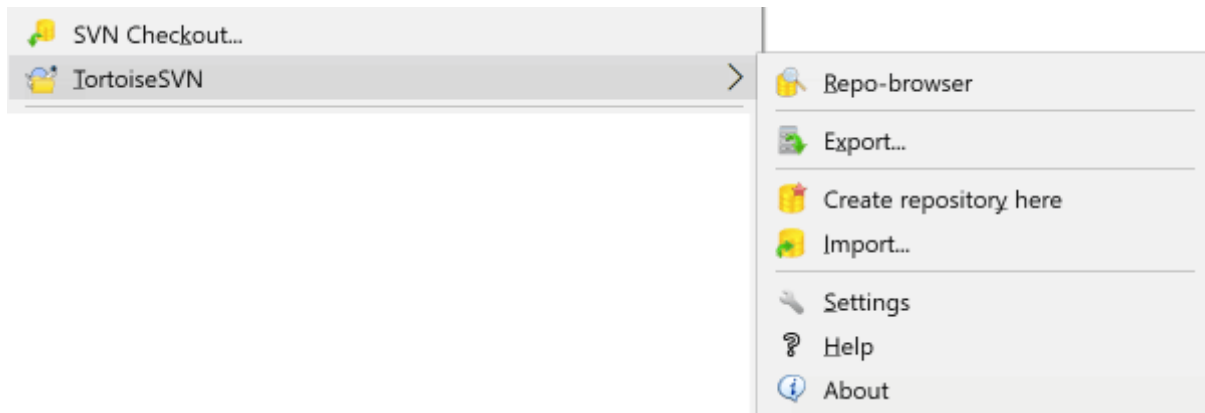
### Werkkopie

Dit is waar je het echte werk uitvoert. Iedere ontwikkelaar heeft zijn eigen werkkopie, soms ook wel bekend als sandbox, op zijn lokale PC. Je kunt de laatste versie van de repository ophalen, deze lokaal bewerken zonder iemand anders te dwarsbomen, en vervolgens jouw wijzigingen (wanneer deze volgens jou goed zijn) terugsturen naar de repository.

Een Subversion werkkopie bevat niet de gehele historie van het project, maar het bewaart wel een kopie van de bestanden (in de staat die ze waren, voordat je ze bewerkte). Dit betekent dat je eenvoudig kan inzien welke wijzigingen je precies hebt gemaakt.



Je moet ook weten waar je TortoiseSVN kunt vinden, want in het Windows Start menu is niet veel te zien. Dit komt doordat TortoiseSVN een shell-uitbreiding is. Start dus eerst de verkenner. Als je dan met de rechter muisknop op een map in de verkenner klikt zul je een paar nieuwe onderdelen in het context menu zien staan:



**Afbeelding 1.1. Het TortoiseSVN menu voor bestanden zonder versiebeheer**

## 1.3. Ga voor een Proefrit

Deze sectie laten we je zien hoe je een paar van de meest gebruikte functies kunt proberen op een kleine test repository. Natuurlijk verklaart het niet alles - dit is immers slechts de snelstart gids. Als je alles geconfigureerd hebt moet je de tijd nemen om de rest van deze handleiding door te lezen. Daarin wordt ook uitgelegd hoe je een echte Subversion server installeert.

### 1.3.1. Aanmaken van een archief

Voor een echt project zul je een repository op een veilige plaats opzetten en een Subversion server om het te controleren. In deze tutorial gebruiken we Subversion's lokale repository mogelijkheid welke directe toegang tot een repository op je hard disk mogelijk maakt, zonder een aparte server.

First create a new empty directory on your PC. It can go anywhere, but in this tutorial we are going to call it C:\svn\_repos. Now right click on the new folder and from the context menu choose TortoiseSVN → Create Repository here.... The repository is then created inside the folder, ready for you to use. We will also create the default internal folder structure by clicking the Create folder structure button.

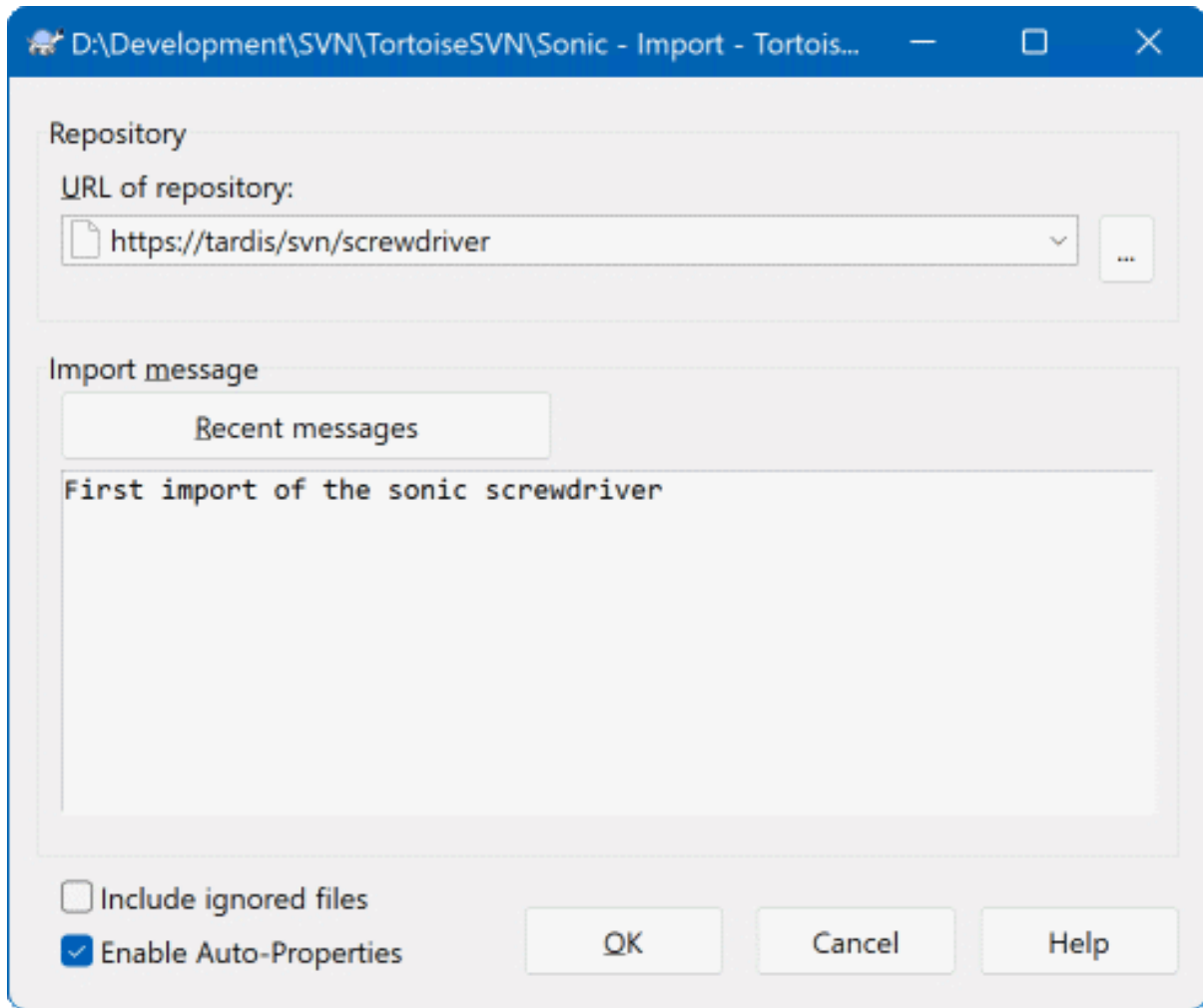


#### Belangrijk

De lokale repository mogelijkheid is erg handig voor test en evaluatie maar tenzij je als enige ontwikkelaar op een enkele computer werkt kun je beter een echte subversion server gebruiken. Het is verleidelijk om in een klein bedrijf het extra werk om een server te installeren te vermijden en de repository op een via het netwerk gedeelde map te benaderen. Doe dat nooit. Je zult data verliezen. Lees [Paragraaf 3.1.4, "Een archief benaderen op een Netwerk schijf"](#) om te weten waarom dit een slecht idee is en hoe je een server moet installeren.

### 1.3.2. Een Project importeren

Now we have a repository, but it is completely empty at the moment. Let's assume I have a set of files in C:\Projects\Widget1 that I would like to add. Navigate to the Widget1 folder in Explorer and right click on it. Now select TortoiseSVN → Import... which brings up a dialog



**Afbeelding 1.2. Het invoer venster**

A Subversion repository is referred to by URL, which allows us to specify a repository anywhere on the Internet. In this case we need to point to our own local repository which has a URL of `file:///c:/svn_repos/trunk`, and to which we add our own project name `Widget1`. Note that there are 3 slashes after `file:` and that forward slashes are used throughout.

Een andere belangrijke functie van dit dialoogvenster is de **Import Message** box. Hierin kun je beschrijven wat je veranderd hebt. Als je later in de project geschiedenis zoekt, zijn deze berichten een waardevolle leidraad voor welke wijzigingen zijn aangebracht en waarom. In dit geval kun je bijvoorbeeld iets simpels als "Importeer het widget1 project" invullen. Klik op **OK** en de map wordt toegevoegd aan uw repository.

### 1.3.3. Een lokale werkkopie (WC) ophalen

Now that we have a project in our repository, we need to create a working copy to use for day-to-day work. Note that the act of importing a folder does not automatically turn that folder into a working copy. The Subversion term for creating a fresh working copy is **Checkout**. We are going to checkout the `Widget1` folder of our repository into a development folder on the PC called `C:\Projects\Widget1-Dev`. Create that folder, then right click on it and select **TortoiseSVN → Checkout...** Then enter the URL to checkout, in this case `file:///c:/svn_repos/trunk/Widget1` and click on **OK**. Our development folder is then populated with files from the repository.



## Belangrijk

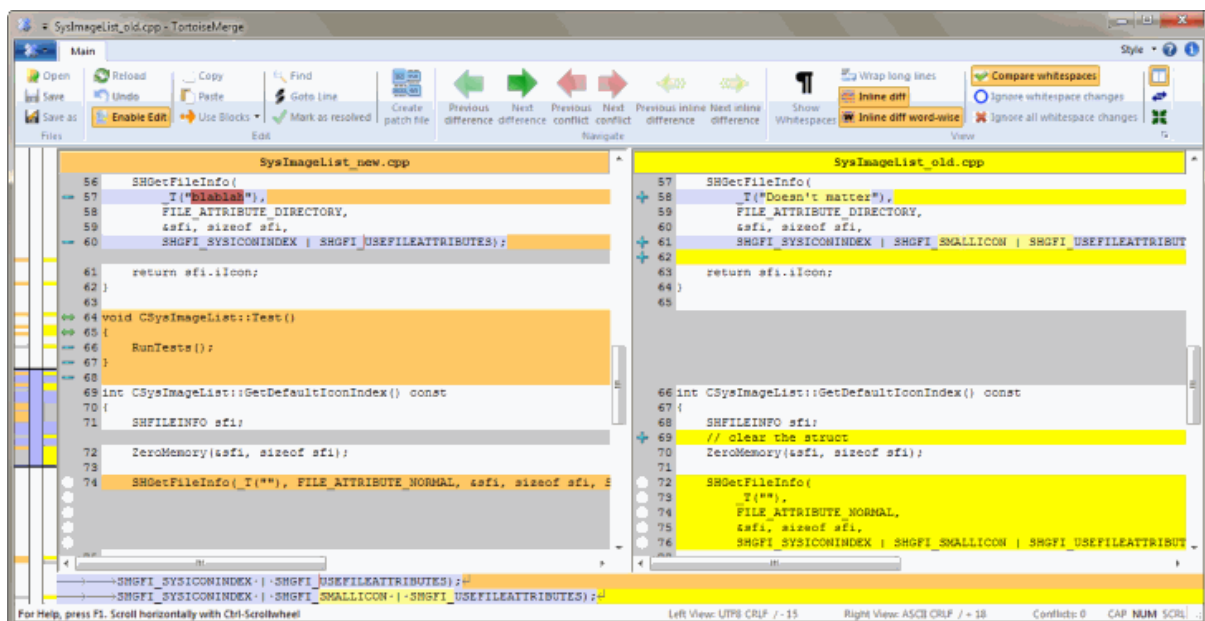
In the default setting, the checkout menu item is not located in the TortoiseSVN submenu but is shown at the top explorer menu. TortoiseSVN commands that are not in the submenu have SVN prepended: SVN Checkout...

U zal merken dat het uitzicht van deze bestandsmap verschilt van de oorspronkelijke map. Elk bestand heeft een groen vinkje in de hoek linksonder. Deze TortoiseSVN-statusicoontjes zijn enkel zichtbaar in een map die onder versiecontrole geplaatst is. Het groene statussymbool betekent dat het bestand in de map onveranderd is van de versie in het SVN-repository.

### 1.3.4. Wijzigingen Doorvoeren

Time to get to work. In the Widget1-Dev folder we start editing files - let's say we make changes to Widget1.c and ReadMe.txt. Notice that the icon overlays on these files have now changed to red, indicating that changes have been made locally.

Maar wat zijn nu de veranderingen? Rechts-klik op een van de veranderde bestanden en selecteer TortoiseSVN → Diff. Het vergelijkprogramma van TortoiseSVN wordt gestart en laat precies zien welke regels veranderd zijn.



Afbeelding 1.3. Bestandsverschillen Overzicht

OK, so we are happy with the changes, let's update the repository. This action is referred to as a Commit of the changes. Right click on the Widget1-Dev folder and select TortoiseSVN → Commit. The commit dialog lists the changed files, each with a checkbox. You might want to choose only a subset of those files, but in this case we are going to commit the changes to both files. Enter up a message to describe what the change is all about and click on OK. The progress dialog shows the files being uploaded to the repository and you're done.

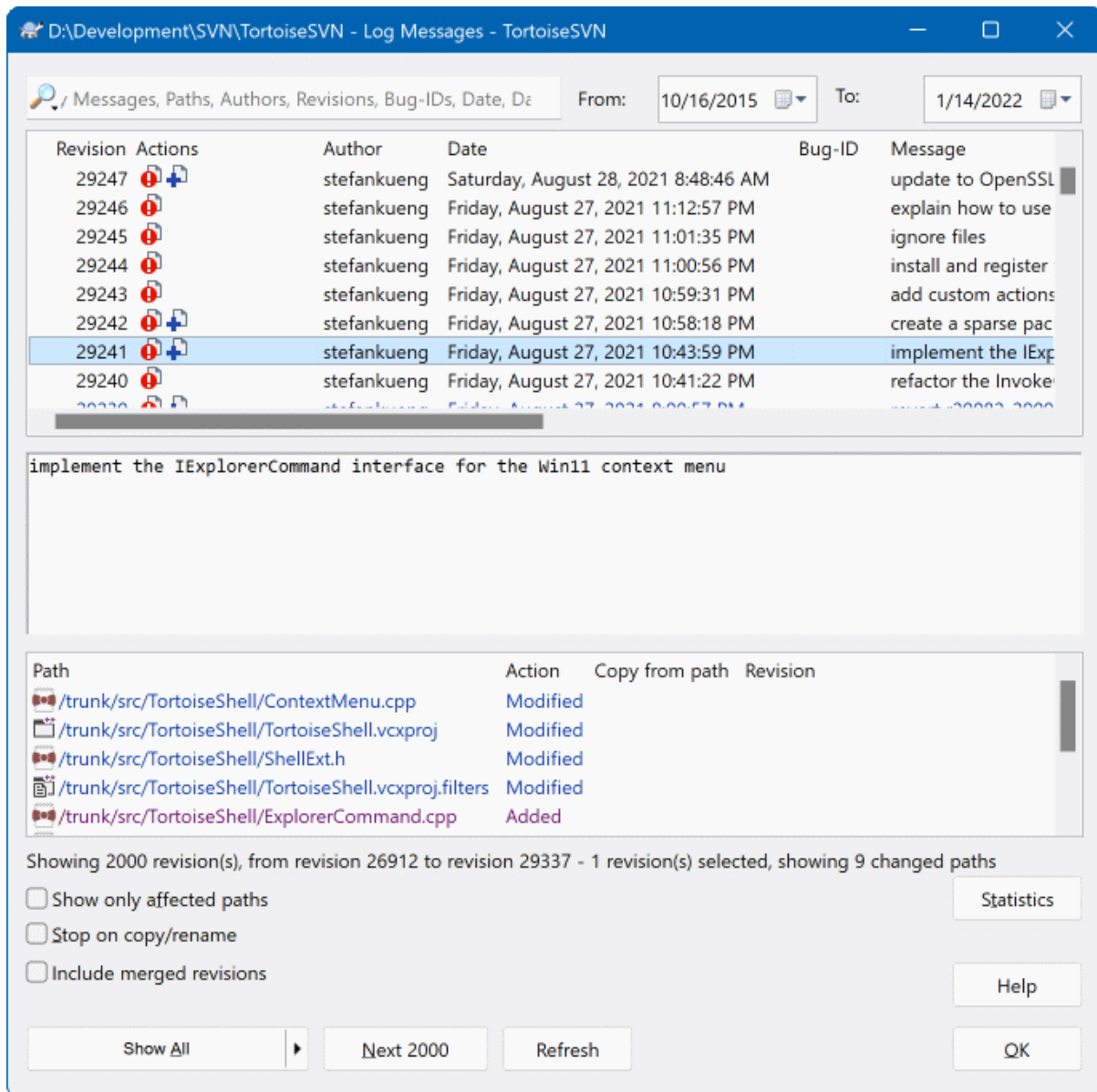
### 1.3.5. Meer Bestanden Toevoegen

As the project develops you will need to add new files - let's say you add some new features in Extras.c and add a reference in the existing Makefile. Right click on the folder and TortoiseSVN → Add. The Add dialog now shows you all unversioned files and you can select which ones you want to add. Another way of adding files would be to right click on the file itself and select TortoiseSVN → Add.

Wanneer je vervolgens de folder wilt vastleggen wordt het nieuwe bestand getoond als *Toegevoegd* en het bestaande bestand als *Aangepast*. Bemerkt dat je door te dubbelklikken op het aangepaste bestand de exacte wijzigingen aan dat bestand kan zien.

### 1.3.6. De project geschiedenis bekijken

One of the most useful features of TortoiseSVN is the Log dialog. This shows you a list of all the commits you made to a file or folder, and shows those detailed commit messages that you entered (you *did* enter a commit message as suggested? If not, now you see why this is important).



#### Afbeelding 1.4. Het logboekscherf

OK, so I cheated a little here and used a screenshot from the TortoiseSVN repository.

The top pane shows a list of revisions committed along with the start of the commit message. If you select one of these revisions, the middle pane will show the full log message for that revision and the bottom pane will show a list of changed files and folders.

Each of these panes has a context menu which provides you with lots more ways of using the information. In the bottom pane you can double click on a file to see exactly what changes were made in that revision. Read [Paragraaf 4.10, "Logboekboodschappen Venster"](#) to get the full story.

### 1.3.7. Wijzigingen ongedaan maken

Een feature van alle versiebeheersystemen is dat ze je toestaan om wijzigingen, die je eerder hebt gemaakt, ongedaan te maken. Zoals je wellicht al verwacht, maakt TortoiseSVN dit erg gemakkelijk voor je om te doen.

If you want to get rid of changes that you have not yet committed and reset your file to the way it was before you started editing, TortoiseSVN → Revert is your friend. This discards your changes (to the Recycle bin, just in case) and reverts to the committed version you started with. If you want to get rid of just some of the changes, you can use TortoiseMerge to view the differences and selectively revert changed lines.

If you want to undo the effects of a particular revision, start with the Log dialog and find the offending revision. Select Context Menu → Revert changes from this revision and those changes will be undone.

## 1.4. We gaan verder...

This guide has given you a very quick tour of some of TortoiseSVN's most important and useful features, but of course there is far more that we haven't covered. We strongly recommend that you take the time to read the rest of this manual, especially [Hoofdstuk 4, \*Dagelijkse gebruikersgids\*](#) which gives you a lot more detail on day-to-day operations.

We hebben veel moeite gedaan om ervoor te zorgen dat het zowel informatief als makkelijk leesbaar is, maar we erkennen dat het veel is! Neem de tijd en wees niet bang om dingen uit te proberen op een test archief. De beste manier om iets te leren is het uitproberen.

---

# Hoofdstuk 2. Basis Versie-Beheer Concepten

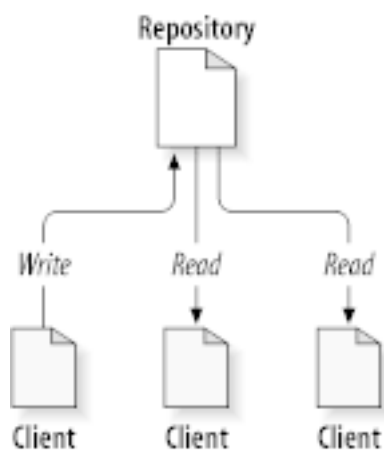
Dit hoofdstuk is een iets gewijzigde versie van hetzelfde hoofdstuk uit het Subversion boek. Een online versie van het Subversion boek is beschikbaar op: <http://svnbook.red-bean.com/>.

Dit hoofdstuk is een korte en informele introductie in Subversion. Als je nieuw bent op gebied van versiebeheer, dan is dit hoofdstuk echt iets voor jou. We beginnen met een uiteenzetting over algemene versiebeheer concepten, verdiepen ons in de specifieke gedachten achter Subversion, en tonen een aantal simpele voorbeelden van toepassingen van Subversion.

Ook al gaan de voorbeelden in dit hoofdstuk over mensen die een verzameling programmacode delen, vergeet niet dat Subversion elke verzameling bestanden kan beheeren - het is niet beperkt tot assisteren van computerprogrammeurs.

## 2.1. Het Archief

Subversion is een centraal systeem voor het delen van informatie. De kern is een *archief*, dat een centrale opslagruimte van data is. Het archief slaat informatie op in de vorm van een *bestandssysteemboom* - een typische hiërarchie van bestanden en mappen. Een ongelimiteerd aantal *clients* kan op het archief aansluiten en deze bestanden lezen of schrijven. Door data te schrijven maakt een client informatie beschikbaar voor anderen; door het lezen van data ontvangt de client informatie van anderen.



**Afbeelding 2.1. Een typisch Client/Server Systeem**

Waarom is dit nu zo interessant? Tot zoverre lijkt dit op de definitie van een typische bestandserver. En inderdaad *is* het archief een soort van bestandserver, maar niet de normale soort. Wat het Subversion archief zo bijzonder maakt, is dat *het elke wijziging onthoudt*, die ooit weggeschreven is: elke wijziging in elk bestand en zelfs wijzigingen in de mappenstructuur, zoals toevoegen, verwijderen en herschikken van bestanden en mappen.

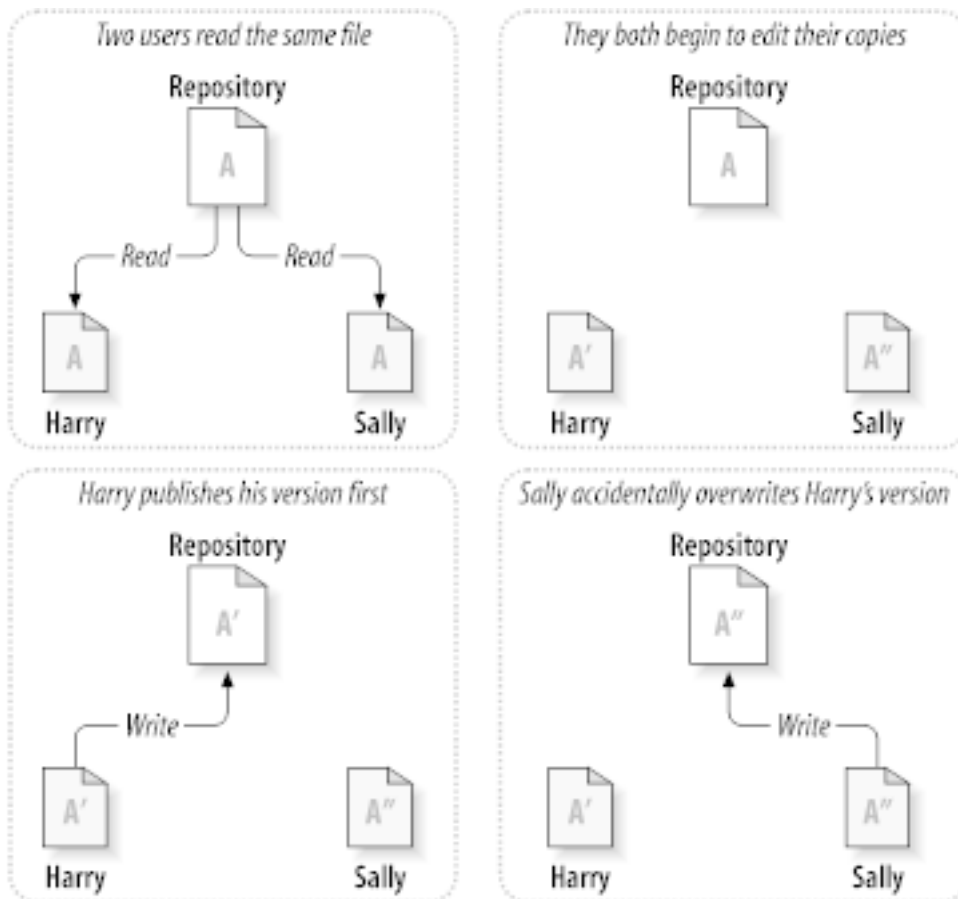
When a client reads data from the repository, it normally sees only the latest version of the filesystem tree. But the client also has the ability to view *previous* states of the filesystem. For example, a client can ask historical questions like, “ what did this directory contain last Wednesday? ”, or “ who was the last person to change this file, and what changes did they make? ” These are the sorts of questions that are at the heart of any *version control system*: systems that are designed to record and track changes to data over time.

## 2.2. Versioneermodellen

Alle versiebeheersystemen moeten hetzelfde fundamentele probleem oplossen: hoe zorgt een systeem ervoor dat gebruikers informatie kunnen delen, maar voorkomt dat ze per ongeluk op elkaars voeten gaan staan? Het is maar al te gemakkelijk voor gebruikers om per ongeluk elkaars wijzigingen te overschrijven in het archief.

### 2.2.1. Het Probleem van Bestand Delen

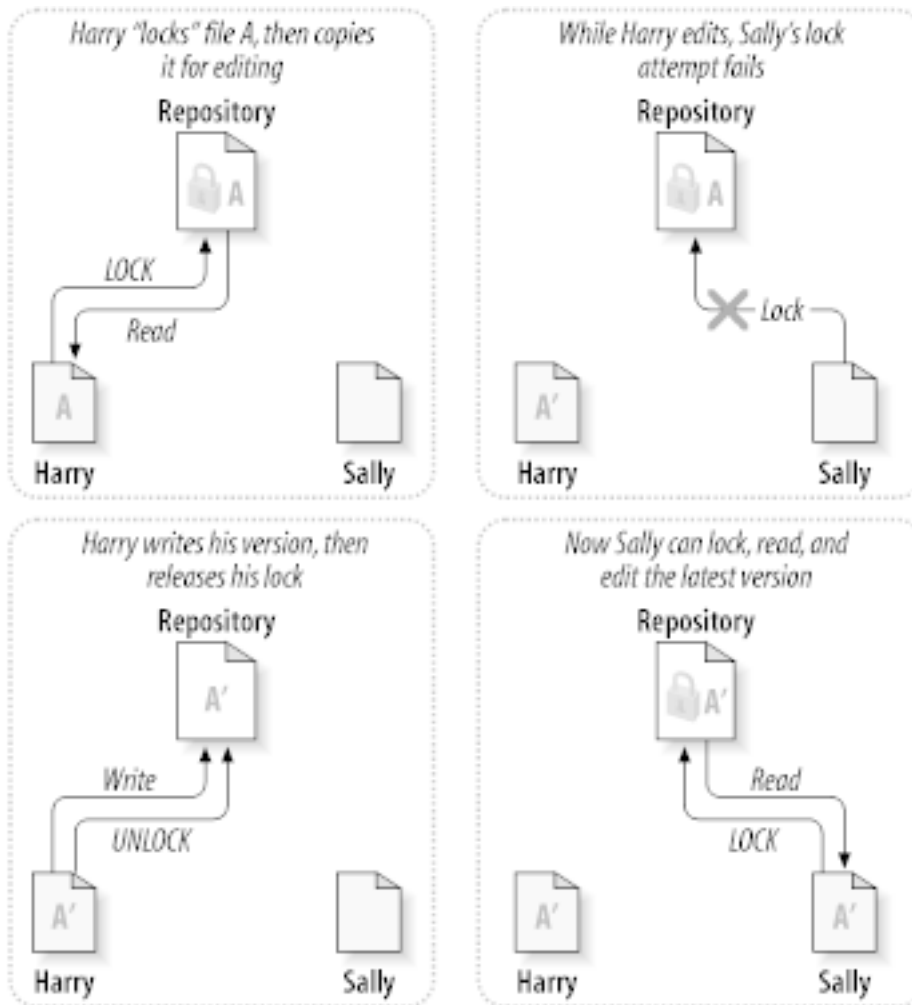
Beschouw het volgende scenario: stel we hebben twee collega's, Harry en Sally. Ze besluiten beide om hetzelfde archief te wijzigen op hetzelfde moment. Als Harry de wijzigingen als eerste in het archief opslaat, dan is het mogelijk dat Sally (enkele ogenblikken later) per ongeluk zijn wijzigingen overschrijft met haar nieuwe versie van het bestand. Alhoewel Harry's wijzigingen niet voor goed verloren zijn (omdat het systeem elke wijziging onthoudt), zijn de wijzigingen die Harry aanbracht *niet* zichtbaar in Sally's nieuwe versie van het bestand, omdat zij Harry's wijzigingen helemaal niet eens gezien heeft. Harry's werk is dus in feite verloren - of ontbreekt in elk geval in de laatste versie van het bestand - en waarschijnlijk ook nog per ongeluk. Dit is precies de situatie die we willen voorkomen!



Afbeelding 2.2. Het probleem om te ontwijken

### 2.2.2. De Vergrendel-Wijzig-Ontgrendel Oplossing

Veel versiebeheersystemen gebruiken een *blokkeren-wijzigen-blokkade opheffen* model om dit probleem te adresseren, hetgeen een zeer simpele oplossing is. In zo'n systeem staat het archief maar aan één persoon tegelijkertijd toe om een bestand te wijzigen op één bepaald moment. Eerst moet Harry het bestand *blokkeren*, voordat hij kan beginnen om het bestand te wijzigen. Blokkeren van een bestand lijkt op het lenen van een boek bij de bibliotheek; als Harry het bestand heeft geblokkeerd, kan Sally geen wijzigingen hierin aanbrengen. Als zij probeert het bestand te blokkeren, zal het archief dit verzoek weigeren. Ze kan alleen het bestand bekijken en wachten tot Harry klaar is met zijn wijzigingen en de blokkade opheft. Nadat Harry de blokkade op het bestand heeft opgeheven, is zijn beurt voorbij en is Sally aan de beurt om het bestand te blokkeren en te wijzigen.



**Afbeelding 2.3. De Vergrendel-Wijzig-Ontgrendel Oplossing**

Het probleem met "blokkeren-wijzigen-blokkade opheffen" is dat het erg beperkend is en vaak voor gebruikers een wegversperring wordt.

- *Blokkeren kan administratieve problemen veroorzaken.* Soms zal het gebeuren dat Harry een bestand blokkeert en dat daarna vergeet. In de tussentijd kan Sally niets doen, omdat ze aan het wachten is om het bestand te mogen wijzigen. En als Harry dan ook nog op vakantie gaat, moet Sally een beheerder vragen om Harry's blokkade op te heffen. Deze situatie veroorzaakt veel onnodige vertraging en verloren tijd.
- *Blokkeren kan onnodige serializatie veroorzaken.* Stel dat Harry aan het begin van een bestand iets wil wijzigen en Sally aan het einde van datzelfde bestand. Deze wijzigingen overlappen elkaar niet. Ze zouden gemakkelijk tegelijkertijd het bestand kunnen wijzigen, waarbij geen grote schade zal ontstaan, ervanuitgaande dat de wijzigingen netjes samengevoegd worden. Er is geen reden om op elkaars beurt te wachten.
- *Blokkeren kan een vals gevoel van veiligheid geven.* Stel dat Harry bestand A blokkeert en wijzigt, terwijl Sally tegelijkertijd bestand B blokkeert en wijzigt. Maar wat als A en B van elkaar afhankelijk zijn, en de wijzigingen die gemaakt zijn, niet compatibel zijn? A en B zullen ineens niet meer samen kunnen werken. Het blokkeringssysteem was niet in staat om dit probleem te voorkomen - maar op de een of andere manier gaf het wel een vals gevoel van veiligheid. Het is voor de hand liggend dat Harry en Sally bedenken dat ze door het blokkeren van de bestanden, een veilige, geïsoleerde taak beginnen, wat ze ervan weerhoudt om hun incompatibele wijzigingen te bespreken.

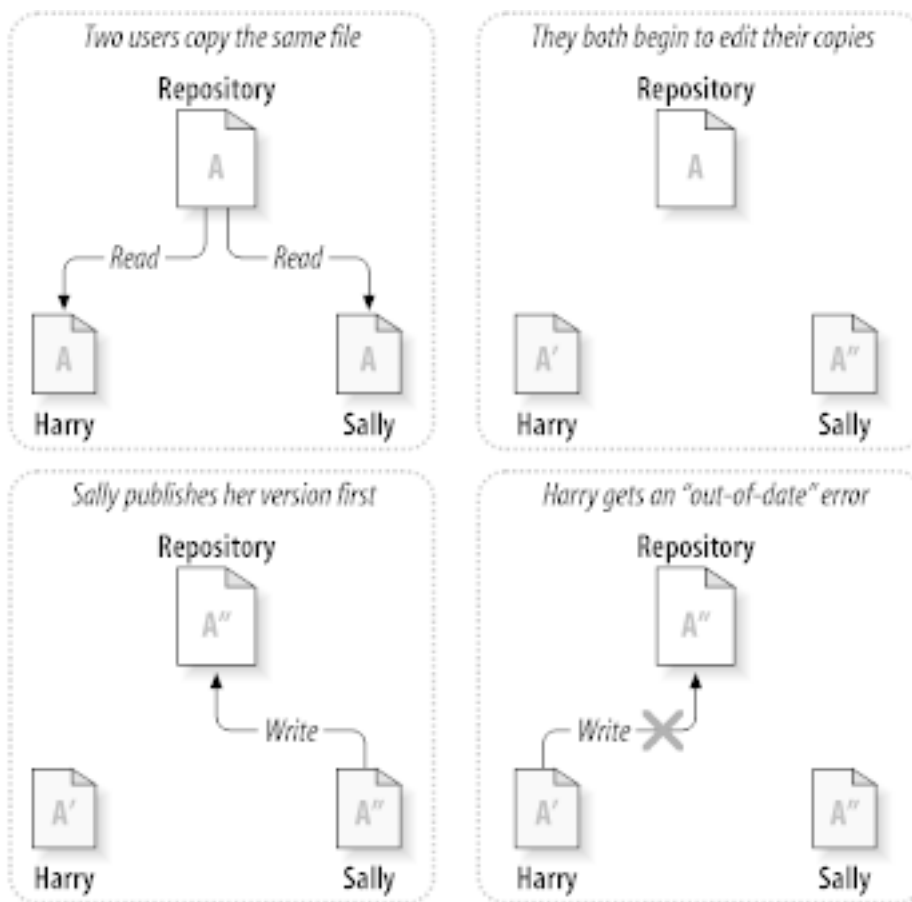
### 2.2.3. De Kopieer-Wijzig-Samenvoeg Oplossing

Subversion, CVS, and other version control systems use a *copy-modify-merge* model as an alternative to locking. In this model, each user's client reads the repository and creates a personal *working copy* of the file or project.

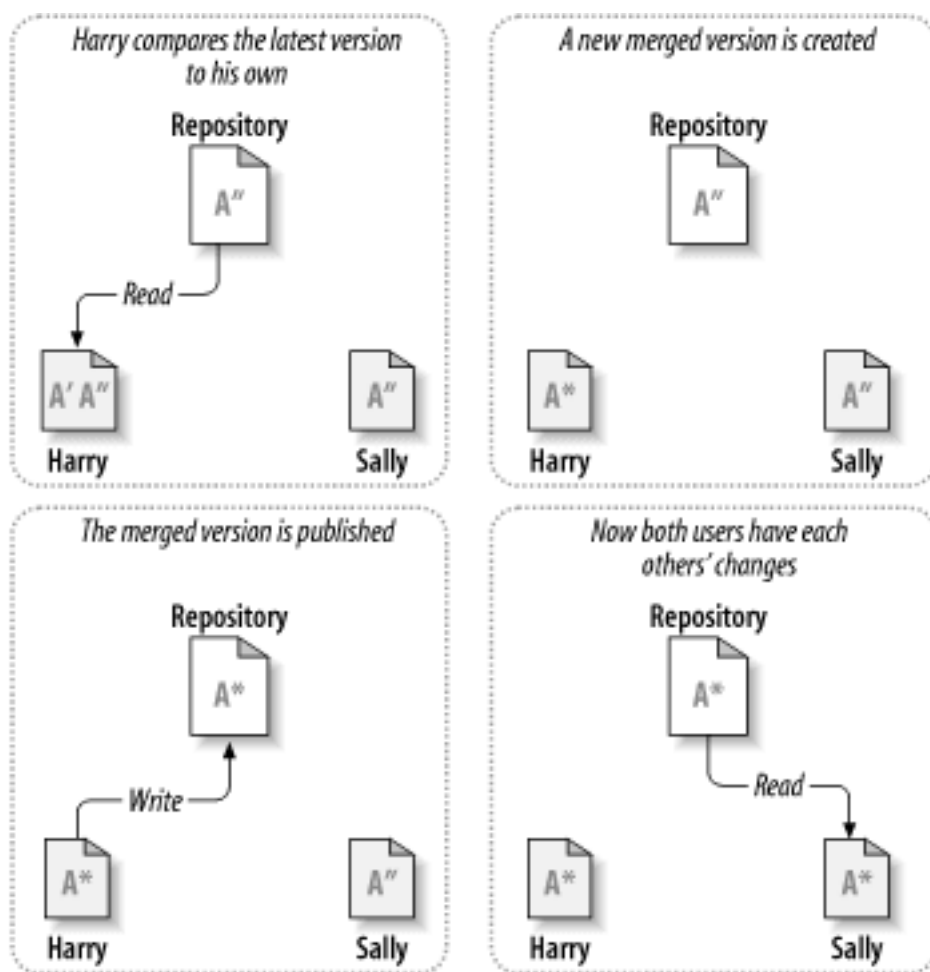


Users then work in parallel, modifying their private copies. Finally, the private copies are merged together into a new, final version. The version control system often assists with the merging, but ultimately a human being is responsible for making it happen correctly.

Zie hier een voorbeeld. Stel dat Harry en Sally elk een werkkopie hebben gemaakt vanaf het archief van hetzelfde project. Ze werken tegelijkertijd en wijzigen hetzelfde bestand A in hun werkkopie. Sally slaat haar wijzigingen als eerste in het archief op. Als Harry later probeert zijn wijzigingen op te slaan, informeert het archief hem dat het bestand A *verouderd* is. Met andere woorden, bestand A in het archief is op de een of andere manier in het archief gewijzigd sinds hij het bestand voor het laatst gekopieerd heeft. Harry zal nu zijn client vragen om de wijzigingen vanuit het archief met zijn werkkopie van bestand A *samen te voegen*. De kans is groot dat wijzigingen die Sally heeft gemaakt, niet overlappen met zijn eigen wijzigingen; dus zodra hij beide sets heeft geïntegreerd, kan hij zijn werkkopie in het archief opslaan.



Afbeelding 2.4. De Kopieer-Wijzig-Samenvoeg Oplossing



**Afbeelding 2.5. ...Kopieren-Wijzigen-Samenvoegen Vervolg**

Maar wat gebeurt er als Sally's wijzigingen *wel* overlappen met Harry's wijzigingen? Wat dan? Deze situatie wordt een *conflict* genoemd, en is meestal helemaal niet zo'n probleem. Als Harry zijn client vraagt om de wijzigingen in het archief samen te voegen met zijn werkkopie, wordt zijn kopie van bestand A gemarkeerd als zijnde in staat van conflict: hij zal beide sets van conflicterende wijzigingen kunnen bekijken en handmatig kiezen tussen de sets. Software kan niet automatisch conflicten oplossen; alleen mensen zijn in staat om te begrijpen en de juiste intelligente keuzes te maken. Zodra Harry handmatig de conflicten heeft opgelost (misschien wel door te overleggen met Sally), kan hij de wijzigingen in het samengevoegde bestand opslaan in het archief.

Het kopiëren-wijzigen-samenvoegen model lijkt misschien wat chaotisch, maar in de praktijk werkt het zeer probleemloos. Gebruikers kunnen parallel werken, en hoeven nooit te wachten op elkaar. Als ze aan dezelfde bestanden werken, zal het in de meeste gevallen geen overlappende wijzigingen betreffen; Conflicten treden niet vaak op. En de tijd die het kost om een conflict op te lossen is veel minder dan de tijd die verloren gaat door het blokkeersysteem.

Uiteindelijk komt het allemaal aan op één kritieke factor: communicatie tussen de gebruikers. Als een gebruiker slecht communiceert, zullen zowel syntactische als semantische conflicten meer gaan optreden. Geen enkel systeem kan gebruikers dwingen om perfect te communiceren, en geen enkel systeem kan semantische conflicten detecteren. Het heeft dus geen enkele zin om in slaap gesust te worden door een blokkeersysteem dat zogenaamd conflicten kan voorkomen; in de praktijk lijkt het of blokkeren de productiviteit meer dan wat dan ook remt.

Er is één voorkomende situatie waar het blokkeren-wijzigen-vrijgeven model beter werkt, en dat is als je bestanden hebt, waarvan wijzigingen niet samenvoegbaar zijn. Als je archief bijvoorbeeld grafische bestanden bevat en twee mensen wijzigen een grafische bestand op hetzelfde moment, is er geen manier om deze wijzigingen samen te voegen. Ofwel de wijzigingen van Harry of die van Sally zullen verloren gaan.

## 2.2.4. Wat doet Subversion?

Subversion gebruikt standaard de kopieer-wijzigen-samenvoegen methode, en in veel gevallen is dat alles wat je nodig hebt. Echter, vanaf versie 1.2, ondersteund Subversion ook het vergrendelen van bestanden. Als je niet samenvoegbare bestanden hebt, of als het management simpelweg het gebruik van de vergrendelwerkwijze oplegt, levert Subversion nog steeds de functionaliteit die je nodig hebt.

## 2.3. Subversion in Actie

### 2.3.1. Werkkopieën

Eerder heb je al wat gelezen over werkkopieën; nu zullen we demonstreren hoe de Subversion client deze creëert en gebruikt.

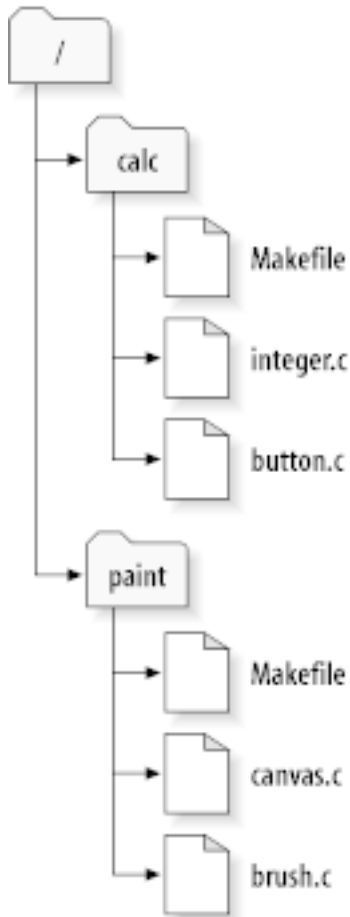
Een Subversion werkkopie is eenvoudigweg een mapstructuur met een verzameling van bestanden op je lokale systeem. Je kunt de bestanden bewerken zoals je wilt en als het broncode bestanden zijn, is het mogelijk om ze vanaf die locatie te compileren op de gebruikelijke manier. Je werkkopie is je eigen privé werkomgeving: Subversion zal zelf geen wijzigingen van anderen in je lokale kopie verwerken, of jouw werk aan anderen beschikbaar stellen, zonder dat je daar zelf opdracht voor hebt gegeven.

Nadat je wijzigingen in je bestanden hebt aangebracht en hebt geverifieerd of deze goed werken, biedt Subversion commando's om je wijzigingen te *publiceren* voor andere mensen die met jou aan het project werken (door middel van het opslaan in het archief). Als andere mensen hun eigen werk publiceren, biedt Subversion je opdrachten om die wijzigingen samen te voegen in je werkkopie (door te lezen vanuit het archief).

A working copy also contains some extra files, created and maintained by Subversion, to help it carry out these commands. In particular, your working copy contains a subdirectory named `.svn`, also known as the working copy *administrative directory*. The files in this administrative directory help Subversion recognize which files contain unpublished changes, and which files are out-of-date with respect to others' work. Prior to 1.7 Subversion maintained `.svn` administrative subdirectories in every versioned directory of your working copy. Subversion 1.7 takes a completely different approach and each working copy now has only one administrative subdirectory which is an immediate child of the root of that working copy.

A standaard Subversion archief bevat vaak de (bron) bestanden voor meerdere projecten; vaak is er voor elk project een submap in de boomstructuur van het archief bestandssysteem. Bij deze indeling correspondeert de werkkopie van een gebruiker meestal met een bepaalde submap van het archief.

Bijvoorbeeld, stel je hebt een archief met twee software projecten.



**Afbeelding 2.6. Het archief bestandssysteem**

Met andere woorden, de hoofdmap van het archief heeft twee submappen: `paint` en `calc`.

To get a working copy, you must *check out* some subtree of the repository. (The term *check out* may sound like it has something to do with locking or reserving resources, but it doesn't; it simply creates a private copy of the project for you.)

Suppose you make changes to `button.c`. Since the `.svn` directory remembers the file's modification date and original contents, Subversion can tell that you've changed the file. However, Subversion does not make your changes public until you explicitly tell it to. The act of publishing your changes is more commonly known as *committing* (or *checking in*) changes to the repository.

Om je wijzigingen te publiceren naar anderen, gebruik je het Subversion commando **Vastleggen**.

Met dit commando worden je wijzigingen in `button.c` vastgelegd in het archief; als een andere gebruiker een werkkopie ophaalt van `/calc`, dan zal deze jouw wijzigingen in de laatste versie zien.

Stel je hebt een medewerker, Sally, die op hetzelfde moment als jij een werkkopie ophaalt van `/calc`. Als jij je wijzigingen aan `button.c` vastlegt, blijft Sally's werk ongewijzigd; Subversion wijzigt alleen werkkopieën op aanvraag van de gebruiker.

Om haar project bij te werken, kan Sally Subversion vragen om haar werkkopie *bij te werken*, door het Subversion commando **Verversen** te gebruiken. Hiermee worden jouw wijzigingen in haar kopie verwerkt, net zoals andere wijzigingen die vastgelegd zijn sinds zij haar werkkopie ophaalde.

Merk op dat Sally niet aan hoeft te geven welke bestanden bijgewerkt moesten worden; Subversion gebruikt hiervoor de informatie in de `.svn` map and informatie uit het archief, om te bepalen welke bestanden bijgewerkt moeten worden.

## 2.3.2. Archief URLs

Subversion archieven kunnen op verschillende methodes benaderd worden – vanaf de lokale harde schijf, of met verschillende netwerk protocollen. Een archief locatie is echter altijd een URL. De vorm van de URL bepaald de toegangsmethode:

Schema	Toegangsmethode
<code>file://</code>	Directe archief toegang op een lokale of netwerk schijf.
<code>http://</code>	Toegang via WebDAV protocol naar een met Subversion-bekende Apache server.
<code>https://</code>	Gelijk aan <code>http://</code> , maar met SSL versleuteling.
<code>svn://</code>	Geautoriseerde TCP/IP toegang met een eigen protocol naar een <code>svnserve</code> server.
<code>svn+ssh://</code>	Geautoriseerd, versleutelde TCP/IP toegang met een eigen protocol naar een <code>svnserve</code> server.

**Tabel 2.1. Archief Toegang URLs**

Over het algemeen wordt voor de Subversion URLs de standaard syntax gebruikt, waardoor het mogelijk is om server namen en poort nummers als onderdeel van de URL te specificeren. De `file://` toegangsmethode wordt normaal gesproken alleen gebruikt voor lokale toegang, alhoewel het ook gebruikt kan worden voor UNC paden naar een netwerk host. De URL krijgt in die gevallen de schrijfwijze `file://hostname/path/to/repos`. Voor een lokale machine is de `hostname` niet nodig of `localhost`. Hierdoor verschijnen lokale paden vaak met drie schuine strepen, `file:///path/to/repos`.

Ook de gebruikers van het `file://` schema op een Windows systeem moet een onofficiële “standaard” syntax gebruiken voor het benaderen van archieven die op dezelfde machine staan, maar op een andere schijf dan de schijf waar op gewerkt wordt. Eén van de twee volgende URL pad vormen kunnen gebruikt worden, waarbij `X` de schijfletter aangeeft waarop het archief zich bevindt.

```
file:///X:/pad/naar/repository
...
file:///X|/pad/naar/repository
...
```

Merk op dat de URL gewone schuine strepen gebruikt, terwijl de standaard (niet URL) methode voor een pad op een Windows machine een schuine streep naar links gebruikt.

Je kunt een FSFS repository via een network share benaderen, dit is *niet* aanbevolen voor diverse redenen:

- Je geeft directe schrijf toegang aan alle gebruikers, deze kunnen per ongeluk de repository verwijderen of breken.
- Not all network file sharing protocols support the locking that Subversion requires. One day you will find your repository has been subtly *corrupted*.
- Je dient de toegangsrechten correct te configureren. Met SAMBA is dit geen eenvoudige opgave.
- Als een gebruiker een nieuwere versie van de cliënt installeert welke het repository formaat bijwerkt naar een nieuwere versie. Dan moeten de andere gebruikers ook upgraden naar de nieuwe cliënt versie om gebruik te kunnen maken van de repository.

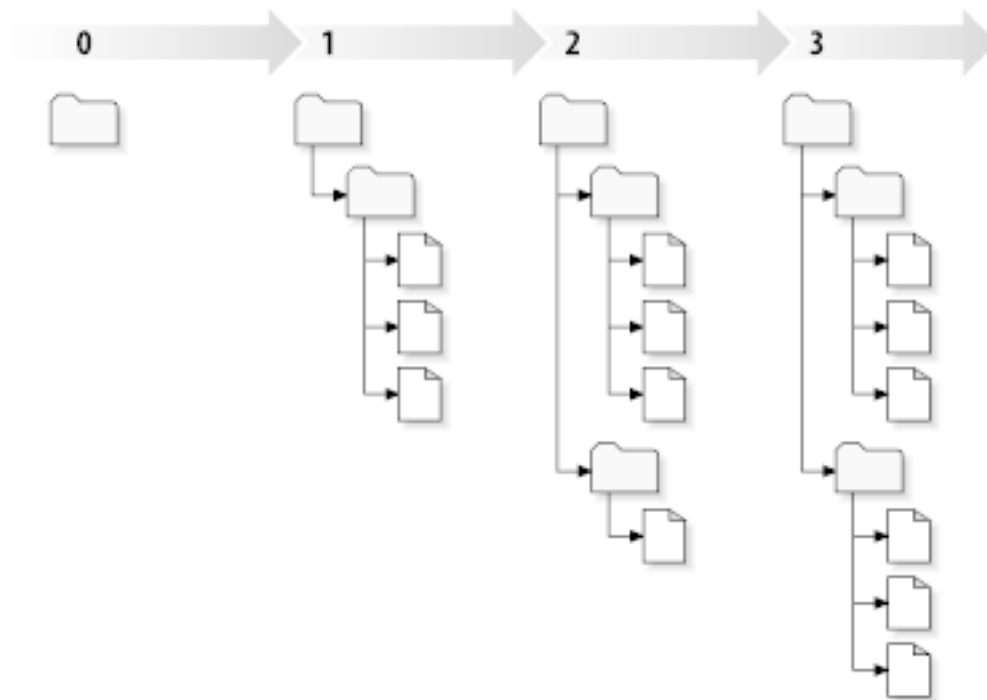
## 2.3.3. Revisies

Een **SVN Vastleggen** actie worden wijzigingen aan meerdere bestanden en mappen als één enkele actie vastgelegd in het archief. In je werkkopie kun je mappen, bestanden en de inhoud daarvan wijzigen, creëren, verwijderen, hernoemen en kopiëren en vervolgens de hele set van wijzigingen in één keer vastleggen.

In het archief wordt ieder vastlegging behandeld als behandeld als een complete transactie in zijn geheel: dat houdt in dat of alle wijzigingen worden vastgelegd of geen enkele wijziging wordt vastgelegd. Subversion houdt vast aan deze werkwijze met het oog op programma crashes, netwerk problemen en andere gebruikersacties.

Elke keer als het archief een Vastlegging accepteert, creëert het een nieuwe status van de bestandssysteem boom, dit wordt een *revisie* genoemd. Elke revisie krijgt een uniek geheel getal, één nummer hoger dan het nummer van de vorige revisie. De initiële revisie van een vers archief heeft nummer nul en heeft alleen een lege hoofdmap.

Een mooie manier om een archief te visualiseren, is het gebruik van serie van bomen. Stel je een verzameling van revisienummers voor, aan de linker kant beginnend bij 0 en oplopend van links naar rechts. Elke revisie heeft een boom van het bestandssysteem onder zich hangen en elke boom is een "momentopnamen" van het archief zoals het er uit zag na het vastleggen.



**Afbeelding 2.7. Het Archief**

#### Globaal Revisie Nummer

Unlike those of many other version control systems, Subversion's revision numbers apply to *entire trees*, not individual files. Each revision number selects an entire tree, a particular state of the repository after some committed change. Another way to think about it is that revision N represents the state of the repository filesystem after the Nth commit. When a Subversion user talks about "revision 5 of foo.c", they really mean "foo.c as it appears in revision 5." Notice that in general, revisions N and M of a file do *not* necessarily differ!

Het is belangrijk om op te merken dat werkkopieën niet altijd overeen komen met één enkele revisie in het archief; er kunnen bestanden in zitten die uit verschillende revisies komen. Bijvoorbeeld, je haalt een werkkopie op uit een archief, waarvan de meest recente revisie nummer 4 is:

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

Op dit moment komt de werk map precies overeen met revisie 4 in het archief. Echter, stel je wijzig nu `button.c` en legt de wijziging vast. Er van uitgaande dat niemand anders wijzigingen heeft vastgelegd, zal jouw toepassing revisie 5 van het archief creëren, en zal je werkkopie er als volgt uit zien.

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

Stel dat op dit moment Sally een wijziging in `integer.c` vastlegt, dan wordt revisie 6 aangemaakt. Als jij zelf dan **svn verversen** uitvoert om je werkkopie bij te werken, dan zal deze er zo uitzien:

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

Sally's wijziging aan `integer.c` zal in je werkkopie zichtbaar worden en je eigen wijziging zal nog steeds in `button.c` zitten. In dit voorbeeld zal de tekst van `Makefile` gelijk zijn voor revisie 4, 5 en 6, maar zal Subversion je werkkopie van `Makefile` markeren met revisie 6. Dit om aan te geven dat dit nog de actuele versie is. Kortweg, als je een schone versie ophaalt van je werkkopie, dan komt die kopie overeen met één enkele revisie in het archief.

### 2.3.4. Hoe Werkkopieën Het Archief Bijhouden

Voor elk bestand in een werk map houdt Subversion twee essentiële stukjes informatie bij in de `.svn/` administratie map:

- what revision your working file is based on (this is called the file's *working revision* ), and
- een tijdstempel met het tijdstip van de laatste keer dat je lokale kopie is ververst vanuit het archief.

Met deze informatie en communicerend met het archief, kan Subversion bepalen in welke van de volgende vier fases een bestand verkeert:

#### Niet gewijzigd en actueel

Het bestand is ongewijzigd in de werk map en er zijn geen wijzigingen in het archief vastgelegd sinds de werk revisie is aangemaakt. Het **vastleggen** en het commando **verversen** op zo'n bestand zal geen actie tot gevolg hebben.

#### Lokaal gewijzigd en actueel

Het bestand is in de werk map gewijzigd en er zijn nog geen wijzigingen aan het bestand vastgelegd in het archief na de basis revisie. De lokale wijzigingen zijn nog niet vastgelegd in het archief, dus het **vastleggen** commando zal je wijzigingen succesvol publiceren en het **verversen** commando zal niets doen.

#### Ongewijzigd en achterhaald

Dit bestand is niet gewijzigd in de werk map, maar er is wel een wijziging in het archief. Dit bestand moet uiteindelijk bijgewerkt worden om overeen te komen met de gepubliceerde revisie. Het **commit** commando zal niets doen, het **verversen** zal de laatste wijzigingen in je werkkopie beschikbaar stellen.

#### Lokaal gewijzigd en achterhaald

Het bestand is lokaal en in het archief gewijzigd. Het **vastleggen** commando zal een *bestand achterhaald* foutmelding geven. Het bestand moet eerst ververst worden; het **verversen** zal proberen de gepubliceerde wijzigingen samen te voegen met de lokale wijzigingen. Als het Subversion niet lukt om de bestanden samen te voegen op een voor hand liggende manier, dan laat Subversion het aan de gebruiker over om dit probleem op te lossen.

## 2.4. Samenvatting

In dit hoofdstuk is een aantal fundamentele Subversion concepten behandeld:

- De begrippen van het centrale archief, de werkkopie van de gebruiker en de verzameling van archief revisie bomen zijn geïntroduceerd.
- Enkele eenvoudige voorbeelden zijn gegeven over hoe twee medewerkers Subversion kunnen gebruiken om wijzigingen naar elkaar te publiceren en van elkaar te ontvangen, door gebruik te maken van het 'kopieer-wijzig-samenvoeg' model.
- Er is kort gesproken over hoe Subversion informatie en wijzigingen bijhoudt in een werkkopie.



---

# Hoofdstuk 3. Het Archief

Onafhankelijk van welk protocol er gebruikt wordt om een archief te benaderen, tenminste één keer moet het archief aangemaakt worden. Dit kan gedaan worden met via een Subversion opdracht scherm of met TortoiseSVN.

Als je nog geen Subversion archief aangemaakt hebt, dan wordt het nu tijd om dat te doen.

## 3.1. Archief creatie

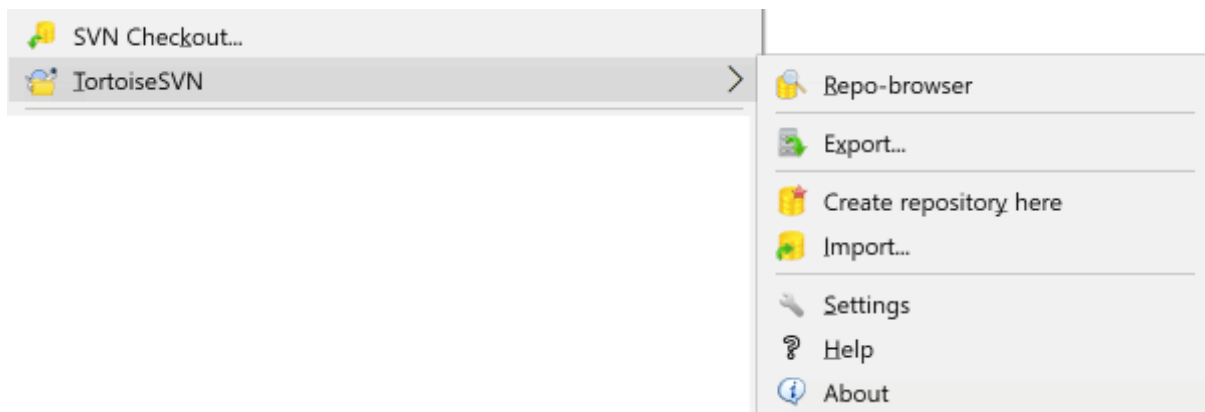
### 3.1.1. Een archief aanmaken via de opdrachtregel applicatie.

1. Maak een lege map aan met de naam SVN (bijv. D:\SVN\), welke gebruikt zal worden als basis map voor allearchieven.
2. Maak nog een map aan `MyNewRepository` in `D:\SVN\`.
3. Open the command prompt (or DOS-Box), change into `D:\SVN\` and type

```
svnadmin create --fs-type fsfs MyNewRepository
```

Nu is er een nieuw archief gemaakt in `D:\SVN\MyNewRepository`.

### 3.1.2. Het Archief Aanmaken Met TortoiseSVN



**Afbeelding 3.1. Het TortoiseSVN menu voor bestanden zonder versiebeheer**

1. Open de Windows Verkenner
2. Maak een nieuwe map aan en noem het bijvoorbeeld `SVNRepository`
3. <action>Klik met de rechter muisknop</action> op de nieuwe map en selecteer <menuchoice><guimenu>TortoiseSVN</guimenu><guimenuitem>Maak hier een archief...</guimenuitem></menuchoice>.

In de nieuwe map wordt dan een archief gemaakt. *Bewerk of wijzig deze bestanden niet zelf!!!*. Als je foutmeldingen krijgt, controleer dan of de map leeg is en of je schrijfrechten hebt.

You will also be asked whether you want to create a directory structure within the repository. Find out about layout options in [Paragraaf 3.1.5, "Archief indeling"](#).

TortoiseSVN will set a custom folder icon when it creates a repository so you can identify local repositories more easily. If you create a repository using the official command line client this folder icon is not assigned.



### Tip

We also recommend that you don't use `file://` access at all, apart from local testing purposes. Using a server is more secure and more reliable for all but single-developer use.

### 3.1.3. Lokale Toegang tot het Archief

Om een lokaal archief te benaderen, moet je het pad naar de map weten. Onthoudt dat Subversion alle paden naar het archief verwacht in het formaat: `file:///C:/SVNRepository/`. Merk op dat er normale schuine strepen worden gebruikt.

Om een archief op een netwerk te benaderen, kan een schijfletter toegewezen worden of door gebruik te maken van het UNC pad. Voor UNC paden moet het formaat `file://ServerName/path/to/repos/` gebruikt worden. Merk op dat er maar 2 schuinen strepen voor staan.

Voordat SVN 1.2 er was, moesten UNC paden op een onduidelijkere manier opgegeven worden `file:/// \ServerName/path/to/repos`. Dit formaat wordt nog wel ondersteund, maar het gebruik wordt niet aanbevolen.

### 3.1.4. Een archief benaderen op een Netwerk schijf

Although in theory it is possible to put a FSFS repository on a network share and have multiple users access it using `file://` protocol, this is most definitely *not* recommended. In fact we would *strongly* discourage it, and do not support such use for various reasons:

- Ten eerste geef je elke gebruiker directe schrijf rechten op het archief, zodat iedereen per ongeluk het hele archief weg kan gooien of het op een andere manier onbruikbaar kan maken.
- Ten tweede ondersteund niet elke netwerk bestandsgebruik protocol de manier van vergrendelen die Subversion nodig heeft, waardoor je archief mogelijk onbruikbaar wordt. Dit hoeft niet direct te gebeuren, maar bijvoorbeeld op een moment dat twee gebruikers het archief tegelijkertijd benaderen.
- Ten derde moeten de permissies exact ingesteld worden, wat misschien nog net lukt met Windows, maar met SAMBA is dit erg lastig.
- Als een gebruiker een nieuwere versie van de cliënt installeert welke het repository formaat bijwerkt naar een nieuwere versie. Dan moeten de andere gebruikers ook upgraden naar de nieuwe cliënt versie om gebruik te kunnen maken van de repository.

`file://` gebruik is bedoeld voor lokaal gebruik met een enkele gebruiker, in het bijzonder voor testen en debuggen. Als je het archief wilt delen, dan moet je *echt* een server opbouwen, wat niet zo moeilijk is als je denkt. Lees [Paragraaf 3.5, "Toegang hebben tot het Archief"](#) voor richtlijnen bij het kiezen en opzetten van een server.

### 3.1.5. Archief indeling

Voordat je je data in het archief opneemt, moet je nadenken hoe je de indeling wilt hebben. Als je één van de aanbevolen indelingen kiest, zul je dat later veel profijt van hebben.

Er zijn een paar standaard aanbevolen indelingen voor een archief. Vaak wordt er een `trunk` map aangemaakt voor de "basislijn" van de ontwikkeling, een `branches` map voor vertakkingen en een `tags` map voor labels (vrijgaves). Als een archief maar één project bevat, worden van deze mappen op hoofdniveau gemaakt:

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

Because this layout is so commonly used, when you create a new repository using TortoiseSVN, it will also offer to create the directory structure for you.

Als een archief meerdere projecten bevat, wordt er vaak voor een indeling per vertakking gekozen:

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

...of per project:

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

Indexeren per project heeft zin als de projecten niet veel met elkaar te maken hebben en elk project afzonderlijk opgehaald wordt uit het archief. Voor projecten die je in één keer wilt ophalen of die met elkaar verbonden zijn in een enkele distributie is het vaak om te indexeren per vertakking. Hiermee hoeft je alleen de basislijn (trunk) op te halen en de relaties tussen de projecten zijn eenvoudiger te zien.

Als de `/trunk /tags /branches` aanpak op hoofdniveau toepast, moet je de hele boomstructuur kopiëren voor elke vertakking en label, en in sommige andere gevallen biedt deze structuur juist de meeste flexibiliteit.

Voor projecten zonder onderlinge relatie heb je mogelijk de voorkeur voor afzonderlijke archieven. Als je wijzigingen vastlegt, wordt het revisienummer van het gehele archief verhoogd, niet het nummer van een project. Het hebben van twee ongerelateerde projecten in één archief betekent dat er grote sprongen kunnen zitten in de revisienummers. De Subversion en TortoiseSVN projecten bevinden zich op hetzelfde hostadres, maar zijn twee aparte archieven waarmee onafhankelijk ontwikkeld kan worden, en dus geen verwarring over build nummers.

Natuurlijk ben je vrij om deze standaard indelingen te negeren. Je kunt elke gewenste variatie creëren, net welke jou en je team het beste past. Onthoudt dat wat je ook kiest, je er niet voor eeuwig aan vast zit. Het is mogelijk om je archief te reorganiseren op elke gewenste moment. Omdat vertakkingen en labels normale mappen zijn, kan TortoiseSVN deze voor je verplaatsen of hernoemen naar hartenlust.

Het schakelen van de ene naar de andere indeling is simpel een door de server uitgevoerde verplaatsactie; Mocht je de indeling van het archief niet handig vinden, verplaats de mappen naar hartenlust.

Als je nog geen basis mapstructuur structuur in het archief hebt aangemaakt, doe dat dan nu. Er zijn twee methodes om dit te doen. Als je de `/trunk /tags /branches` structuur wilt aanmaken, dan kun je de archief-verkenner hiervoor gebruiken (door drie keer apart vastleg acties). Als je een dieper hiërarchische structuur wilt maken, dan is het makkelijker om dat eerst op de schijf te doen en deze structuur dan in één keer te importeren, door:

1. creëer een nieuwe lege map op je harde schijf
2. Maak je gewenste mapstructuur in die map aan – sla er nog geen bestanden in op!
3. import this structure into the repository via a right click on the folder that contains this folder structure and selecting TortoiseSVN → Import... In the import dialog enter the URL to your repository and click OK. This will import your temp folder into the repository root to create the basic repository layout.

Note that the name of the folder you are importing does not appear in the repository, only its contents. For example, create the following folder structure:

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

Import C:\Temp\New into the repository root, which will then look like this:

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

## 3.2. Archief Reservekopie

Onafhankelijk van welke type archief je gebruikt is het van vitaal belang dat je regelmatig een reservekopie maakt en deze controleert. Als er een fout optreedt op de server, kun je misschien nog wel bij een recente versie van je bestanden komen, maar zonder het archief zal alle historie verloren zijn.

The simplest (but not recommended) way is just to copy the repository folder onto the backup medium. However, you have to be absolutely sure that no process is accessing the data. In this context, access means *any* access at all. If your repository is accessed at all during the copy, (web browser left open, WebSVN, etc.) the backup will be worthless.

The recommended method is to run

```
svnadmin hotcopy path/to/repository path/to/backup
```

to create a copy of your repository in a safe manner. Then backup the copy.

The `svnadmin` tool is installed automatically when you install the Subversion command line client. The easiest way to get this is to check the option to include the command line tools when installing TortoiseSVN, but if you prefer you can download the latest version of command line tools directly from the [Subversion](https://subversion.apache.org/packages.html#windows) [https://subversion.apache.org/packages.html#windows] website.

## 3.3. Server gesitueerde hook scripts

A hook script is a program triggered by some repository event, such as the creation of a new revision or the modification of an unversioned property. Each hook is handed enough information to tell what that event is, what target(s) it's operating on, and the username of the person who triggered the event. Depending on the hook's output or return status, the hook program may continue the action, stop it, or suspend it in some way. Please refer to the chapter on [Hook Scripts](http://svnbook.red-bean.com/en/1.8/svn.reposadmin.create.html#svn.reposadmin.create.hooks) [http://svnbook.red-bean.com/en/1.8/svn.reposadmin.create.html#svn.reposadmin.create.hooks] in the Subversion Book for full details about the hooks which are implemented.

Deze hook scripts worden uitgevoerd door de server waarop het archief zich bevindt. Met TortoiseSVN is het mogelijk om eigenlijk hook scripts te maken, die lokaal uitgevoerd worden bij bepaalde gebeurtenissen. Lees [Paragraaf 4.31.8, "Hook Scripts op de Client"](#) voor meer informatie.

Sample hook scripts can be found in the `hooks` directory of the repository. These sample scripts are suitable for Unix/Linux servers but need to be modified if your server is Windows based. The hook can be a batch file or an executable. The sample below shows a batch file which might be used to implement a pre-revprop-change hook.

```
rem Only allow log messages to be changed.
if "%4" == "svn:log" exit 0
echo Property '%4' cannot be changed >&2
```

exit 1

Note that anything sent to stdout is discarded. If you want a message to appear in the Commit Reject dialog you must send it to stderr. In a batch file this is achieved using `>&2`.



## Dwingende Hooks

If a hook script rejects your commit then its decision is final. But you can build an override mechanism into the script itself using the *Magic Word* technique. If the script wants to reject the operation it first scans the log message for a special pass phrase, either a fixed phrase or perhaps the filename with a prefix. If it finds the magic word then it allows the commit to proceed. If the phrase is not found then it can block the commit with a message like “You didn't say the magic word”. :-)

## 3.4. Links voor Ophalen

Als je je Subversion archief beschikbaar wilt stellen aan anderen, dan wil je mogelijk een link naar je archief op je website zetten. Je kunt hiervoor bijvoorbeeld een *Ophalen-link* voor andere TortoiseSVN gebruikers gebruiken.

Bij de installatie van TortoiseSVN, registreert deze een nieuw protocol, `tsvn:`. Als een TortoiseSVN gebruiker op zo'n link klikt, zal automatisch het Ophalen dialoogvenster geopend worden met de URL ingevuld.

Om een dergelijke link in je eigen HTML pagina op te nemen, moet je de code toevoegen die er bijvoorbeeld zo uitziet:

```
<a href="tsvn:http://project.domain.org/svn/trunk">
</a>
```

Of course it would look even better if you included a suitable picture. You can use the [TortoiseSVN logo](https://tortoisesvn.net/images/TortoiseCheckout.png) [https://tortoisesvn.net/images/TortoiseCheckout.png] or you can provide your own image.

```
<a href="tsvn:http://project.domain.org/svn/trunk">
<img src=TortoiseCheckout.png></a>
```

Het is ook mogelijk om een link naar een specifieke revisie te maken, bijvoorbeeld:

```
<a href="tsvn:http://project.domain.org/svn/trunk?100">
</a>
```

## 3.5. Toegang hebben tot het Archief

Voor het gebruik van TortoiseSVN (of een andere Subversion client) moet je een plaatse hebben waar het archief opgeslagen wordt. Je kunt het archief lokaal opslaan en benaderen met het `file://` protocol, of je kunt het archief op een serverplaatsen en het benaderen met het `http://` of `svn://` protocol. De beide server protocollen kunnen ook met een versleuteling werken. Je gebruikt dan `https://`, `svn+ssh://` of `svn://` met SASL.

If you are using a public hosting service such as [SourceForge](https://sourceforge.net) [https://sourceforge.net] or your server has already been setup by someone else then there is nothing else you need to do. Move along to [Hoofdstuk 4, Dagelijkse gebruikersgids](#).

Indien je geen server hebt, alleen werkt of Subversion en TortoiseSVN evalueert, dan is een lokaal archief waarschijnlijk de beste keuze. Maak een archief aan op je PC zoals omschreven in [Hoofdstuk 3, Het Archief](#). De rest van dit hoofdstuk kan overgeslagen worden en lees verder vanaf [Hoofdstuk 4, Dagelijkse gebruikersgids](#).

Als je een archief op een netwerk schijf wilt maken voor meerdere gebruikers, denk daar dan nog eens over na. Lees [Paragraaf 3.1.4, “Een archief benaderen op een Netwerk schijf”](#) om te begrijpen waarom wij dat een

slecht idee vinden. Het opzetten van een server is niet zo moeilijk als het lijkt en een server geeft je een betere betrouwbaarheid en mogelijk snelheid.

More detailed information on the Subversion server options, and how to choose the best architecture for your situation, can be found in the Subversion book under *Server Configuration* [<http://svnbook.red-bean.com/en/1.8/svn.serverconfig.html>].

In the early days of Subversion, setting up a server required a good understanding of server configuration and in previous versions of this manual we included detailed descriptions of how to set up a server. Since then things have become easier as there are now several pre-packaged server installers available which guide you through the setup and configuration process. These links are for some of the installers we know about:

- *VisualSVN* [<https://www.visualsvn.com/server/>]
- *CollabNet* [<https://www.collab.net/products/subversion>]

You can always find the latest links on the *Subversion* [<https://subversion.apache.org/packages.html>] website.

You can find further How To guides on the *TortoiseSVN* [<https://tortoisesvn.net/usefultips.html>] website.

---

# Hoofdstuk 4. Dagelijkse gebruikersgids

Dit document beschrijft het dagelijkse gebruik van de TortoiseSVN client. Het is *geen* introductie tot versiebeheersystemen en *geen* introductie tot Subversion (SVN). Het is meer een document wat je opent als je ongeveer weet wat je wilt doen, maar niet meer weet hoe je het moet doen.

Als je een introductie tot versiebeheer met Subversion zoekt, dan bevelen we het fantastische boek *Version Control with Subversion* [<http://svnbook.red-bean.com/>] aan.

Dit document wordt continue verbeterd, net zoals TortoiseSVN en Subversion. Als je een fout vindt, meld deze dan aan via de maillijst, zodat we de documentatie kunnen bijwerken. Enkele screenshots in dit document geven mogelijk niet de actuele status van de software weer. Vergeef ons alsjeblieft. We werken aan TortoiseSVN in onze vrije tijd.

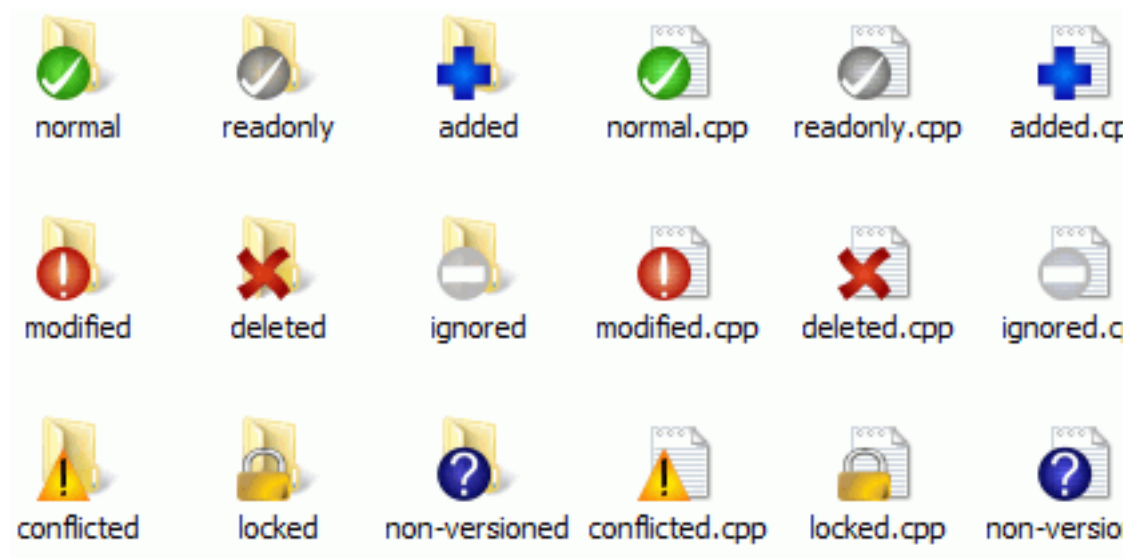
Om het meeste effect te bereiken met de Daily User Guide, moet je:

- TortoiseSVN al geïnstalleerd hebben.
- bekend zijn met versiebeheersystemen.
- de basis van Subversion kennen.
- een server geïnstalleerd hebben en/of toegang hebben tot een Subversion archief.

## 4.1. Algemene Features

This section describes some of the features of TortoiseSVN which apply to just about everything in the manual. Note that many of these features will only show up within a Subversion working copy.

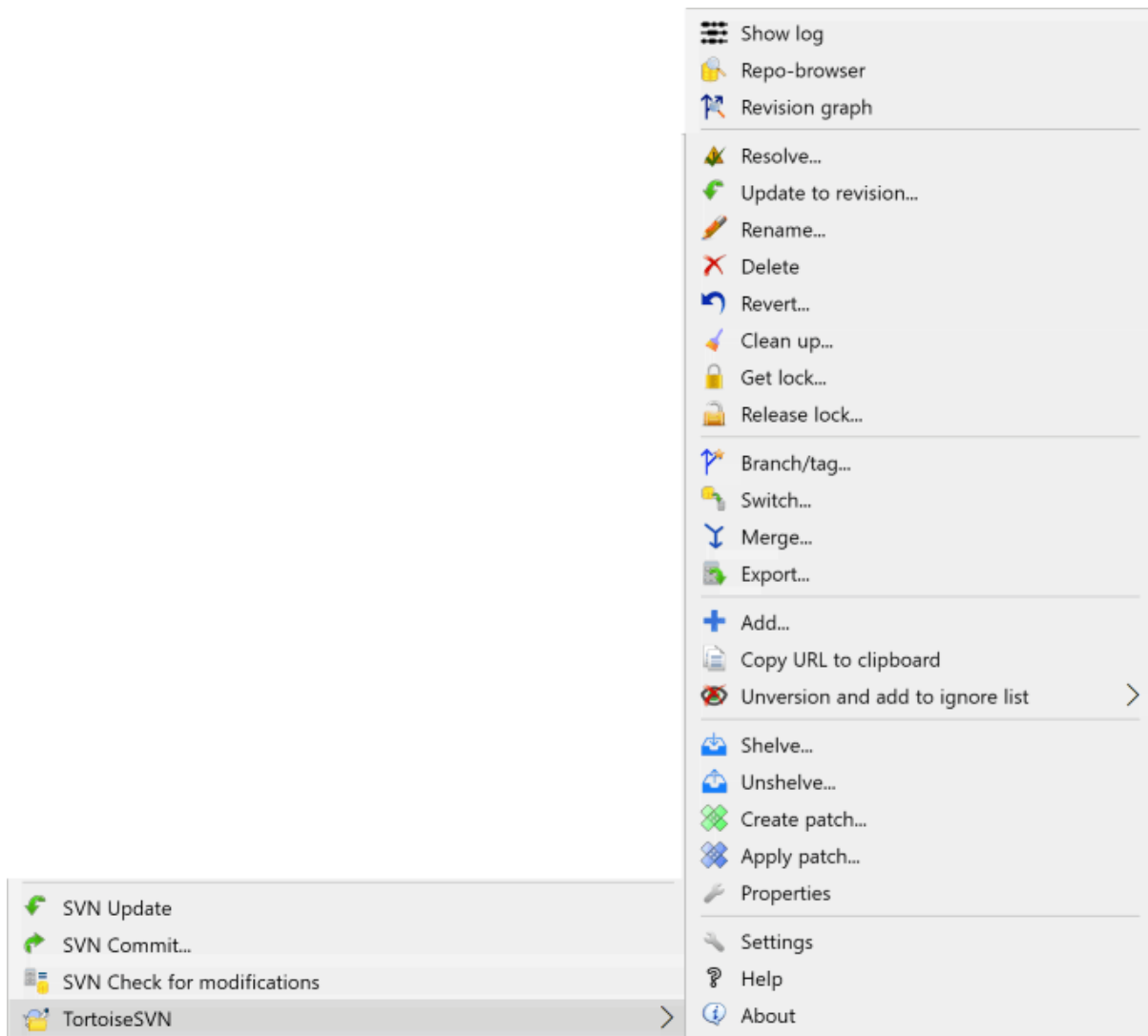
### 4.1.1. Overlappende Pictogrammen



**Afbeelding 4.1. Overlappende pictogrammen weergegeven in de verkener**

Eén van de meest zichtbare functies van TortoiseSVN zijn de overlappende pictogrammen (ook wel icon overlays genoemd), welke weergegeven worden op bestanden in je werkkopie. Deze laten in één oogopslag zien welke bestanden gewijzigd zijn. Lees [Paragraaf 4.7.1, “Overlappende Pictogrammen”](#) om te leren wat de verschillende overlappende pictogrammen betekenen.

### 4.1.2. Contextsmenus



**Afbeelding 4.2. Contextmenu van een directory reeds onder versiebeheer**

Alle TortoiseSVN commando's worden vanuit het context menu van een verkener gegeven. De meeste zijn direct zichtbaar als je met rechts klikt op een bestand of map. Welke commando's beschikbaar zijn, hangt af van het feit of het bestand, de map of de bovenliggende map beheerd wordt met versiebeheer. Het TortoiseSVN menu is ook te zien als onderdeel van het Verkener bestandsmenu.



### Tip

Some commands which are very rarely used are only available in the extended context menu. To bring up the extended context menu, hold down the **Shift** key when you right click.

In enkele gevallen kun je meerdere TortoiseSVN instanties zien. Dit is geen bug!





### Abbeelding 4.3. Verkenner bestandmenu van een snelkoppeling in een versiebeheerde map

Dit is een voorbeeld van een snelkoppeling zonder versiebeheer in een map met versiebeheer, waarbij in het verkenner bestandsmenu *drie* instanties van TortoiseSVN aanwezig zijn. Een is voor de map, een voor de snelkoppeling zelf en een derde voor het object waar de snelkoppeling naar wijst. Om deze instanties te onderscheiden, hebben de iconen een rechtsonder indicator waarmee aangegeven wordt of het een menu is voor een bestand, een map, een snelkoppeling of voor meerdere geselecteerde items.

#### 4.1.3. Slepen en Plaatsen

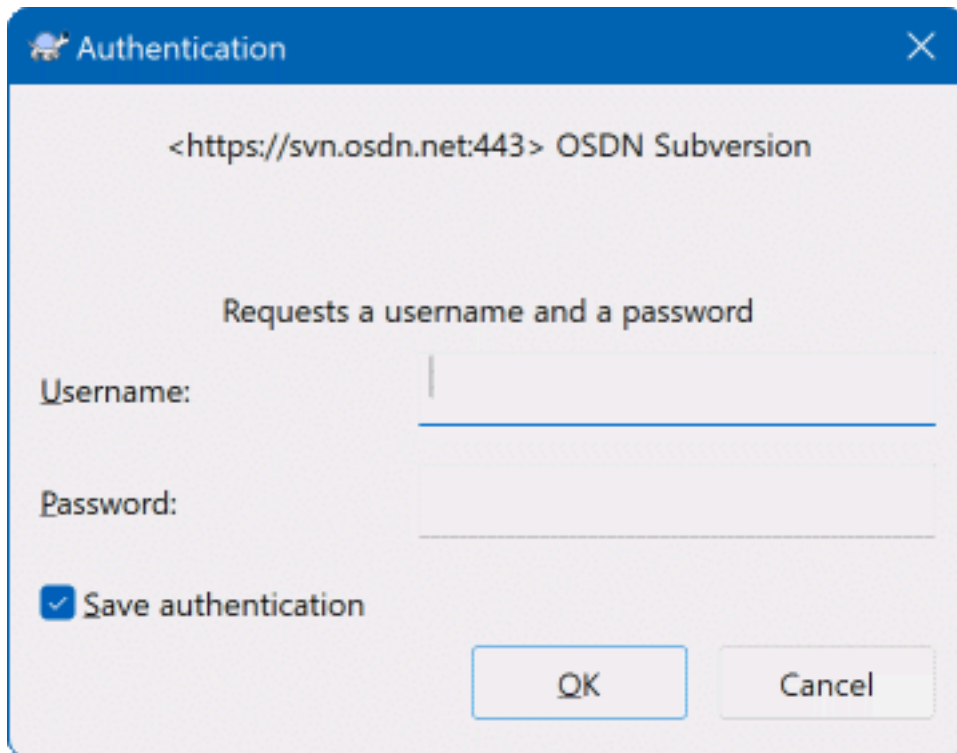
Er zijn andere commando's beschikbaar bij Slepen en plaatsen. Als je bestanden of mappen met rechts sleept naar een nieuwe locatie binnen je werkkopie of als je een bestand of map zonder versiebeheer naar een map met versiebeheer sleept.

#### 4.1.4. Algemene snelkoppelingen

Enkele veel gebruikte acties hebben de welke bekende Windows sneltoets combinaties, hoewel deze niet zijn aangegeven bij de knoppen of in menu's. Als je niet weet hoe je de voor de hand liggende functies moet gebruiken, bijvoorbeeld verversen, lees dan deze paragraaf.

## 4.1.5. Authenticatie

Als je een archief met wachtwoordbeveiliging probeert te bereiken, dan krijg je een authenticatie scherm te zien.



### Afbeelding 4.4. Authenticatie dialoog

Voer je gebruikersnaam en wachtwoord in. Het selectievakje zorgt ervoor dat TortoiseSVN je logingegevens opslaat in Subversion's standaard map: %APPDATA%\Subversion\auth in drie submappen.

- `svn.simple` bevat inloggegevens voor basis authenticatie (gebruikersnaam/wachtwoord). Merk op dat wachtwoorden worden opgeslagen met WinCrypt API encryptie, niet als platte tekst.
- `svn.ssl.server` bevat SSL server certificaten.
- `svn.username` bevat alleen gebruikersnamen, voor authenticatie zonder wachtwoord.

If you want to clear the authentication cache, you can do so from the **Saved Data** page of TortoiseSVN's settings dialog. The button **Clear all** will clear the cached authentication data for all repositories. The button **Clear...** however will show a dialog where you can chose which cached authentication data should be deleted. Refer to [Paragraaf 4.31.6, "Bewaarde Gegevens Instellingen"](#).

Some people like to have the authentication data deleted when they log off Windows, or on shutdown. The way to do that is to use a shutdown script to delete the %APPDATA%\Subversion\auth directory, e.g.

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

You can find a description of how to install such scripts at <http://www.windows-help-central.com/windows-shutdown-script.html>.

Voor meer informatie over hoe u een authenticatie server en een server voor toegangscontrole kan opzetten, zie [Paragraaf 3.5, "Toegang hebben tot het Archief"](#).

### 4.1.6. Maximaliseren van Windows

In verschillende vensters van TortoiseSVN wordt er veel informatie weergegeven. Het is vaak alleen maar nodig om verticaal of horizontaal de volledige schermbreedte te gebruiken, in plaats van te maximaliseren zodat het hele scherm gevuld is. Voor het gemak zijn er snelkoppelingen toegevoegd aan de Maximaliseren knop. Gebruik de middelste muisknop om verticaal te maximaliseren en de rechter muisknop om horizontaal te maximaliseren.

## 4.2. Data Importeren In Een Archief

### 4.2.1. Importeren

Als je data importeert in een bestaand archief waarin al enkele projecten aanwezig zijn, dan is de structuur van het archief al vastgesteld. Als je gegevens importeert in een nieuw archief, dan is het de moeite waard om even goed na te denken over de structuur van het archief. Lees [Paragraaf 3.1.5, “Archief indeling”](#) voor advies hierover.

This section describes the Subversion import command, which was designed for importing a directory hierarchy into the repository in one shot. Although it does the job, it has several shortcomings:

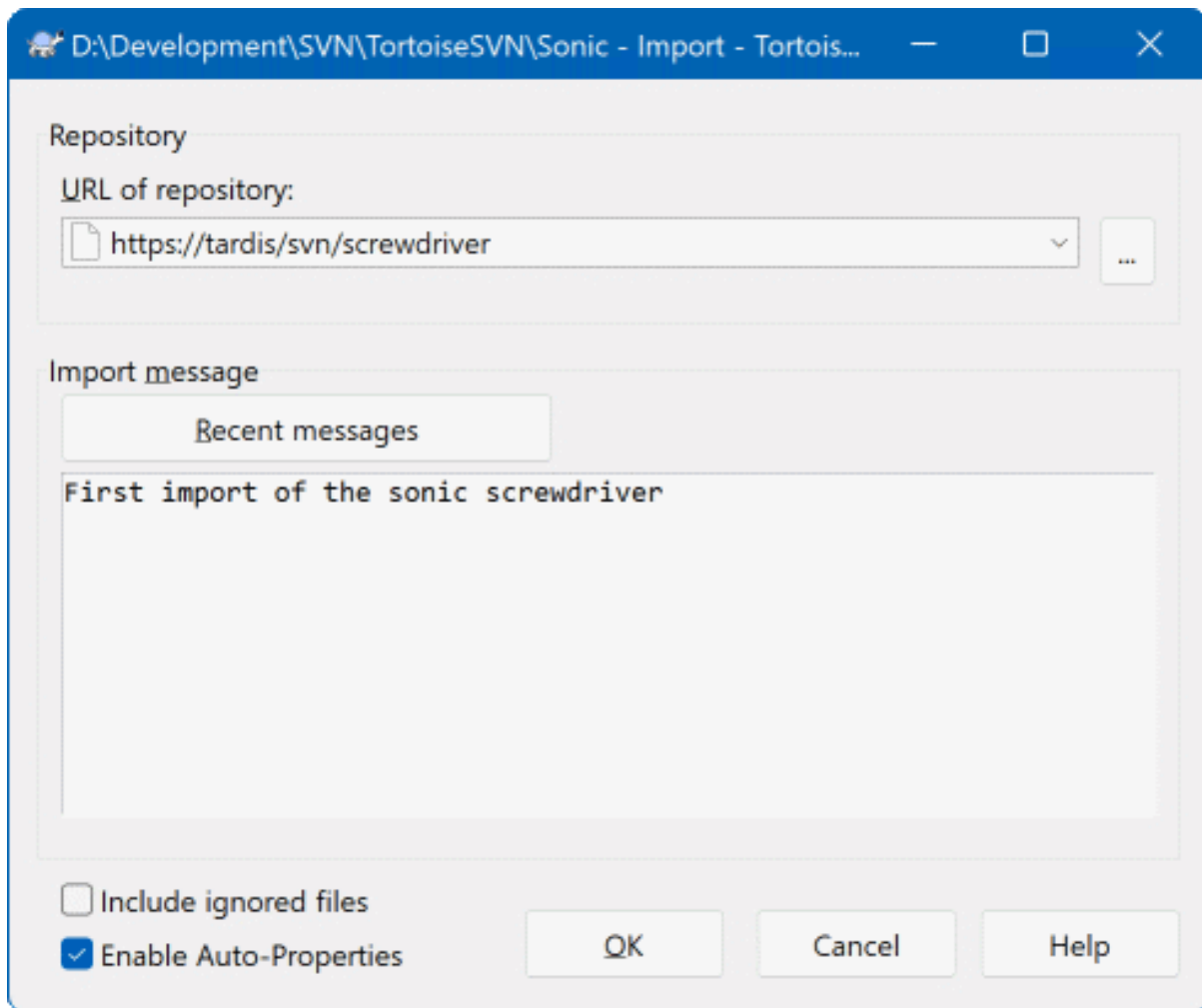
- Het is niet mogelijk om aan te geven welke bestanden en mappen geïmporteerd moeten worden, behalve dan door het gebruik van globale negeer instellingen.
- De geïmporteerde map wordt geen werkkopie. Je moet het Ophalen commando gebruiken om weer een kopie van de bestanden van de server te halen.
- Het importeren vanaf een verkeerd mapniveau gaat erg makkelijk.

For these reasons we recommend that you do not use the import command at all but rather follow the two-step method described in [Paragraaf 4.2.2, “Invoeren op zijn Plaats”](#), unless you are performing the simple step of creating an initial `/trunk /tags /branches` structure in your repository. Since you are here, this is how the basic import works ...

Voordat je je project in het archief importeert, moet je:

1. Alle bestanden verwijderen die niet nodig zijn om het project te bouwen (tijdelijke bestanden, bestanden die door een compiler gemaakt worden bijv. \*.obj, gecompileerde binaire bestanden,...)
2. De bestanden in mappen en sub-mappen sorteren. Ondanks dat het mogelijk is om bestanden te hernoemen of verplaatsen, is het verstandig om je projectstructuur in één keer goed te hebben voordat je importeert.

Selecteer nu in de Windows verkenner de hoofdmap van je project mapstructuur en klik met rechts om het context menu te openen. Selecteer TortoiseSVN → Importeren... waardoor dit venster wordt geopend:



#### Afbeelding 4.5. Het invoer venster

In this dialog you have to enter the URL of the repository location where you want to import your project. It is very important to realise that the local folder you are importing does not itself appear in the repository, only its content. For example if you have a structure:

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

and you import `C:\Projects\Widget` into `http://mydomain.com/svn/trunk` then you may be surprised to find that your subdirectories go straight into `trunk` rather than being in a `Widget` subdirectory. You need to specify the subdirectory as part of the URL, `http://mydomain.com/svn/trunk/Widget-X`. Note that the import command will automatically create subdirectories within the repository if they do not exist.

Het import bericht wordt gebruikt als bericht in het logboek.

Bestanden en mappen die het overeen komen met de globale-negeren patronen worden *niet* geïmporteerd. Om dit gedrag te omzeilen kun je de optie **Genegeerde bestanden meenemen** selecteren. Lees [Paragraaf 4.31.1, "Algemene instellingen"](#) voor meer informatie voor het instellen van een globaal negeren patroon.

Zodra je op OK klikt zal TortoiseSVN de complete mapstructuur inclusief alle bestanden in het archief importeren. Het archief is nu in het archief met versiebeheer opgeslagen. Merk op dat de map die je geïmporteerd hebt **NIET**

nog geen versiebeheer heeft. Om een *werkkopie* met versiebeheer te krijgen moet je eerst nog het project nog Ophalen uit het archief. Of lees verder om te weten te komen hoe je een map direct in versiebeheer opneemt.

### 4.2.2. Invoeren op zijn Plaats

Als je al een archief hebt en je daar een nieuwe mapstructuur aan wilt toevoegen, volg dan deze stappen:

1. Use the repository browser to create a new project folder directly in the repository. If you are using one of the standard layouts you will probably want to create this as a sub-folder of trunk rather than in the repository root. The repository browser shows the repository structure just like Windows explorer, so you can see how things are organised.
2. Checkout the new folder over the top of the folder you want to import. You will get a warning that the local folder is not empty. Ignore the warning. Now you have a versioned top level folder with unversioned content.
3. Gebruik de optie TortoiseSVN → Toevoegen... op deze map met versiebeheer om bepaalde of alle inhoud toe te voegen. Je kunt bestanden toevoegen en verwijderen, `svn:ignore` eigenschappen instellen op mappen en andere wijzigingen die je nodig hebt.
4. Leg de hoofmap vast (SVN Vastleggen) en je hebt een nieuwe mapstructuur met versiebeheer en een lokale werkkopie gemaakt van je bestaande map.

### 4.2.3. Speciale Bestanden

Soms wil je een bestand van versiebeheer voorzien, waarin gebruiker specifieke data staat. Dat houdt dat je een bestand hebt die elke ontwikkelaar/gebruiker aan zijn/haar lokale instellingen wil aanpassen. Het is lastig om zo'n bestand van versiebeheer te voorzien, want elke gebruiker zou zijn/haar wijzigingen telkens opslaan in het archief.

In zulke situaties bevelen we het gebruik van sjablonen aan. Maak een bestand aan met alle gegevens die je gebruikers nodig hebben. Sla het bestand op in het archief en laat de ontwikkelaars het bestand ophalen. Vervolgens moet elke ontwikkelaar een *kopie maken* en deze hernoemen. Daarna levert het wijzigen van het bestand geen problemen meer op.

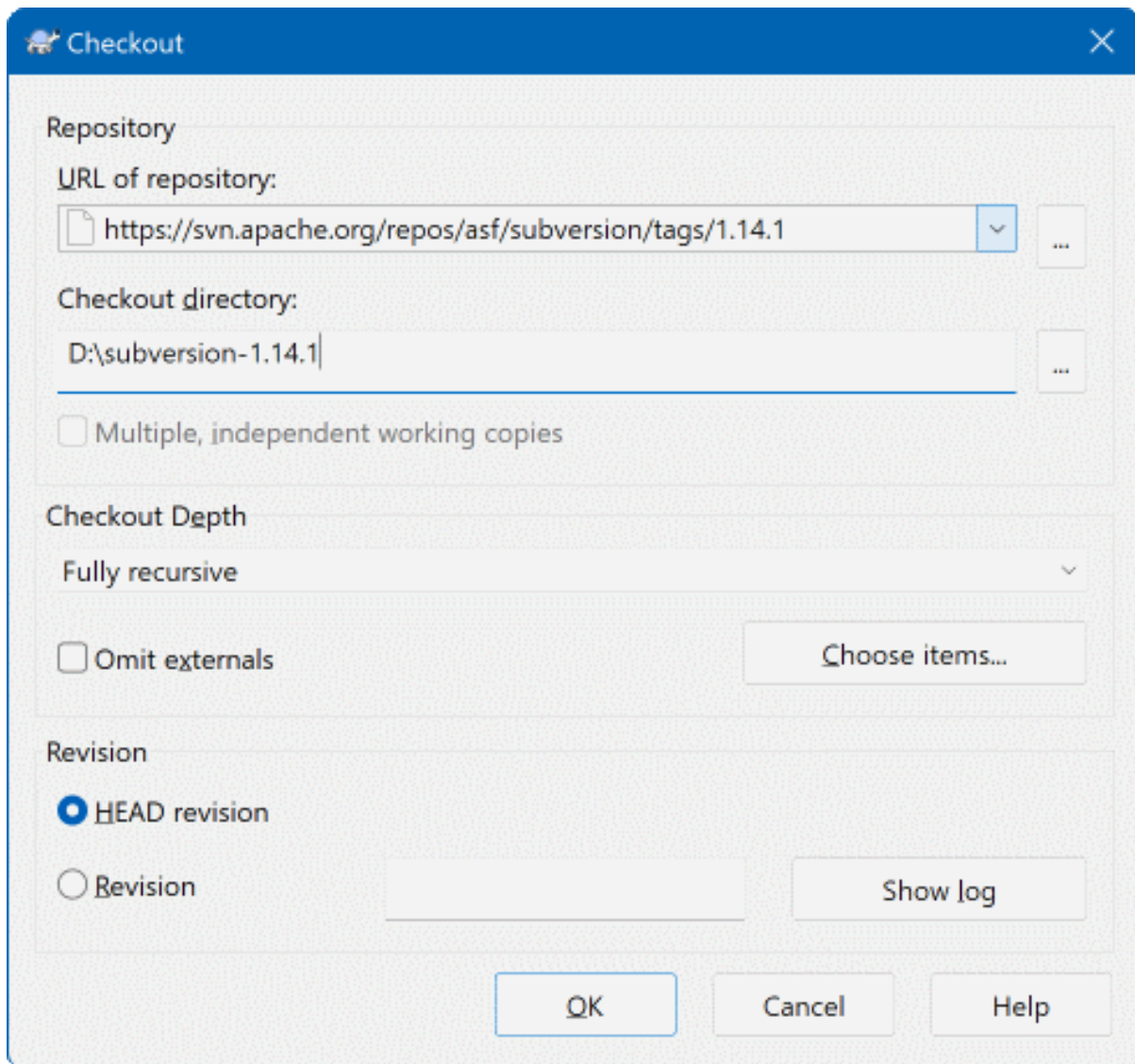
As an example, you can have a look at TortoiseSVN's build script. It calls a file named `default.build.user` which doesn't exist in the repository. Only the file `default.build.user.tmp`. `default.build.user.tmp` is the template file which every developer has to create a copy from and rename that file to `default.build.user`. Inside that file, we added comments so that the users will see which lines they have to edit and change according to their local setup to get it working.

So as not to disturb the users, we also added the file `default.build.user` to the ignore list of its parent folder, i.e. we've set the Subversion property `svn:ignore` to include that filename. That way it won't show up as unversioned on every commit.

## 4.3. Een lokale werkkopie (WC) ophalen

Om een werkkopie te verkrijgen, moet je de bestanden *ophalen* uit het archief.

Selecteer in windows verkenner een map waarin je je werkkopie wilt hebben. Klik rechts om het context menu te openen en selecteer TortoiseSVN → SVN Ophalen..., waarna het volgende venster wordt geopend:



**Afbeelding 4.6. Het Ophalen venster**

Als je een mapnaam invoert die nog niet bestaat, dan zal deze automatisch worden aangemaakt.



### Belangrijk

In the default setting, the checkout menu item is not located in the TortoiseSVN submenu but is shown at the top explorer menu. TortoiseSVN commands that are not in the submenu have SVN prepended: SVN Checkout...

#### 4.3.1. Ophaaldiepte

Je kunt de *diepte* van de recursiviteit instellen van de mappen en gegevens die je wilt ophalen. Als je maar een bepaald deel wilt hebben van een grote mapstructuur, dan kun je alleen de hoofdmap ophalen en vervolgens de mappen selecteren die je recursief wilt ophalen.

Volledig recursief

Gehele boom uitchecken, inclusief alle knopen en subfolders

Enkel niveau, inclusief folders

Haal de gespecificeerde map op, inclusief alle bestanden en onderliggende mappen, maar niet de inhoud van de onderliggende mappen.

#### Alleen bestanden

Haal de gespecificeerde map op, inclusief alle bestanden, maar haal niet de mappen op.

#### Alleen dit object

Haal alleen de map op. Haal de bestanden en onderliggende mappen niet op.

#### Werkkopie

Behoudt de mapdiepte zoals in de werkkopie gespecificeerd is. Deze optie is niet gebruikt in het Ophalen venster, maar wordt standaard gebruikt bij vensters die mapdiepte gebruiken.

#### Exclusief

Wordt gebruikt om de werkkopie mapdiepte te beperken, nadat een map al met de inhoud is gevuld. Deze optie is alleen beschikbaar in het **Verversen naar revisie** venster.

To easily select only the items you want for the checkout and force the resulting working copy to keep only those items, click the **Choose items...** button. This opens a new dialog where you can check all items you want in your working copy and uncheck all the items you don't want. The resulting working copy is then known as a *sparse checkout*. An update of such a working copy will not fetch the missing files and folders but only update what you already have in your working copy.

If you check out a sparse working copy (i.e., by choosing something other than `fully recursive` for the checkout depth), you can easily add or remove sub-folders later using one of the following methods.

#### 4.3.1.1. Sparse Update using Update to Revision

Right click on the checked out folder, then use TortoiseSVN → **Update to Revision** and select **Choose items...** This opens the same dialog that was available in the original checkout and allows you to select or deselect items to include in the checkout. This method is very flexible but can be slow as every item in the folder is updated individually.

#### 4.3.1.2. Sparse Update using Repo 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...**

#### 4.3.1.3. Sparse Update using Check for Modifications

In the check for modifications dialog, first **shift** click on the button **Check repository**. The dialog will show all the files and folders which are in the repository but which you have not checked out as `remotely added`. Right click on the folder(s) you would like to add to your working copy, then use **Context menu** → **Update**.

Deze functie is handig als je maar een klein deel wilt ophalen van een grote mapstructuur, maar ook het gemak wilt hebben dat je maar één werkkopie hoeft bij te werken. Stel je heb een grote mapstructuur met de submappen `Project01` tot `Project99`, waarvan je alleen `Project03`, `Project25` en `Project76/SubProj` wilt ophalen. Doorloop dan de volgende stappen:

1. Haal de hoofdmap op met diepte "Alleen dit object". Je krijgt dan alleen een lege hoofdmap.
2. Selecteer de nieuwe map en kies TortoiseSVN → **Archief-verkenner** om de inhoud van het archief te bekijken.
3. Klik met rechts op `Project03` en **Context menu** → **Verversen naar revisie...** Laat de standaard instellingen zo staan en klik op **OK**. Die map is nu volledig gevuld.

Herhaal dezelfde stappen voor `Project25`.

4. Ga naar `literal>Project76/SubProj`



## Veranderen van de werkkopie mapdiepte

Once you have checked out a working copy to a particular depth you can change that depth later to get more or less content using Context menu → Update item to revision.... In that dialog, be sure to check the Make depth sticky checkbox.



## Een oudere server gebruiken

Servers met oudere versies dan 1.5 begrijpen de werkkopie mapdiepte aanvragen niet. Deze kunnen dus niet altijd efficiënt met zulke aanvragen omgaan. Het commando werkt wel, maar een oudere server verzendt waarschijnlijk alle data. De lokale applicatie moet de data dan filteren. Dit kan betekenen dat er meer netwerkverkeer optreedt dan gewenst. Als het mogelijk is, zou je de server moeten bijwerken naar versie 1.5 of hoger.

Als het project referenties heeft naar externe projecten die je *niet* direct wilt ophalen, gebruik dan het selectievak Externen uitsluiten.



## Belangrijk

Als Externen uitsluiten aangevinkt is of je wilt de mapdiepte vergroten, dan moet je hiervoor gebruikmaken van TortoiseSVN → Verversen naar revisie... in plaats van TortoiseSVN → Verversen. De normale Verversen functie zal alle externe verbindingen verversen en de bestaande mapdiepte blijven gebruiken.

Het is aanbevolen om alleen de *basislijn* (trunk) of dieper gelegen mappen van je mapstructuur op te halen. Als je de hoofdmap van het hele archief in je URL opneemt, dan zal je harddisk waarschijnlijk vollopen. Je krijgt dan namelijk een complete kopie van het hele archief, inclusief alle takken en labels van je project.



## Exporteren

Soms wil je een locale kopie maken zonder de .svn mappen, bijvoorbeeld om een zip-bestand te maken van je bronbestanden. Lees [Paragraaf 4.27, "Exporteren van een Subversion Werkkopie"](#) om te weten te komen hoe dit moet.

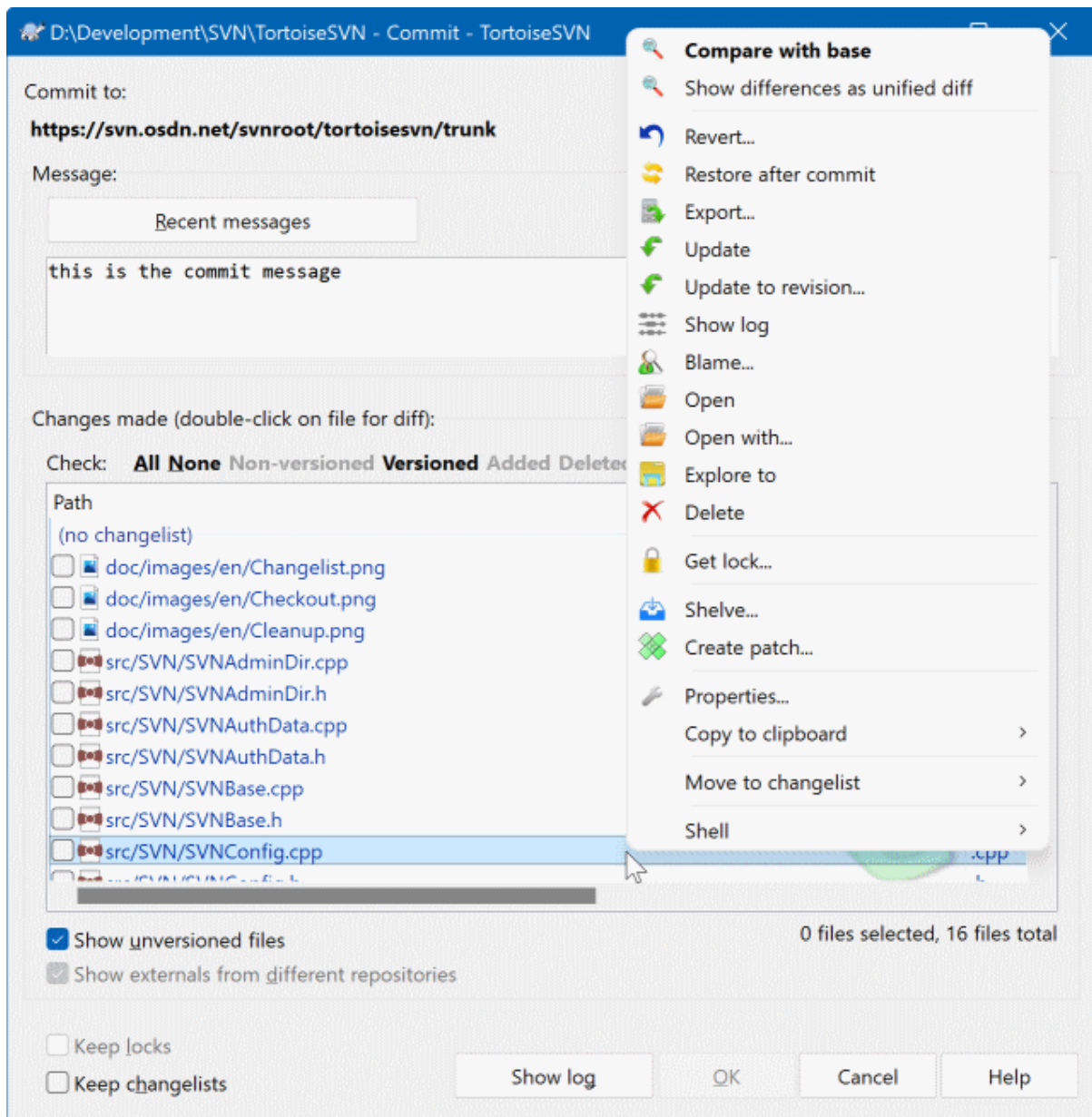
## 4.4. Je Wijzigingen Vastleggen In Het Archief

Het versturen van de wijzigingen die jij hebt gedaan, staat bekend als *vastleggen* van de wijzigingen. Voordat je je wijzigingen vastlegt, moet je zekerstellen dat je werkkopie bijgewerkt is naar de laatste stand van zaken. Dit kun je doen door TortoiseSVN → Verversen te kiezen. Je kunt ook eerst TortoiseSVN → Kijk of er updates zijn gebruiken, om te bekijken welke bestanden er lokaal en op de server gewijzigd zijn.

### 4.4.1. De Commit Dialoog

Als je werkkopie helemaal bij is en er zijn geen conflicten, dan ben je klaar om je wijzigingen vast te leggen. Selecteer de bestanden en mappen die je wilt vastleggen en kies TortoiseSVN → Vastleggen....





### Afbeelding 4.7. De Commit dialog

De vastleggen dialoog geeft elk gewijzigd bestand weer, inclusief toegevoegde, verwijderde en bestanden zonder versiebeheer. Deselecteer een gewijzigd bestand, als je deze niet wilt vastleggen. Als je een bestand zonder versiebeheer in het archief wilt opnemen, dan kun je deze direct aanvinken zodat deze ook vastgelegd wordt.

To quickly check or uncheck types of files like all versioned files or all modified files, click the link items just above the list of shown items.

For information on the coloring and overlays of the items according to their status, please see [Paragraaf 4.7.3, "Lokale en Server Status"](#).

Objecten die verplaatst zijn naar een andere archieflocatie worden aangeduid met een (s) teken. Het kan zijn dat je omgeschakeld bent terwijl je werkte aan een tak en vergeten was om terug te schakelen naar de basislijn. Dit is je waarschuwing!



## Bestanden of mappen vastleggen?

Als je bestanden vastlegt, dan zal de vastleggen dialoog alleen de bestanden laten zien die je geselecteerd hebt. Als je een map vastlegt, dan zal de vastleggen dialoog de gewijzigde bestanden automatisch selecteren. Als je een nieuw aangemaakt bestand vergeten bent, dan wordt deze automatisch gevonden als je een map vastlegt. Het vastleggen van een map betekend *niet* dat elk bestand als gewijzigd wordt aangeduid; het wordt alleen maar eenvoudiger voor doordat het je werk uit handen neemt.



## Veel bestanden zonder versiebeheer in de vastleggen dialoog.

Als je vindt dat het vastleggen venster te veel bestanden zonder versiebeheer weergeeft (bijv. reservekopie bestanden van de compiler of tekstverwerker), dan zijn daar verschillende oplossingen voor. Je kunt:

- het bestand (of een wildcard extensie) toevoegen aan de lijst met de te negeren bestanden op de instellingen pagina. Dit zal op elke werkkopie die je hebt worden toegepast.
- het bestand toevoegen aan de literal>svn:ignore met TortoiseSVNVerwijder en voeg toe aan negerlijst . Dit zal alleen de mappen raken waarvoor de svn:ignore eigenschap is ingesteld. Met het SVN Eigenschappen venster kun je de svn:ignore eigenschap van een map aanpassen.add the file to the svn:global-ignores list using TortoiseSVNAdd to ignore list (recursively) This will affect the directory on which you set the svn:global-ignores property and all subfolders as well.

Dubbelklikken op een gewijzigd bestand in het vastleggen venster zal het externe vergelijkingsprogramma opstarten om jouw wijzigingen te laten zien. Het context menu zal je meer opties geven, zoals weergegeven in de screenshot. Je kunt ook bestanden hier vandaan verslepen naar andere programma's zoals een tekstverwerker of IDE.

Je kunt objecten selecteren en de-selecteren door op het selectievak links ervan te klikken. Bij mappen kun je **Shift**-selecteren gebruiken om de actie recursief te maken.

De kolommen in het onderste venster kunnen aangepast worden. Als je met rechts klikt op de koptekst van een kolom, dan wordt er een context menu weergegeven waarmee je kunt selecteren welke kolommen er getoond moeten worden. Je kunt ook de breedte van de kolommen aanpassen door de overgang tussen twee kolommen in het koptekst van de kolommen te verslepen. De aanpassingen worden opgeslagen, waardoor je de volgende keer dezelfde weergave weer terugkrijgt.

Als je een vastlegging opstart, wordt standaard alle blokkades die jij gezet hebt op bestanden verwijderd nadat de vastlegging succesvol is afgerond. Als je deze blokkades wilt behouden, zorg er dan voor dat het **Blokkades behouden** selectievak is aangekruist. De standaard status van deze optie wordt gelezen uit de `no_unlock` optie in het Subversion configuratiebestand. Lees [Paragraaf 4.31.1, "Algemene instellingen"](#) voor informatie over hoe je het configuratiebestand kunt aanpassen.



## Warning when committing to a tag

Usually, commits are done to the trunk or a branch, but not to tags. After all, a tag is considered fixed and should not change.

If a commit is attempted to a tag URL, TortoiseSVN shows a confirmation dialog first to ensure whether this is really what is intended. Because most of the time such a commit is done by accident.

However, this check only works if the repository layout is one of the recommended ones, meaning it uses the names `trunk`, `branches` and `tags` to mark the three main areas. In case the setup is different, the detection of what is a tag/branch/trunk (also known as `classification patterns`), can be configured in the settings dialog: [Paragraaf 4.31.2, "Revisie Grafiek Instellingen"](#)



## Slepen en Plaatsen

Je kunt bestanden van een andere locatie naar het vastleggen venster slepen, als de werkkopieën maar in het zelfde archief zitten. Bijvoorbeeld, je hebt een enorm grote werkkopie met meerdere windows verkenners open om verschillende mappen in de hiërarchie te bekijken. Als je niet vanaf je hoofdmap wilt vastleggen (met een langdurende zoektoch naar alle wijzigingen), dan kun je het vastleggen venster openen op één map en bestanden vanaf een andere locatie in het venster slepen, zodat het in één actie meegenomen wordt bij het vastleggen.

Je kunt bestanden zonder versiebeheer in de werkkopie ook naar het vastleggen venster slepen, dan worden deze automatisch in SVN opgeslagen.

Dragging files from the list at the bottom of the commit dialog to the log message edit box will insert the paths as plain text into that edit box. This is useful if you want to write commit log messages that include the paths that are affected by the commit.



## Externe Hernoemingen Repareren

Soms worden bestanden hernoemt buiten Subversion, waardoor deze bestanden in de lijst weergegeven worden als missend of zonder versiebeheer. Om te voorkomen dat je de geschiedenis verliest, moet je Subversion vertellen dat er een verbinding is. Selecteer de oude (missende) naam en de nieuwe nieuwe en gebruik Context Menu → Repareer Verplaatsing om de bestanden aan elkaar te koppelen als hernoemden.



## Herstellen van Bestandhernoemingen bij externe kopieën

Als je een kopie van een bestand hebt gemaakt en vergeten bent om het Subversion commando te gebruiken, dan kun je dat repareren om te ervoor te zorgen dat de geschiedenis niet verloren gaat. Selecteer de oude (normaal of gewijzigde) en de nieuwe naam (zonder versiebeheer) en gebruik Context Menu → Repareer Kopieer-actie om de twee bestanden aan elkaar te koppelen als kopie.

### 4.4.2. Werksets

Het vastleggen venster ondersteunt de werkset functie van Subversion om te helpen bij het groeperen van gerelateerde bestanden. Lees meer over deze functie in [Paragraaf 4.8, "Werksets"](#).

### 4.4.3. Alleen delen van bestanden inchecken

Sometimes you want to only commit parts of the changes you made to a file. Such a situation usually happens when you're working on something but then an urgent fix needs to be committed, and that fix happens to be in the same file you're working on.

right click on the file and use Context Menu → Restore after commit. This will create a copy of the file as it is. Then you can edit the file, e.g. in a text editor and undo all the changes you don't want to commit. After saving those changes you can commit the file.



## Het gebruiken van TortoiseMerge

If you use TortoiseMerge to edit the file, you can either edit the changes as you're used to, or mark all the changes that you want to include. right click on a modified block and use Context Menu → Mark this change to include that change. Finally right click and use Context Menu → Leave only marked changes which will change the right view to only include the changes you've marked before and undo the changes you have not marked.

After the commit is done, the copy of the file is restored automatically, and you have the file with all your modifications that were not committed back.

#### 4.4.4. Objecten van de vastleggingslijst uitsluiten

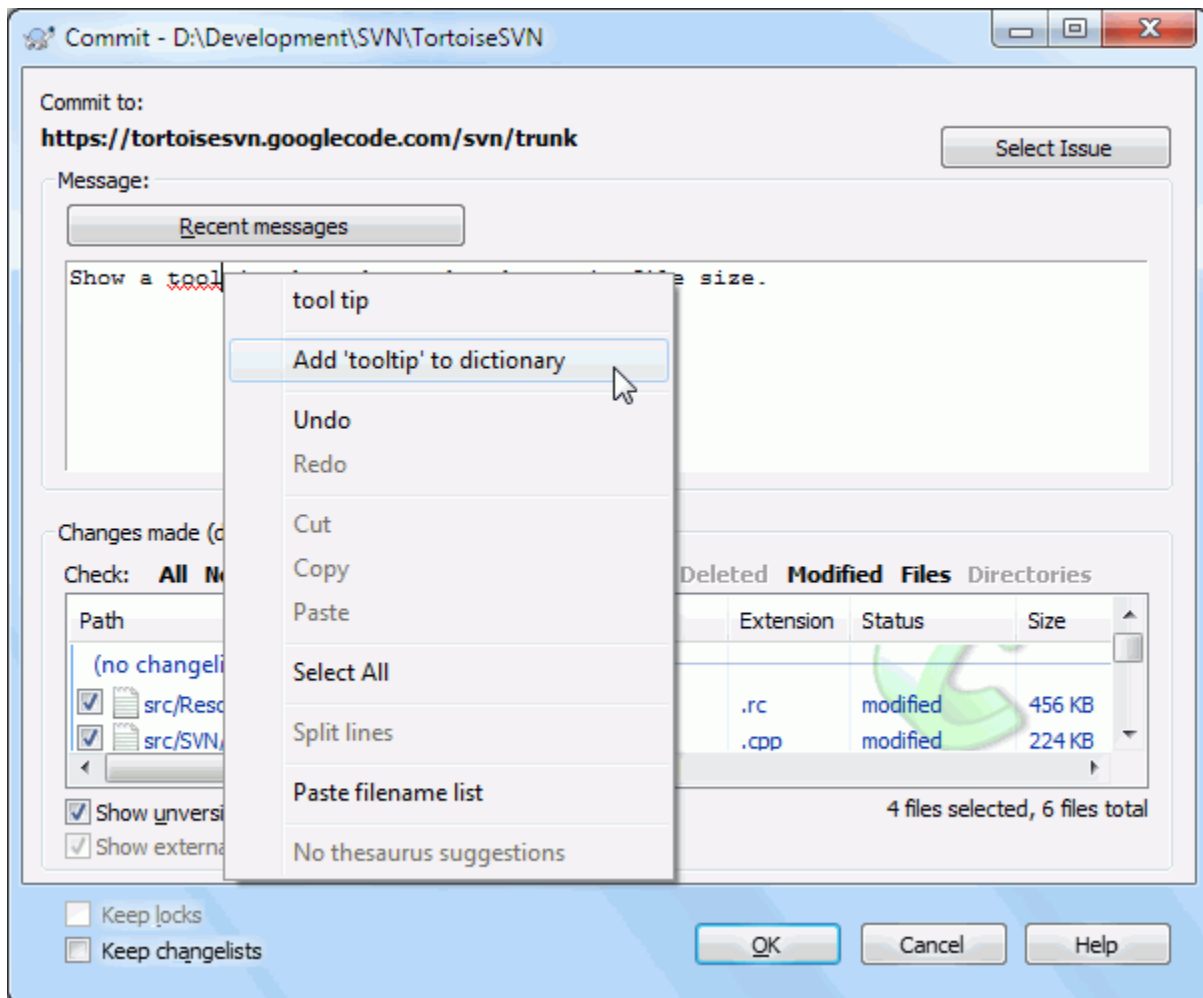
Soms heb je bestanden met versiebeheer die regelmatig wijzigen, maar die je eigenlijk niet wilt vastleggen. Dit kan er op duiden dat je een zwak punt heb in je bouwproces – waarom zitten die bestanden in versiebeheer? Zou je sjablonen moeten gebruiken? Soms is het gewoon onoverkomelijk. Een klassieke reden is dat je IDE de tijdstempel wijzigt elke keer als je het bouwproces hebt doorlopen. Het projectbestand moet wel in subverion komen, omdat het je instellingen bevat, maar het is niet nodig om deze elke keer vast te leggen als de alleen de tijdstempel is gewijzigd.

Om je uit dit soort vervelende situaties te helpen, hebben we een werkset gemaakt met de naam `ignore-on-commit` (negeren-bij-vastleggen). Elk bestand dat op deze lijst staat, zal automatisch gedeselecteerd worden in het vastleggen venster. Je kunt wijzigingen nog steeds wel vastleggen, maar je moet de bestanden dan handmatig selecteren in het vastleggen venster.

#### 4.4.5. Vastleg Log Meldingen

Zorg er voor dat je een log bericht invult, welke de wijzigingen beschrijft die je doorgevoerd hebt. Dit helpt om te bepalen wat er wanneer gewijzigd is als je op een later moment door de log berichten bladert. Het bericht kan zo kort of lang zijn als je wilt; bij veel project is gedefinieerd wat je er in moet zetten, welke taal je moet gebruiken en soms zelfs een strakke formattering.

Je kunt je tekst eenvoudig opmaken door gebruik te maken van stijlregels die lijken op de regels van emails. Om stijlen toe te passen op tekst, gebruikt `*tekst*` voor vet, `_tekst_` voor onderstrepen en `^tekst^` voor cursief.



**Afbeelding 4.8. De Vastleggen Dialoog Spellingscontrole**

TortoiseSVN bevat spellingscontrole om je te helpen berichten te maken zonder typerfouten. Verkeerd gespelde woorden worden gemarkeerd. Gebruik het context menu om de suggesties te bekijken. Natuurlijk is niet *technische* technische term die je kent bekend, waardoor correct gespelde woorden soms worden aangeduid als verkeerd gespeld. Maak je geen zorgen. Je kunt deze toevoegen aan je persoonlijke woordenboek via het context menu.

The log message window also includes a filename and function auto-completion facility. This uses regular expressions to extract class and function names from the (text) files you are committing, as well as the filenames themselves. If a word you are typing matches anything in the list (after you have typed at least 3 characters, or pressed **Ctrl+Space**), a drop-down appears allowing you to select the full name. The regular expressions supplied with TortoiseSVN are held in the TortoiseSVN installation `bin` folder. You can also define your own regexes and store them in `%APPDATA%\TortoiseSVN\autolist.txt`. Of course your private autolist will not be overwritten when you update your installation of TortoiseSVN. If you are unfamiliar with regular expressions, take a look at the introduction at [https://en.wikipedia.org/wiki/Regular\\_expression](https://en.wikipedia.org/wiki/Regular_expression), and the online documentation and tutorial at <http://www.regular-expressions.info/>.

Het kan best lastig zijn om de regex precies goed te krijgen. Om je te helpen bij het vinden van een passende expressie, is er een venster waarin je een expressie kunt typen en vervolgens de bestanden waarop deze getest moeten worden. Start deze vanaf een opdrachtprompt met het commando `<literal>TortoiseProc.exe /command:autotexttest</literal>`

The log message window also includes a commit message snippet facility. These snippets are shown in the autocomplete dropdown once you type a snippet shortcut, and selecting the snippet in the autocomplete dropdown then inserts the full text of the snippet. The snippets supplied with TortoiseSVN are held in the TortoiseSVN installation `bin` folder. You can also define your own snippets and store them in `%APPDATA%\TortoiseSVN`

\snippet.txt. # is the comment character. Newlines can be inserted by escaping them like this: \n and \r. To insert a backslash, escape it like this: \\.

Je kunt eerder gebruikte log berichten hergebruiken. Klik hiervoor op **Recente boodschappen**. Je kunt dan de laatste paar berichten die je gebruikt hebt bij deze werkkopie zien. Het aantal opgeslagen berichten kan aangepast worden in het TortoiseSVN instellingen venster.

Je kunt alle opgeslagen vastleggingsberichten wissen bij de **Opgeslagen gegevens** pagina van TortoiseSVN's instellingen. Je kunt ook individuele berichten wissen vanuit het **Recente boodschappen** venster door op de **Delete** knop te drukken.

Met het commando **Context Menu** → **Plak lijst bestandsnamen** kun je de gemarkeerde paden in het bericht invoegen.

Een andere methode voor het invoegen van de paden is door de bestanden te slepen van de bestandslijst naar het bewerk venster.



### Speciale Map Instellingen

Er zijn verschillende speciale mapeigenschappen, waarmee er meer controle gekregen wordt over de formattering van vastleggingsberichten en de taal van de spellingscontrole. Lees voor meer informatie [Paragraaf 4.18, "Project Instellingen"](#).

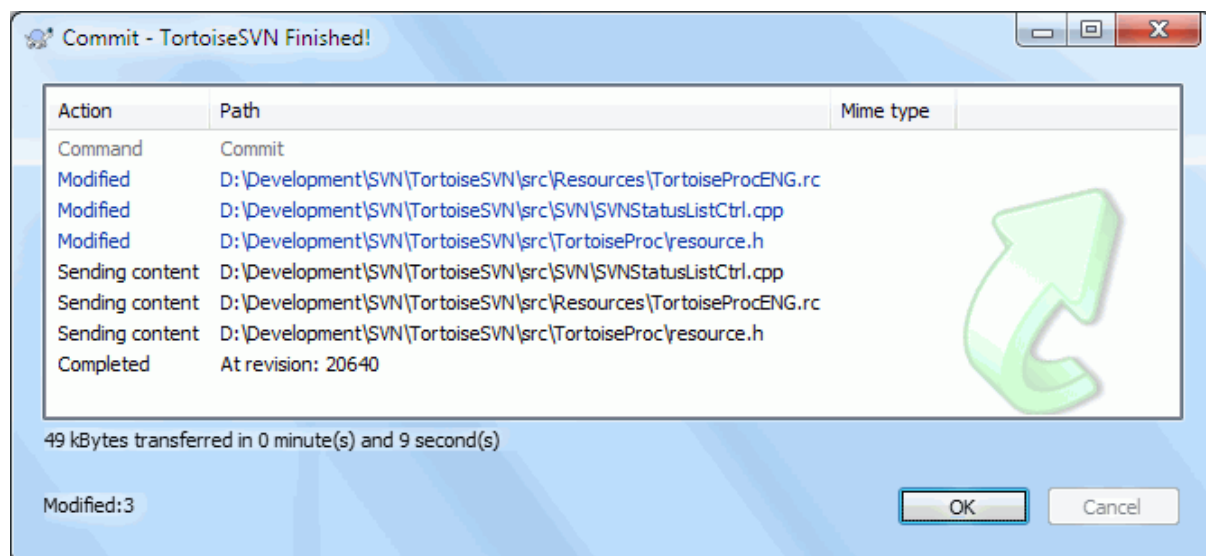


### Integratie met Bug Tracking Programma's

Als het bug tracking systeem hebt geactiveerd, dan kun je één of meerdere Problemen in het Bug-ID / Issue-Nr: tekstveld invullen. Meerdere problemen moeten gescheiden worden door komma's. Als je gebruik maakt van gebruikelijke expressies bij Bug tracking, voeg dan je referenties toe als onderdeel van het log bericht. Voor meer informatie: [Paragraaf 4.29, "Integratie met Bug Tracking Systemen / Issue Trackers"](#).

## 4.4.6. Vastlegging voortgang

Na het drukken op de OK knop zal een venster verschijnen met de voortgang van de vastlegging.



**Afbeelding 4.9. Het voortgangsvenster geeft de voortgang van een vastlegging weer**

Het voortgangsvenster gebruikt kleurcodering om de verschillende vastleggingsacties weer te geven.

Blauw

Vastleggen van een wijziging.

Paars

Een nieuwe toevoeging committen.

Donker rood

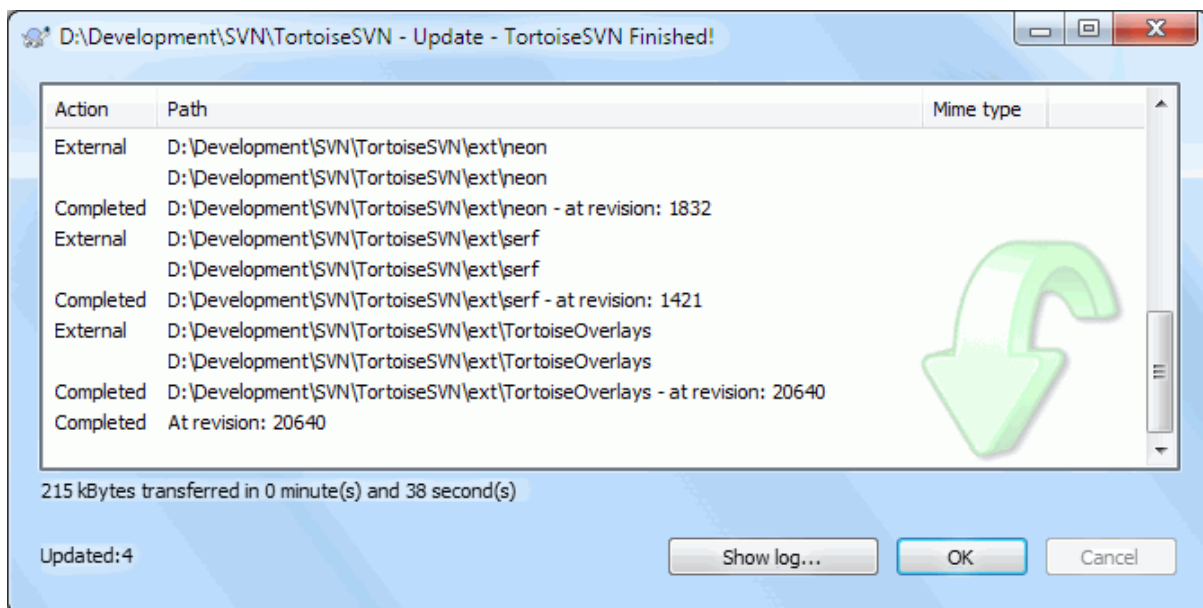
Vastleggen van een verwijdering of vervanging.

Zwart

Alle andere items.

Dit is het standaard kleurenschema, maar je kunt deze naar je eigen wensen aanpassen in het instellingen venster. Lees voor meer informatie [Paragraaf 4.31.1.5, "TortoiseSVN kleurinstellingen"](#).

## 4.5. Je Werkkopie Bijwerken Met De Wijzigingen Van Anderen



**Afbeelding 4.10. Voortgangsvenster van een afgeronde verversing**

Je moet er regelmatig voor zorgen dat wijzigingen die door anderen gemaakt zijn, in jouw werkkopie meegenomen worden. Het proces van het ophalen van de wijzigingen vanaf de server naar je lokale werkkopie staat bekend als *verversen*. Verversen kan op een enkel bestand, een aantal bestanden of recursief op een hele mapstructuur.

Om te verversen, selecteer de bestanden en mappen. Klik rechts en selecteer TortoiseSVN → Verversen in het verkenner context menu. Er zal een venster getoond worden die de voortgang van de verversing weergeeft. Wijzigingen van anderen worden in jouw bestanden verwerkt, zonder dat jouw wijzigingen in (mogelijk) dezelfde bestanden daarbij verloren gaan. Het archief wordt niet door het verversen bijgewerkt of veranderd.

Het voortgangsvenster gebruikt kleurcodering om verschillende acties te markeren.

Paars

Nieuw object toegevoegd aan je werkkopie.

Donker rood

Overtollig object verwijdert uit je werkkopie, of missend object vervangen in je werkkopie.

Groen

Wijzigingen uit het archief succesvol samengevoegd met je lokale wijzigingen.

Fel rood

Het samenvoegen van wijzigingen uit het archief met de lokale wijzigingen heeft geresulteerd in conflicten die opgelost moeten worden.



**Zwart**

Ongewijzigd object in je werkkopie bijgewerkt met de nieuwe versie uit het archief.

Dit is het standaard kleurenschema, maar je kunt deze naar je eigen wensen aanpassen in het instellingen venster. Lees voor meer informatie [Paragraaf 4.31.1.5, “TortoiseSVN kleurinstellingen”](#).

Als er tijdens het samenvoegen *conflicten* zijn opgetreden (als bijvoorbeeld anderen dezelfde regel op een andere manier hebben gewijzigd als jijzelf hebt gedaan), dan zal het venster deze met rood weergeven. Je kunt dubbel klikken op deze regels om het externe samenvoegprogramma te starten.

Als het verversen afgerond is, geeft het voortgangsvenster onder de bestandslijst een samenvatting weer van het aantal vernieuwingen, toevoegingen, verwijderingen, conflicten, etc.. Deze samenvatting kan naar het klembord gekopieerd worden met **Ctrl+C**.

The standard Update command has no options and just updates your working copy to the HEAD revision of the repository, which is the most common use case. If you want more control over the update process, you should use TortoiseSVN → Update to Revision... instead. This allows you to update your working copy to a specific revision, not only to the most recent one. Suppose your working copy is at revision 100, but you want it to reflect the state which it had in revision 50 - then simply update to revision 50.

In the same dialog you can also choose the *depth* at which to update the current folder. The terms used are described in [Paragraaf 4.3.1, “Ophaal diepte”](#). The default depth is Working copy, which preserves the existing depth setting. You can also set the depth *sticky* which means subsequent updates will use that new depth, i.e. that depth is then used as the default depth.

To make it easier to include or exclude specific items from the checkout click the Choose items... button. This opens a new dialog where you can check all items you want in your working copy and uncheck all the items you don't want.

You can also choose whether to ignore any external projects in the update (i.e. projects referenced using `svn:externals`).

**Let op**

Als je een bestand of map naar een bepaalde revisie ververst, dan moet je geen wijzigingen in die bestanden aanbrengen. Je krijgt dan een foutmelding als je je wijzigingen probeert vast te leggen dat je niet aan de laatste versie hebt gewerkt. Als je wijzigingen ongedaan wilt maken en opnieuw wilt starten vanaf een eerdere revisie, dan kun je dat doen via het revisie log venster. Lees [Paragraaf B.4, “Draai Revisies Terug in het Archief \(ongedaan maken\)”](#) voor meer instructies en alternatieve methodes.

Verversen naar Revisie kan soms handig om te bekijken hoe je project er een tijd geleden uitzag. Het verversen van individuele bestanden naar een eerdere revisie is zeker geen goed idee, omdat je werkkopie dan inconsistent wordt. Als de naam van een bestand gewijzigd is, dan zal dat bestand gewoon uit je werkkopie verdwijnen. Dit komt omdat er geen bestand met die naam in een eerdere revisie aanwezig was. Je moet je ook realiseren dat zo'n object (uit een eerdere revisie) een normaal overlappend pictogram heeft, waardoor het niet te onderscheiden is welke bestanden van de laatste revisie zijn.

Als je een lokale kopie wilt hebben van een oude versie van een bestand, dan kun je beter het Context Menu → Bewaar revisie in... commando gebruiken vanuit het log venster van het betreffende bestand.

**Meerdere Bestanden/Mappen**

Als je in de verkenner meerdere bestanden selecteert en vervolgens Verversen kiest, dan worden al die bestanden en mappen stuk voor stuk ververst. TortoiseSVN zorgt er voor dat bestanden en mappen uit hetzelfde archief bijgewerkt worden naar dezelfde revisie, zelfs als er gedurende het verversen een andere vastlegging heeft plaatsgevonden.



## 4.6. Conflicten oplossen

Zo af en toe zul je tegen een *conflict* aanlopen als je bestanden in het archief ververst of samenvoegt, of als je je werkkopie omschakelt naar een andere URL. Er zijn twee soorten conflicten:

### bestandsconflicten

Een bestandsconflict treedt op als er twee of meer ontwikkelaars dezelfde regel in één bestand hebben aangepast.

### mapstructuur conflicten

Een mapstructuur conflict treedt op als een ontwikkelaar een bestand of map verplaatst, hernoemd of verwijderd heeft, welke ook door een andere ontwikkelaar is verplaatst, hernoemd, verwijderd of gewijzigd is.

### 4.6.1. Bestandsconflicten

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. The conflicting area in a text file 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:

#### filename.ext.mine

Dit is het bestand zoals het aanwezig was in je werkkopie, voordat je je werkkopie ging verversen. Oftewel de versie zonder conflictmarkeringen. Dit bestand bevat jouw laatste wijzigingen en niets anders.

#### filename.ext.rOLDREV

Dit de BASE revisie van het is bestand voordat je je werkkopie ging verversen. Oftewel het bestand die je opgehaald hebt voordat je jouw laatste wijzigingen er in ging zetten.

#### filename.ext.rNEWREV

Dit is het bestand die zojuist door je Subversion applicatie is ontvangen van de server toen je jouw werkkopie ging verversen. Dit bestand komt overeen met de HEAD revisie in het archief.

You can either launch an external merge tool / conflict editor with TortoiseSVN → Edit Conflicts or you can use any text editor to resolve the conflict manually. You should decide what the code should look like, do the necessary changes and save the file. Using a merge tool such as TortoiseMerge or one of the other popular tools is generally the easier option as they generally present the files involved in a 3-pane view and you don't have to worry about the conflict markers. If you do use a text editor then you should search for lines starting with the string <<<<<<<.

Daarna moet je het commando TortoiseSVN → Opgelost kiezen en je wijzigingen vastleggen in het archief. Merk op dat het Opgelost commando het conflict zelf niet oplost. Het zorgt er alleen voor dat de bestanden filename.ext.mine en filename.ext.r\* worden verwijderd, zodat je je wijzigingen kunt vastleggen.

Subversion zal niet proberen om conflicten met binaire bestanden zelf op te lossen. Het lokale bestand blijft ongewijzigd (precies zoals je het als laatste had opgeslagen) en je krijgt filename.ext.r\* bestanden eraast. Als je je eigen wijzigingen wilt laten vervallen en de versie in het archief wilt behouden, gebruik dan het Terugzetten commando. Als je je eigen versie wilt gebruiken en de versie in het archief wilt overschrijven, gebruik dan het Opgelost commando en leg vervolgens je wijzigingen vast.

Het is mogelijk om het Opgelost commando op meerdere bestanden toe te passen. Klik met rechts op de bovenliggende map en selecteer TortoiseSVN → Opgelost... Er wordt dan een venster geopend met alle

conflicterende bestanden in die map. Je kunt dan selecteren welke bestanden gemarkeerd moeten worden als Opgelost.

## 4.6.2. Conflicten in eigenschappen

Een bestandsconflict treedt op als er twee of meer ontwikkelaars dezelfde regel in één bestand hebben aangepast. Zo'n conflict kan enkel maar opgelost worden door de ontwikkelaars.

If one of the changes must override the other then choose the option to **Resolve using local property** or **Resolve using remote property**. If the changes must be merged then select **Manually edit property**, sort out what the property value should be and mark as resolved.

## 4.6.3. Mapstructuur conflicten

Een conflict in mapstructuur ontstaat als een ontwikkelaar bestanden of mappen verplaatst, hernoemd of verwijderd heeft, die een andere ontwikkelaar toevallig ook verplaatst, hernoemd, verwijderd of gewijzigd heeft. Er zijn nog veel meer situaties die tot een mapstructuur conflict kunnen leiden en elke situatie heeft weer een andere oplossing nodig.

Als een bestand lokaal uit Subversion verwijderd is, dan wordt deze ook van het lokale bestandssysteem verwijderd. Dat houdt in dat er geen overlappend pictogram weergegeven kan worden als dit bestand onderdeel is van een mapstructuurconflict en je er dus ook niet met rechts op kunt klikken om het probleem op te lossen. Gebruik dan het **Kijk of er updates zijn** venster om de **Conflicten oplossen** optie te kunnen benaderen.

TortoiseSVN kan helpen om de beste plaatst voor het samenvoegen van wijzigingen te vinden, maar voor het oplossen van conflicten is er toch meer handwerk nodig. Onthoudt dat na een verversing de BASE altijd de revisie van bestanden uit het archief bevat van het moment van verversen. Als je een wijziging terugzet nadat je ververst hebt, dan kom je terug op de stand van het archief, niet naar de stand die je toen je begon met het wijzigen.

### 4.6.3.1. Lokale verwijdering, inkomende wijziging bij verversing

1. Ontwikkelaar A wijzigt `Foo.c` en verstuurt deze naar de repository.
2. Ontwikkelaar B heeft tegelijkertijd `Foo.c` naar `Bar.c` verplaatst in zijn werkkopie of heeft `Foo.c` of de bovenliggende map verwijderd.

Een verversing van de werkkopie van ontwikkelaar B resulteert in een mapstructuur conflict:

- `Foo.c` is verwijderd uit de werkkopie, maar deze is gemarkeerd als een mapstructuur conflict.
- Als het conflict veroorzaakt is door een hernoeming in plaats van verwijdering, dan zal `Bar.c` gemarkeerd worden als toegevoegd, maar deze zal dan niet de wijzigingen van ontwikkelaar A bevatten.

Ontwikkelaar B moet nu kiezen of hij de wijzigingen van ontwikkelaar A wil behouden. Als het om een naamswijziging gaat, dan kan hij de wijzigingen van `Foo.c` samenvoegen in het naar `Bar.c` hernoemde bestand. Bij eenvoudige bestands- of mapverwijderingen kan hij er voor kiezen om het object met de wijzigingen van ontwikkelaar A te behouden en de verwijdering te negeren. Of, door middel van het als Opgelost markeren van de conflicten zonder iets gewijzigd te hebben, worden de wijzigingen van ontwikkelaar A volledig genegeerd.

The conflict edit dialog offers to merge changes if it can find the original file of the renamed `Bar.c`. If there are multiple files that are possible move sources, then a button for each of these files is shown which allow you to chose the correct file.

### 4.6.3.2. Lokale wijziging, inkomende verwijdering bij verversing

1. Ontwikkelaar A hernoemt `Foo.c` naar `Bar.c` en legt de wijziging vast in het archief.
2. Ontwikkelaar B wijzigt `Foo.c` in zijn werkkopie.

Of bij een verplaatsing van een map ...

1. Ontwikkelaar A hernoemt hoofdmap `FooFolder` in `BarFolder` en legt de wijziging vast in het archief.

2. Ontwikkelaar B wijzigt `Foo.c` in zijn werkkopie.

Een verversing van de werkkopie van ontwikkelaar B leidt tot een mapstructuur conflict. Een eenvoudig bestandsconflict:

- `Bar.c` wordt toegevoegd aan de werkkopie als een normaal bestand.
- `Foo.c` wordt als toegevoegd (inclusief geschiedenis) gemarkeerd en heeft een mapstructuur conflict.

Een mapstructuur conflict:

- `BarFolder` wordt aan de werkkopie toegevoegd als een normale map.
- `FooFolder` wordt gemarkeerd als toegevoegd (inclusief geschiedenis) en heeft een mapstructuurconflict.

`Foo.c` wordt gemarkeerd als gewijzigd.

Ontwikkelaar B moet nu de keuze maken of deze meegaat met de herindeling van ontwikkelaar A. Als ontwikkelaar B daarin meegaat, kan hij zijn wijzigingen in het nieuwe bestand samenvoegen. Zo niet, dan kan ontwikkelaar B de wijzigingen van A terugdraaien en zijn eigen lokale versie behouden.

To merge her local changes with the reshuffle, Developer B must first find out to what filename the conflicted file `Foo.c` was renamed/moved in the repository. This can be done by using the log dialog. Then use the button which shows the correct source file to resolve the conflict.

If Developer B decides that A's changes were wrong then she must choose the **Mark as resolved** button in the conflict editor dialog. This marks the conflicted file/folder as resolved, but Developer A's changes need to be removed by hand. Again the log dialog helps to track down what was moved.

#### 4.6.3.3. Lokale verwijdering, inkomende verwijdering bij verversing

1. Ontwikkelaar A hernoemt `Foo.c` naar `Bar.c` en legt de wijziging vast in het archief.
2. Ontwikkelaar B hernoemt `Foo.c` naar `Bix.c`

Een verversing van de werkkopie van ontwikkelaar B resulteert in een mapstructuur conflict:

- `Bix.c` wordt gemarkeerd als toegevoegd inclusief geschiedenis.
- `Bar.c` is toegevoegd aan de werkkopie met 'normale' status.
- `Foo.c` wordt gemarkeerd als verwijderd en leidt tot een mapstructuur conflict.

Om dit conflict op te lossen, moet ontwikkelaar B uitzoeken welke bestandsnaam de conflicterend `Foo.c` heeft gekregen in het archief. Het log venster kan hiervoor gebruikt worden.

Ontwikkelaar B moet dan bepalen welke bestandsnaam van `Foo.c` behouden moet worden, degene van ontwikkelaar A of de zelf gekozen nieuwe bestandsnaam.

Wanneer ontwikkelaar B het conflict handmatig heeft opgelost, dan moet het conflict als opgelost worden gemarkeerd in het conflict bewerkingsvenster.

#### 4.6.3.4. Lokaal missend, binnenkomende wijziging bij het samenvoegen

1. Ontwikkelaar A werkt in de basislijn en wijzigt `Foo.c` en legt de wijzigingen vast in het archief.
2. Ontwikkelaar B werkt aan een branch en hernoemt `Foo.c` in `Bar.c` en legt zijn wijzigingen vast in het archief.

Het samenvoegen van de wijzigingen van A op de basislijn naar de tak-werkkopie van ontwikkelaar B leidt tot een mapstructuur conflict:

- `Bar.c` bevindt zich al in de werkkopie met status 'normaal'.
- `Foo.c` wordt gemarkeerd als missend met een mapstructuur conflict.

Ontwikkelaar B moet het bestand als opgelost markeren in het conflict bewerkingsvenster, om ervoor te zorgen dat het bestand van de conflict lijst wordt gehaald. Vervolgens moet besloten worden of het missende bestand `Foo.c` vanuit het archief naar de werkkopie gekopieerd moet worden, of dat de wijzigingen van ontwikkelaar A in `Foo.c` samengevoegd moeten worden in `Bar.c`, of de wijzigingen te negeren door het conflict als opgelost te markeren zonder iets gewijzigd te hebben.

Merk op dat als je het missende bestand eerst uit het archief kopieert en het dan als opgelost markeert, dan je kopie meteen weer verwijderd wordt. Je moet eerst het conflict oplossen en daarna pas kopiëren.

#### 4.6.3.5. Lokale wijziging, inkomende verwijdering bij samenvoegen

1. Ontwikkelaar A werkt aan de basislijn en verplaatst `<filename>Foo.c</filename>` naar `<filename>Bar.c</filename>` en legt de wijziging vast in het archief.
2. Ontwikkelaar B werkt aan een vertakking (branch), wijzigt `Foo.c` en legt de wijziging vast in het archief.
1. Ontwikkelaar A werkt aan de basislijn en hernoemt de bovenliggende map `FooFolder` in `BarFolder` en legt de wijziging vast in het archief.
2. Ontwikkelaar B werkt aan de vertakking (branch) en wijzigt `Foo.c` in de werkkopie.

Het samenvoegen van de wijzigingen van A op de basislijn naar de tak-werkkopie van ontwikkelaar B leidt tot een mapstructuur conflict:

- `Bar.c` is gemarkeerd als toegevoegd.
- `Foo.c` is gemarkeerd als gewijzigd en met een mapstructuur conflict.

Ontwikkelaar B moet nu de keuze maken of deze meegaat met de herindeling van ontwikkelaar A. Als ontwikkelaar B daarin meegaat, kan hij zijn wijzigingen in het nieuwe bestand samenvoegen. Zo niet, dan kan ontwikkelaar B de wijzigingen van A terugdraaien en zijn eigen lokale versie behouden.

To merge her local changes with the reshuffle, Developer B must first find out to what filename the conflicted file `Foo.c` was renamed/moved in the repository. This can be done by using the log dialog for the merge source. The conflict editor only shows the log for the working copy as it does not know which path was used in the merge, so you will have to find that yourself. The changes must then be merged by hand as there is currently no way to automate or even simplify this process. Once the changes have been ported across, the conflicted path is redundant and can be deleted.

If Developer B decides that A's changes were wrong then she must choose the **Mark as resolved** button in the conflict editor dialog. This marks the conflicted file/folder as resolved, but Developer A's changes need to be removed by hand. Again the log dialog for the merge source helps to track down what was moved.

#### 4.6.3.6. Lokale verwijdering, inkomende verwijdering bij samenvoegen

1. Ontwikkelaar A werkt aan de basislijn en verplaatst `<filename>Foo.c</filename>` naar `<filename>Bar.c</filename>` en legt de wijziging vast in het archief.
2. Developer B working on a branch moves `Foo.c` to `Bix.c` and commits it to the repository.

Het samenvoegen van de wijzigingen van A op de basislijn naar de tak-werkkopie van ontwikkelaar B leidt tot een mapstructuur conflict:

- `Bix.c` is gemarkeerd met de normale (ongewijzigde) status.
- `Bar.c` is gemarkeerd als toegevoegd inclusief geschiedenis.
- `Foo.c` is gemarkeerd als missend en heeft een mapstructuur conflict.

To resolve this conflict, Developer B has to find out to what filename the conflicted file `Foo.c` was renamed/moved in the repository. This can be done by using the log dialog for the merge source.

Ontwikkelaar B moet dan bepalen welke bestandsnaam van `Foo.c` behouden moet worden, degene van ontwikkelaar A of de zelf gekozen nieuwe bestandsnaam.

Wanneer ontwikkelaar B het conflict handmatig heeft opgelost, dan moet het conflict als opgelost worden gemarkeerd in het conflict bewerkingsvenster.

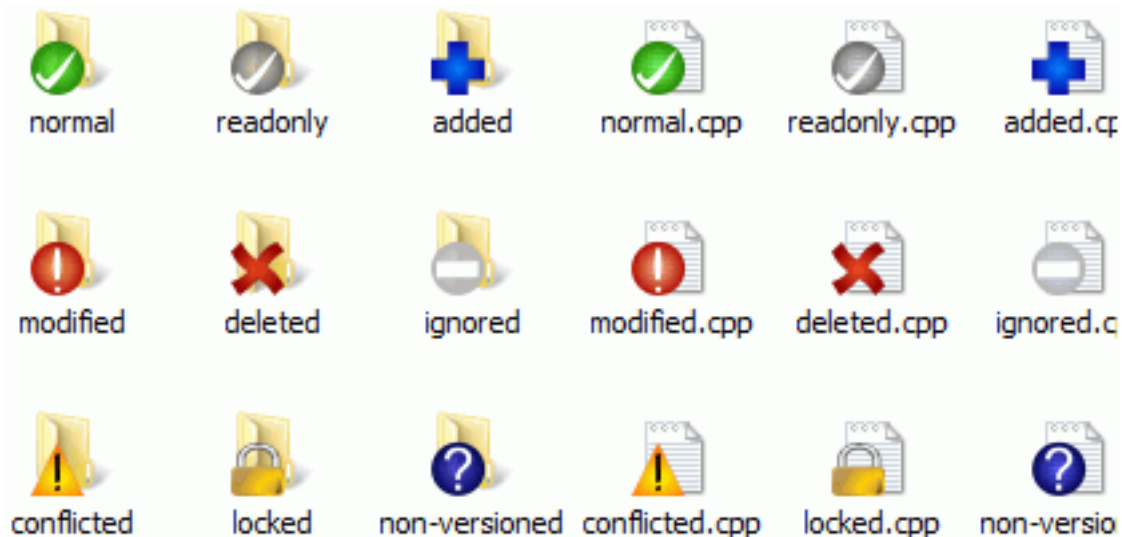
#### 4.6.3.7. Andere mapstructuur conflicten

Er zijn nog enkele andere situaties welke worden aangeduid als mapstructuur conflicten, dit komt doordat bij de conflicten mappen in plaats van bestanden betrokken zijn. Als je bijvoorbeeld een map met dezelfde naam in de basislijn en tak toevoegd en deze vervolgens probeert samen te voegen, dan zal er een mapstructuur conflict ontstaan. Als je de map van het samenvoeg doel wilt gebruiken, markeer het conflict dan als opgelost. Als je de map uit de samenvoeg bron wilt gebruiken, dan moet je de map in de doel structuur er met SVN Verwijderen en vervolgens de samenvoeging opnieuw starten. Als je een ingewikkeldere oplossing wilt hebben, dan zul je dat handmatig moeten doen.

## 4.7. Statusinformatie ophalen

Terwijl je aan je werkkopie werkt, wil je vaak weten welke bestanden je gewijzigd, toegevoegd, verwijderd of hernoemt hebt, maar ook welke bestanden gewijzigd en vastgelegd zijn door anderen.

### 4.7.1. Overlappende Pictogrammen



**Afbeelding 4.11. Overlappende pictogrammen weergegeven in de verkenner**

Nu je een werkkopie hebt opgehaald uit het Subversion archief, kun je de bestanden zien in de Windows verkenner met de gewijzigde pictogrammen/iconen. Dit is één van de redenen waarom TortoiseSVN zo populair is. TortoiseSVN voegt zogenaamde overlay iconen, dat zijn iconen die over de normale iconen geprojecteerd worden, toe aan de originele iconen. Afhankelijk van de status in Subversion wordt telkens een ander pictogram weergegeven.



Een verse werkkopie opgehaald uit het archief heeft een groen vinkje als overlappend pictogram. Dit betekent dat dit object de Subversion status *normaal* heeft.



Zodra je een bestand gewijzigd en opgeslagen hebt, wordt de status gewijzigd in *gewijzigd* en zal het overlappende pictogram wijzigen in een rood pictogram met een uitroepteken. Op deze manier kun je makkelijk zien welke bestanden gewijzigd zijn sinds de laatste keer dat je je werkkopie ververst hebt.



Als bij het verversen zich een *conflict* voordoet, dan zal er een geel pictogram met een uitroepteken te zien zijn.



Als je de `svn:needs-lock` eigenschap ingesteld hebt op een bestand, dan zal Subversion dat bestand als alleen-lezen instellen totdat het bestand voor jou vergrendeld is. Dergelijke bestanden krijgen deze pictogram om aan te geven dat je het bestand eerst moet vergrendelen, voordat je het kunt wijzigen.



Als je een bestand vergrendeld hebt en de Subversion status is *normaal*, dan herinnert dit overlappend pictogram je eraan dat je het bestand nog vrij moet geven als je anderen de mogelijkheid wilt geven om daarin wijzigingen vast te laten leggen.



Dit pictogram geeft aan dat bestanden of mappen in de huidige map gemarkeerd zijn om te *verwijderen* uit het archief of dat een bestand in het archief mist in de folder van je werkkopie.



Het plusteken geeft aan dat het bestand of de map van versiebeheer voorzien moet worden en dus aan het archief *toegevoegd* moet worden.



Het balk-teken geeft aan dat een bestand of map *genegeerd* wordt voor versiebeheer. Dit pictogram is optioneel.



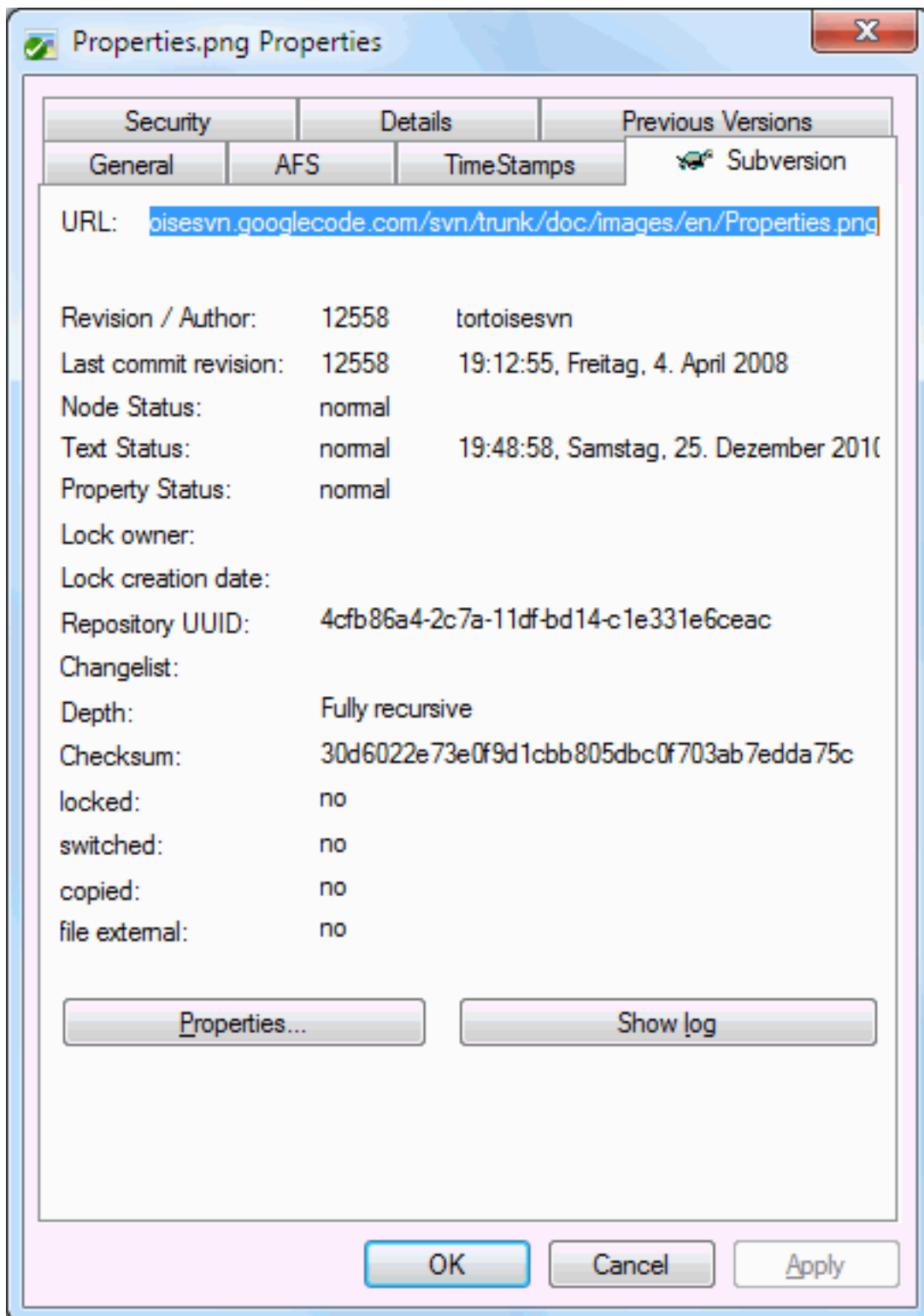
Dit pictogram geeft de bestanden en mappen aan die niet van versiebeheer voorzien zijn, maar ook niet genegeerd worden. Dit pictogram is optioneel.

In de praktijk zul je zien dat niet alle pictogrammen gebruikt zullen worden op jouw systeem. Dit komt doordat Windows maar een beperkt aantal overlappende pictogrammen toestaat. Als je ook nog een oude versie van TortoiseCVS gebruikt, dan zijn er niet genoeg sloten voor overlappende pictogrammen beschikbaar. TortoiseSVN probeert een goede burger (“Good Citizen (TM)”) te zijn en beperkt het gebruik van overlappende pictogrammen om andere applicaties ook een kans te geven.

Now that there are more Tortoise clients around (TortoiseCVS, TortoiseHg, ...) the icon limit becomes a real problem. To work around this, the TortoiseSVN project introduced a common shared icon set, loaded as a DLL, which can be used by all Tortoise clients. Check with your client provider to see if this has been integrated yet :-)

Een beschrijving van de overlappende pictogrammen en hoe deze zich met de Subversion status en andere technische details verhouden kun je vinden in [Paragraaf F.1, “Overlappende Pictogrammen”](#).

#### 4.7.2. Gedetailleerde status

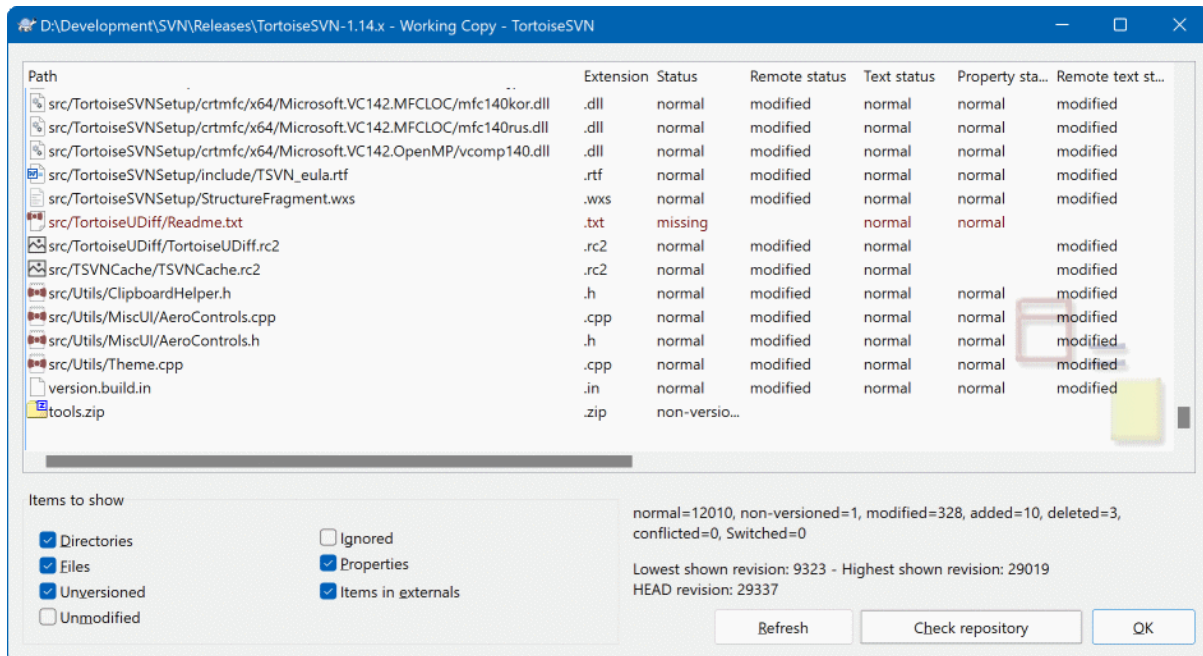


**Afbeelding 4.12. Verkenner eigenschappen pagina, Subversion tab**

In sommige gevallen wilt u meer gedetailleerde informatie over een bestand/map dan het overlappende pictogram. U kunt alle informatie welke Subversion levert verkrijgen in het eigenschappen dialoog van de verkenner. Selecteer het bestand of directory en selecteer Windows Menu → eigenschappen in het contextmenu (noot: dit is het normale eigenschappen menu-optie welke de verkenner levert, niet degene in het TortoiseSVN submenu!). In het

eigenschappen dialoogvenster heeft TortoiseSVN een nieuwe eigenschappen pagina toegevoegd, voor bestanden/folders welke onder Subversion controle vallen, waar u alle relevante informatie over het geselecteerde bestand/directorie kunt vinden.

### 4.7.3. Lokale en Server Status



#### Afbeelding 4.13. Controleer op Wijzigingen

Het is vaak handig om te weten welke bestanden anderen gewijzigd en vastgelegd hebben. Hiervoor is het commando TortoiseSVN → Kijk of er updates zijn... erg handig. Het venster zal elk bestand weergeven dat op welke manier dan ook gewijzigd is in je werkkopie en ook de bestanden zonder versiebeheer.

Als je op de Controleer Archief klikt, dan kun je ook de wijzigingen in het archief bekijken. Hiermee kun je, voordat je je wijzigingen vastlegt, nagaan of er conflicten zullen ontstaan. Je kunt ook alleen geselecteerde bestanden verversen, zonder dat de hele map verversed hoeft te worden. De Controleer Archief haalt standaard de status op van de werkkopie met de instelling van mapdiepte van de werkkopie. Als je alle bestanden in het archief wilt zien, ook degene die je niet opgehaald hebt, dan moet je de **Shift** ingedrukt houden als je op de knop Controleer Archief drukt.

In dit venster worden kleurcoderingen toegepast om de status weer te geven.

#### Blauw

Lokaal gewijzigde objecten.

if unchanged files are inside a directory that's been moved, the status will show a + sign in the status column, and it will be colored in blue as well.

#### Paars

Toegevoegde objecten. Objecten die toegevoegd zijn met geschiedenisinformatie, hebben een + teken in de Tekst status kolom en in tooltip wordt aangegeven waar heb object vandaan gekopieerd is.

#### Donker rood

Verwijderde of missende objecten.

#### Groen

Objecten die lokaal en in het archief gewijzigd zijn. De wijzigingen zullen samengevoegd worden bij het verversen, waarbij conflicten *kunnen* ontstaan.



#### Fel rood

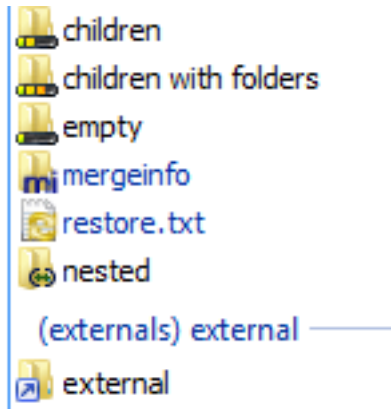
Objecten die lokaal gewijzigd zijn en verwijderd uit het archief, of die gewijzigd zijn in het archief en lokaal verwijderd. Hierdoor *zal* een conflict ontstaan bij het verversen.

#### Zwart

Ongewijzigde objecten en objecten zonder versiebeheer.

Dit is het standaard kleurenschema, maar je kunt deze naar je eigen wensen aanpassen in het instellingen venster. Lees voor meer informatie [Paragraaf 4.31.1.5, "TortoiseSVN kleurinstellingen"](#).

Overlay icons are used to indicate other states as well. The screenshot below shows all the possible overlays that are shown if necessary.



Overlays are shown for the following states:

- Checkout depth `empty`, meaning only the item itself.
- Checkout depth `files`, meaning only the item itself and all file children without child folders.
- Checkout depth `immediates`, meaning only the item itself and all file and folder children, but without children of the child folders.
- Nested items, i.e., working copies inside the working copy.
- External items, i.e., all items that are added via an `svn:externals` property.
- Items that are restored after a commit. See [Paragraaf 4.4.3, "Alleen delen van bestanden inchecken"](#) for details.
- Items that have property modifications, but only to the `svn:mergeinfo` property. If any other property is modified, the overlay is not used.

Items which have been switched to a different repository path are also indicated using an `(s)` marker. You may have switched something while working on a branch and forgotten to switch back to trunk. This is your warning sign! The context menu allows you to switch them back to the normal path again.

Vanuit het context menu van het venster kun je de verschillen van de wijzigingen opvragen. Controleer de wijzigingen die *jij* lokaal gemaakt hebt door te kiezen voor **Context Menu** → **Vergelijk met basis**. Om de wijzigingen te bekijken die anderen in het archief hebben opgeslagen kies je **Context Menu** → **Toon verschillen als unified diff**.

Het is ook mogelijk om wijzigingen in individuele bestanden terug te draaien. Als je per ongeluk een bestand verwijderd hebt, dan zal deze weergegeven worden als *Ontbrekend*. Je kunt deze dan terughalen met *Maak ongedaan*.

Bestanden zonder versiebeheer of bestanden die op de negeren lijst staan, kunnen verwijderd worden (naar de prullenbak) van de lokale kopie door de optie **Context Menu** → **Verwijderen** te gebruiken. Als je bestanden permanent wilt verwijderen (zonder ze naar de prullenbak te doen), dan moet je **Shift** ingedruukt houden als je op **Verwijderen** klikt.

If you want to examine a file in detail, you can drag it from here into another application such as a text editor or IDE, or you can save a copy simply by dragging it into a folder in explorer.

Je kunt de kolommen aanpassen. Als je met rechts klikt op de koptekst van een kolom, dan krijg je een context menu te zien waarmee je kunt aangeven welke kolommen je wilt zien. De breedte van de kolommen kun je aanpassen door de kolomovergangen te verslepen. De aanpassingen worden opgeslagen, zodat deze er een volgende keer net zo uit zien.

Je kunt groepen bestanden in een wijzigingenlijst opnemen. Dit is handig als je aan meerdere taken, die niets met elkaar te maken hebben, tegelijk werkt. Lees hiervoor [Paragraaf 4.4.2, "Werksets"](#).

Onderaan het scherm kun je zien welke reeks van archiefrevisies er in je werkkopie gebruikt worden. Dit zijn de revisies van de *vastlegging*, niet van de laatste *verversing*; dit geeft de reeks van revisies weer van de laatste vastleggingen, niet van de laatste verversing. Merk op dat de reeks van revisies alleen slaat op de objecten die weergegeven zijn in het venster, niet op je hele werkkopie. Als je deze informatie van je hele werkkopie wilt zien, dan moet je het selectievak Niet gewijzigde bestanden weergeven selecteren.



### Tip

Als je een platte weergave wilt hebben van je werkkopie, oftewel alle bestanden van alle niveau's uit je mapstructuur, dan kun je daarvoor het beste de *Kijk of er updates zijn...* functie gebruiken. Zet de optie *Ongewijzigde bestanden weergeven* aan om alle bestanden in je werkkopie te zien.



### Externe hernoemingen repareren

Soms worden bestanden hernoemt buiten Subversion, waardoor deze bestanden in de lijst weergegeven worden als missend of zonder versiebeheer. Om te voorkomen dat je de geschiedenis verliest, moet je Subversion vertellen dat er een verbinding is. Selecteer de oude (missende) naam en de nieuwe naam en gebruik Context Menu → *Repareer Verplaatsing* om de bestanden aan elkaar te koppelen als hernoemden.



### Herstellen van Bestandhernoemingen bij externe kopieën

Als je een kopie van een bestand hebt gemaakt en vergeten bent om het Subversion commando te gebruiken, dan kun je dat repareren om te ervoor te zorgen dat de geschiedenis niet verloren gaat. Selecteer de oude (normaal of gewijzigde) en de nieuwe naam (zonder versiebeheer) en gebruik Context Menu → *Repareer Kopieer-actie* om de twee bestanden aan elkaar te koppelen als kopie.

## 4.7.4. Diffs bekijken

Vaak wil je de inhoud van bestanden even bekijken, om te bekijken wat je precies gewijzigd hebt. Dit kun je doen door een gewijzigd bestand te selecteren en *Vergelijk* in TortoiseSVN's context menu te selecteren. Hiermee wordt het externe vergelijk-programma gestart, waarmee het huidige bestand vergeleken zal worden met de ongewijzigde kopie (BASE revisie) die opgeslagen was na de laatste verversing.



### Tip

Zelfs bestanden die niet in je werkkopie zitten of als je verschillende versies verspreid opgeslagen hebt, kun je verschillen bekijken:

Selecteer in de verkenner de twee bestanden die je wilt vergelijken (bijvoorbeeld met **Ctrl** en de muis) en selecteer vanuit TortoiseSVN's context menu *Vershil*

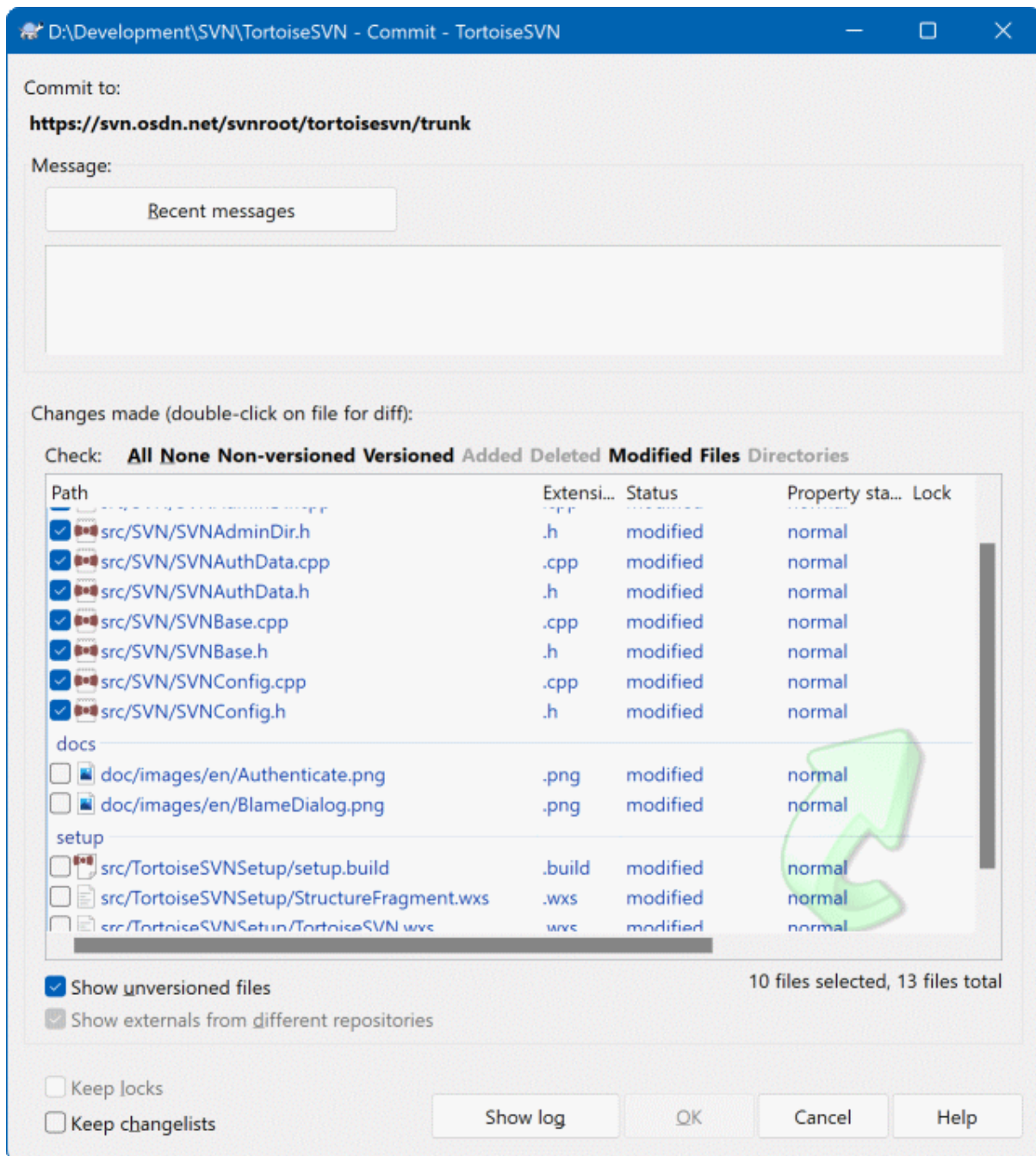
## 4.8. Werksets

In de ideale wereld werk je maar aan één ding tegelijk en zal je werkkopie alleen maar logische wijzigingen bevatten. Ok, dan terug naar de werkelijkheid. Het komt vaak voor dat je aan verschillende taken, die niets met elkaar te maken hebben, tegelijk moet werken. Als je dan naar het vastleggen venster kijkt, staan alle wijzigingen door elkaar. De *werkset* functie helpt je met het groeperen van bestanden, waardoor je een beter overzicht krijgt waar je mee bezig bent. Dit werkt natuurlijk alleen als de wijzigingen elkaar niet overlappen. Als bij twee verschillende taken hetzelfde bestand gewijzigd wordt, dan is er geen methode om die wijzigingen van elkaar te scheiden.

Je kunt werksets vinden op verschillende plaatsen, maar de belangrijkste zijn die in het vastleggen venster en kijk-of-er-updates-zijn venster. Laten we beginnen met het kijk-of-er-updates-zijn venster nadat je aan verschillende functies en meerdere bestanden hebt gewerkt. Als je het venster voor het eerst opent, dan worden alle bestanden in de lijst weergegeven. Stel voor dat je deze lijst nu wilt organiseren en de bestanden wilt groeperen.

Selecteer één of meer bestanden en gebruik de functie **Context Menu** → **Verplaats naar werkset** om een object aan de werkset toe te voegen. In het begin zullen er geen werksets bestaan, waardoor er bij de eerste keer dat je dit doet een nieuwe werkset aangemaakt zal worden. Geef de werkset een naam die de lading dekt en klik op **OK**. Vervolgens zal de groepen met objecten weergegeven worden.

Als je eenmaal een werkset hebt aangemaakt, dan kun je er objecten naartoe slepen. Dit kan vanuit een andere werkset of vanuit de Windows Verkenner. Het verslepen vanuit de Verkenner is handig, omdat je objecten al aan de werkset kunt toevoegen nog voordat deze gewijzigd zijn. Dit kun je ook wel doen vanuit het kijk-of-er-updates-zijn venster, maar dan moet je alle bestanden zonder versiebeheer laten weergeven.



#### Afbeelding 4.14. Vastleggen venster met werksset

In het vastleggen venster kun je de bestanden dan zien, maar dan gegroepeerd per werksset. Buiten dat je een directe visuele groepering van de bestanden ziet, kun je ook nog de groepen van een kop voorzien waarmee je de set van bestanden voor het vastleggen kunt selecteren.

TortoiseSVN heeft één werksset gereserveerd voor eigen gebruik, dat is de negeren-bij-vastleggen set. Deze wordt gebruikt voor het markeren van bestanden met versiebeheer, die je bijna nooit wilt vastleggen ook al zijn er lokale wijzigingen. Deze functie wordt beschreven in [Paragraaf 4.4.4, "Objecten van de vastleggingslijst uitsluiten"](#).

Als je bestanden vastlegt die opgenomen zijn in een werksset, dan zou je verwachten dat de koppeling aan de werksset niet langer nodig is. Normaal gesproken worden bestanden automatisch uit een werksset verwijderd als je je wijzigingen vastlegt. Als je het bestand in de werksset wilt behouden, gebruik dan de optie **Werksets bewaren** onderaan het vastleggen venster.



### Tip

Werksets is een lokale functie. Het aanmaken en verwijderen van werksets zal het archief niet veranderen, noch de werkkopie van iemand anders. Het is een handig manier voor je om je bestanden te organiseren.



### Waarschuwing

Note that if you use changelists, externals will no longer show up in their own groups anymore. Once there are changelists, files and folders are grouped by changelist, not by external anymore.

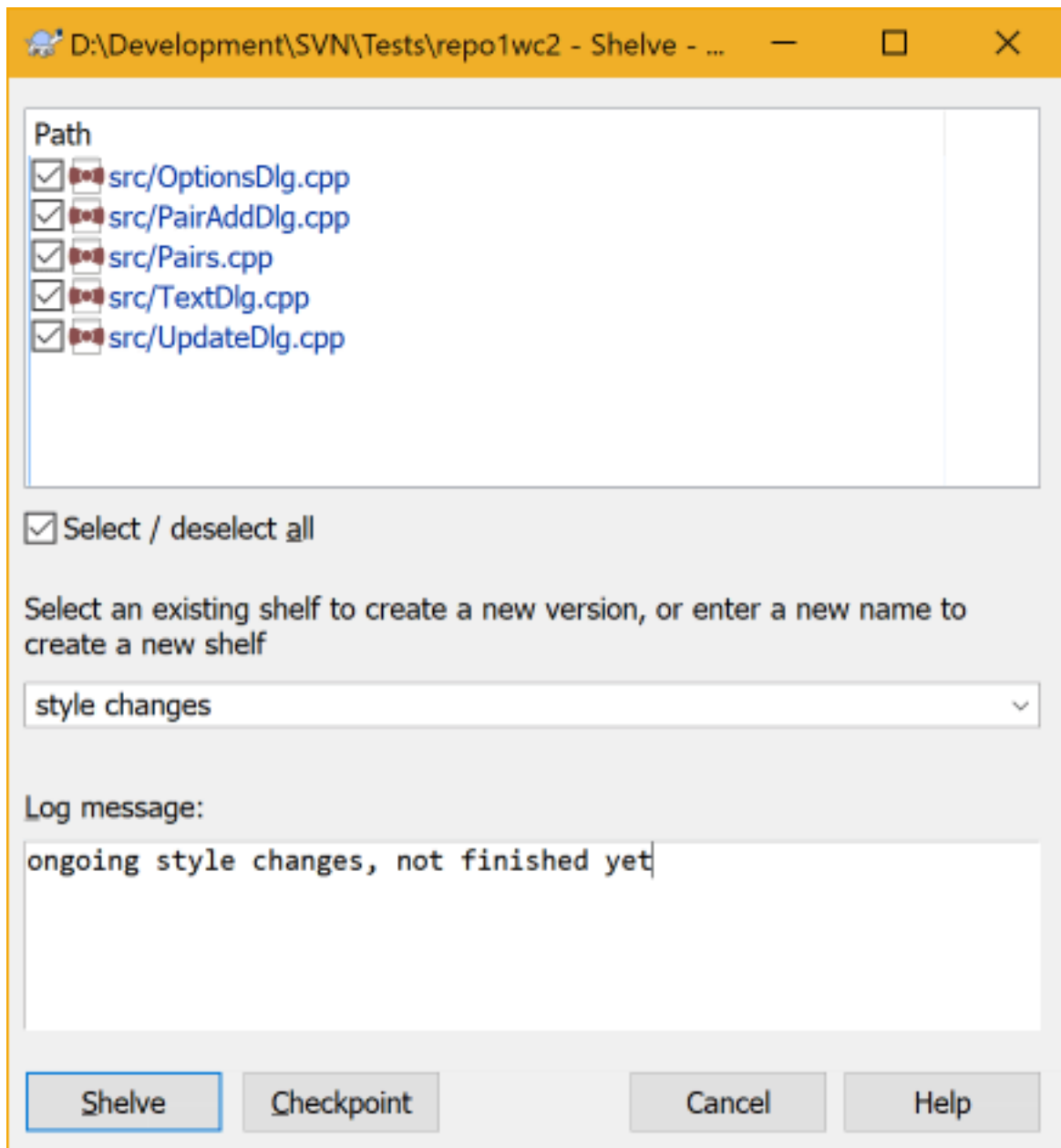
## 4.9. Shelving

More often than wanted, it's necessary to stop what you were working on and work on something else. For example a serious problem needs immediate dealing with and you have to stop working on the new feature. If possible, you should commit the changes you have done so far and then start working on the urgent issue, but often those changes would break the build or are just not ready for committing yet.

So if you can't commit your local changes yet, you have to put them aside while you're working on the urgent issue. The *shelving* feature helps you do exactly that: you can store your local changes on a shelf, get your working copy in a clean state again and work on the issue. After you're finished with the urgent issue and you've committed those changes, you can *unshelve* your shelved work and continue working on your previous task again.

Two new commands are implemented for this. One for shelving and one for unshelving.

To shelve your local changes, select your working copy and use Context Menu → Shelve The following dialog allows you to select the files you want to shelve and give a name under which you want to store them.

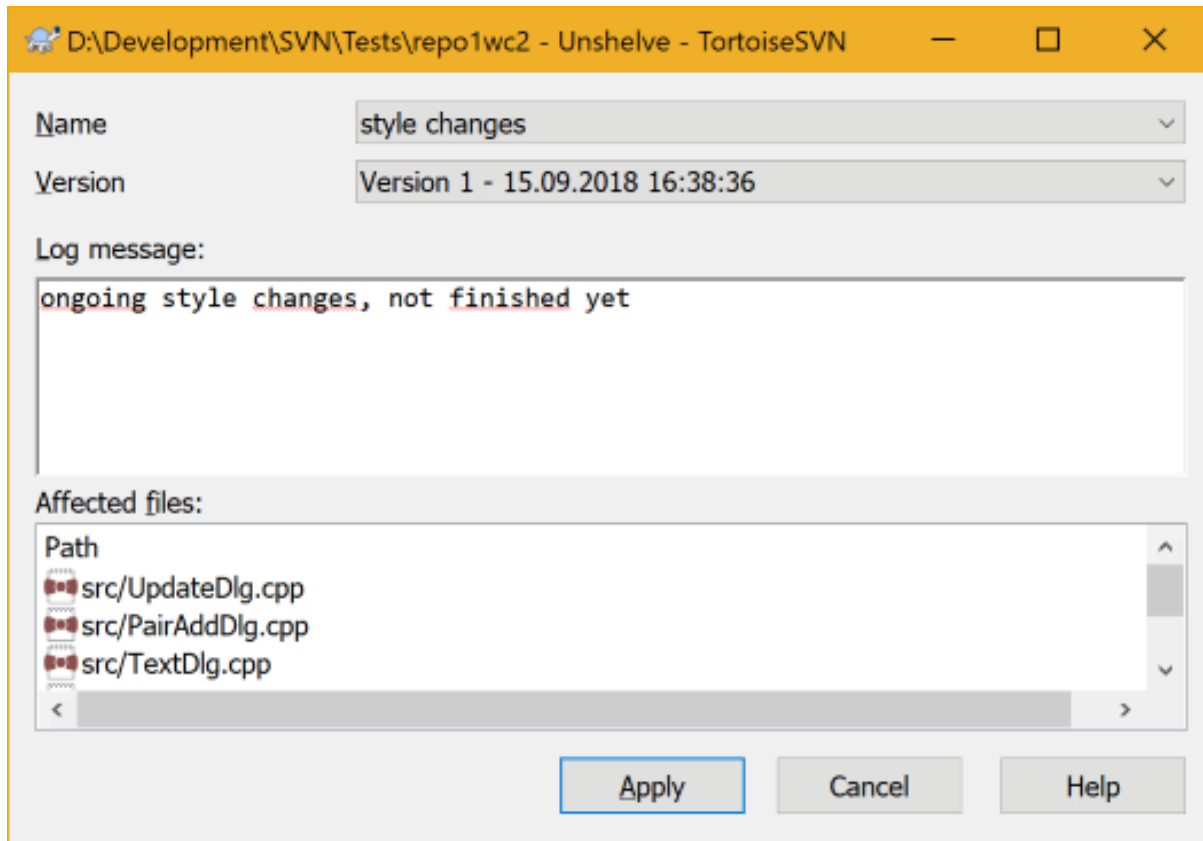


#### Afbeelding 4.15. Shelve dialog

If you select an existing shelf, then a new version is created for that shelf. If you provide a new name, a new shelf is created for the selected files.

If you click the **Shelve** button, the shelf is created and your working copy files are reset to a clean state. If you click the **Checkpoint** button, the shelf is created but your local modifications are kept.

To unshelve your changes, use **Context Menu** → **Unshelve** to get the unshelve dialog. This dialog shows you a list of all shelved items. Select the shelved item you want and the version to apply back to your working copy and click **Apply**.



Afbeelding 4.16. Unshelve dialog



### Tip

Shelves are purely a local client feature. Creating and removing Shelves will not affect the repository, nor anyone else's working copy.



### Experimental

The shelving feature is still marked as `experimental`.

That means that while shelving works as advertised, it is still in a stage where it's heavily improved and worked on. That also means that there's no guarantee that the shelves you create are upwards compatible and future versions might not be able to use them. And of course the UI might change as well in future versions to accommodate new features and behaviors.

## 4.10. Logboekboodschappen Venster

Bij elke wijziging die je doorvoert en vastlegt, moet je een logboekbericht maken. Dat helpt je om later te achterhalen waarom je welke wijzigingen hebt gedaan. Daarbij heb je dan ook meteen een gedetailleerd rapport van je ontwikkelproces.

Het Logboek boodschappen venster haalt alle logboekboodschappen op en toont deze. Het venster is verdeeld in 3 gedeeltes.

- Het bovenste gedeelte geeft een lijst van revisies weer, waarin wijzigingen aan bestanden en mappen zijn vastgelegd. Deze samenvatting geeft ook de datum, de tijd, degene die de revisie heeft vastgelegd en het begin van het logboekbericht weer.

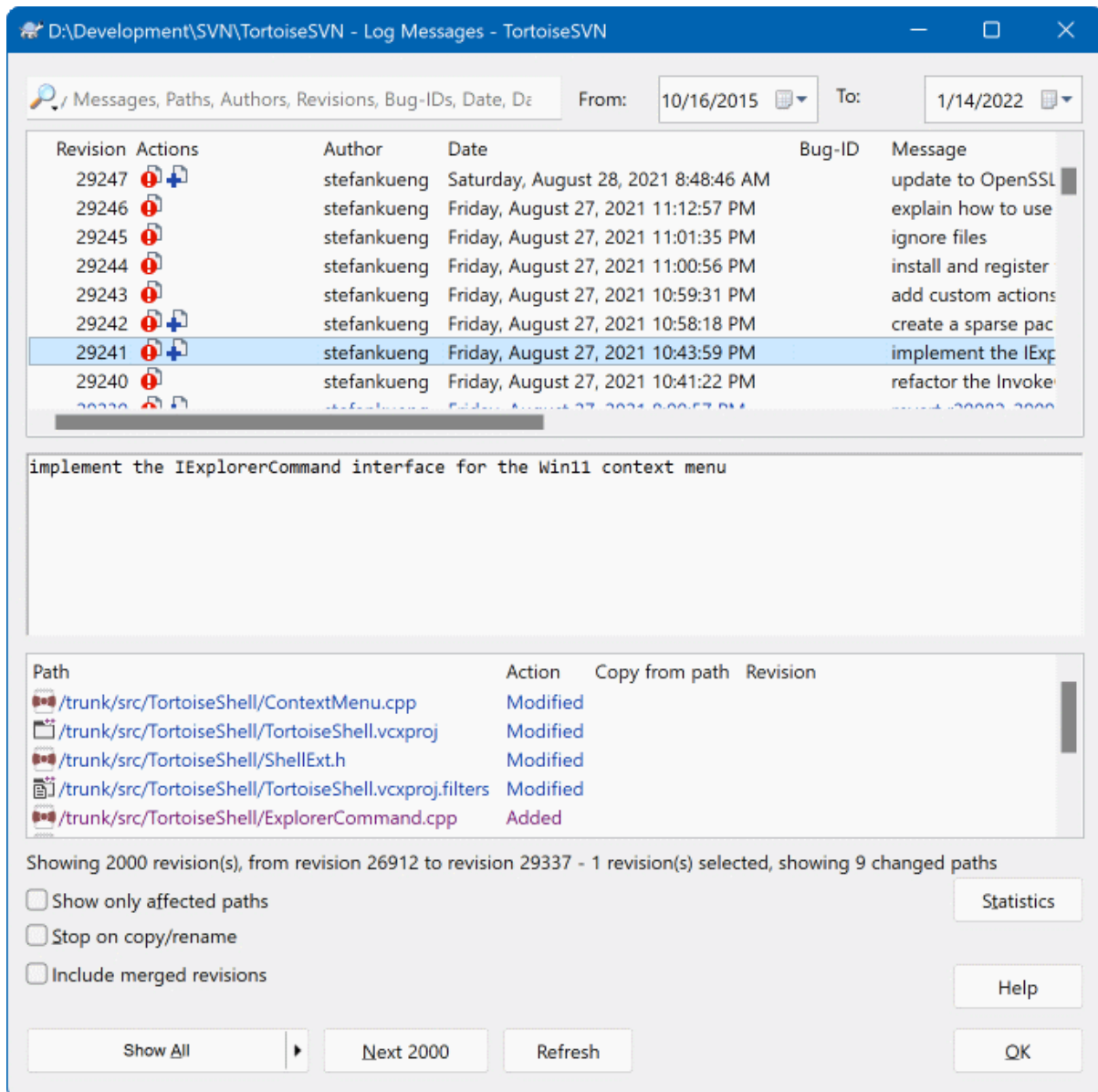


Blauwe regels geven aan dat er iets naar deze revisie is gekopieerd, verplaatst of hernoemt is (mogelijk vanuit een vertakking/branch).

- Het middelste gedeelte geeft het complete logboekbericht weer van de geselecteerde revisie.
- Het onderste gedeelte geeft de bestanden en mappen weer, die in deze revisie gewijzigd zijn.

Maar het doet nog veel meer dan dat alleen - er zijn ook functies beschikbaar in het context menu, waarmee je nog veel meer informatie over de geschiedenis van het project kunt opvragen.

#### 4.10.1. Het Logboek Boodschappen Venster Opstarten



**Afbeelding 4.17. Het Logboek Boodschappen Venster**

Er zijn verschillende plaatsen van waaruit je het Logboek boodschappen venster kunt openen:

- Vanuit het TortoiseSVN context submenu
- Vanuit de eigenschappen pagina



- Vanuit het voortgangsvenster nadat een verversing is afgerond. Dan zal het Logboek venster alleen de revisies weergeven van wijzigingen die doorgevoerd zijn, nadat je voor het laatst ververs hebt.
- From the repository browser

Als het archief niet beschikbaar is, zal er een venster krijgen met de vraag *Wilt u offline werken?*. Dit wordt verder beschreven in [Paragraaf 4.10.10, “Offline Modus”](#).

## 4.10.2. Revisie Logboek Acties

Het bovenste gedeelte van het venster heeft een *Acties* kolom, waarin met pictogrammen wordt aangegeven wat er in die revisie gedaan is. Er zijn vier verschillende pictogrammen, die elke in een aparte kolom worden weergegeven.



Als in een revisie bestanden of mappen gewijzigd zijn, dan wordt het *gewijzigd* pictogram weergegeven in de eerste kolom.



Als in een revisie een bestand of map is toegevoegd, dan wordt het *toegevoegd* pictogram weergegeven in de tweede kolom.



Als in een revisie bestanden of mappen verwijderd zijn, dan wordt het *verwijderd* pictogram weergegeven in de derde kolom.



Als in een revisie bestanden of mappen vervangen zijn, dan wordt het *vervangen* pictogram weergegeven in de vierde kolom.



If a revision moved or renamed a file or directory, the *moved* icon is shown in the fourth column.



If a revision replaced a file or directory by moving/renaming it, the *move replaced* icon is shown in the fourth column.

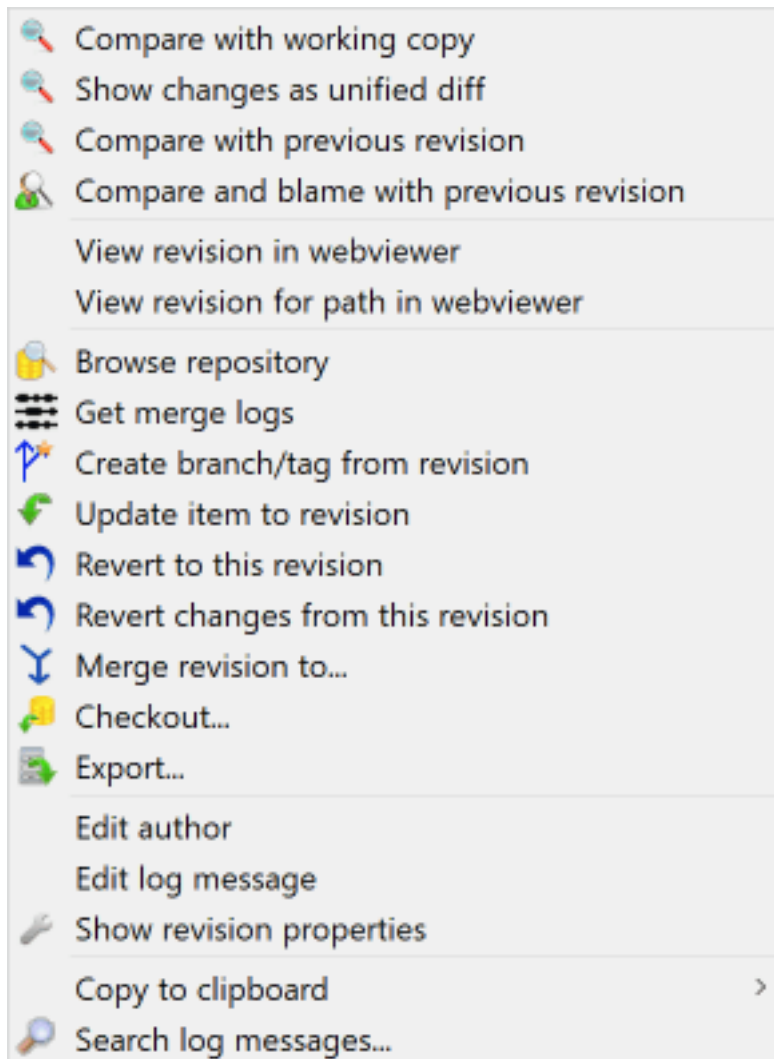


If a revision merged a file or directory, the *merged* icon is shown in the fourth column.



If a revision reverse merged a file or directory, the *reverse merged* icon is shown in the fourth column.

### 4.10.3. Extra Informatie Opvragen



#### Afbeelding 4.18. Het Context Menu Van Het Bovenste Gedeelte In Het Logboek Boodschappen Venster

The top pane of the Log dialog has a context menu that allows you to access much more information. Some of these menu entries appear only when the log is shown for a file, and some only when the log is shown for a folder.

##### Vergelijk met werkkopie

Vergelijk de geselecteerde revisie met je werkkopie. Het standaard Diff-programma is TortoiseMerge, welke meegeleverd wordt met TortoiseSVN. Als het logboek venster wordt weergegeven voor een map, dan zal in de lijst de gewijzigde bestanden weergegeven worden en heb je de mogelijkheid om de wijzigingen van individuele bestanden te bekijken.

##### Vergelijk en verklaar met werkkopie (BASE)

Blame the selected revision, and the file in your working BASE and compare the blame reports using a visual diff tool. Read [Paragraaf 4.24.2, “Verklaar Verschillen”](#) for more detail. (files only)

##### Toon wijzigingen als unified diff

Bekijk de wijzigingen van de geselecteerde revisie als Unified-Diff bestand (GNU patch formaat). Hiermee worden alleen de verschillen getoond met enkele regels voor de context. Het is lastiger om dit te lezen dan een visuele vergelijking, maar geeft wel alle wijzigingen compact weer.

If you hold down the **Shift** key when clicking on the menu item, a dialog shows up first where you can set options for the unified diff. These options include the ability to ignore changes in line endings and whitespaces.

#### Vergelijk met vorige revisie

Vergelijk de geselecteerde revisie met de vorige revisie. Deze functie werkt ongeveer gelijk als het vergelijken met je werkkopie. Als je deze functie gebruikt op mappen, dan zal er eerst een venster weergegeven worden met gewijzigde bestanden. Je kunt dan selecteren welke bestanden je wilt vergelijken.

#### Vergelijk en verklaar met vorige revisie

Show the changed files dialog allowing you to select files. Blame the selected revision, and the previous revision, and compare the results using a visual diff tool. (folders only)

#### Bewaar revisie in...

De geselecteerde revisie opslaan in een bestand, zodat je de oude versie van het bestand hebt. (Alleen voor bestanden)

#### Openen / Openen met...

Open het geselecteerde bestand, ofwel het met standaardprogramma voor dat bestandstype, ofwel met een programma naar keuze. (enkel bestanden)

#### Verklaren...

Blame the file up to the selected revision. (files only)

#### Archief bekijken

Open de archief-verkenner om de geselecteerde bestanden en mappen te onderzoeken zoals deze waren op het moment van de geselecteerde revisie.

#### Maak tak/label van revisie

Maak een tak of label van de geselecteerde revisie. Deze functie is handig als je bijvoorbeeld vergeten bent een label aan te maken terwijl je wijzigingen al vastgelegd hebt, die eigenlijk niet in die versie zouden moeten zitten.

#### Ververs onderdeel naar revisie

Ververs je werkkopie met de gegevens van de geselecteerde revisie. Deze functie is handig als je de status van een project wilt bekijken uit het verleden. Ook is deze handig als er verschillende nieuwe revisies zijn aangemaakt en je je eigen werkkopie stapsgewijs wilt bijwerken in plaats van in één keer de hele sprong te nemen. Het is verstandig om een hele map te verversen in plaats van losse bestanden. Hiermee voorkom je dat je werkkopie inconsistenties zal bevatten.

Als je een eerdere wijziging permanent ongedaan wilt maken, gebruik dan **Ongedaan maken tot deze revisie**.

#### Ongedaan maken tot deze revisie

Maak de wijzigingen ongedaan en ga terug naar de geselecteerde revisie. Mocht je wijzigingen gemaakt hebben en tot de conclusie komen dat je eigenlijk terug wilt naar de stand van zaken bij revisie N, dan is dit het commando dat je moet gebruiken. Met deze functie maak je de wijzigingen die je doorgevoerd hebt in je werkkopie ongedaan. Deze functie verandert dus *niets* in het archief, totdat je zelf wijzigingen vastlegt. Merk op dat deze functie *alle* wijzigingen die gedaan zijn na de geselecteerde revisie ongedaan maakt en dus het bestand of de map vervangt door de oudere versie.

Als je werkkopie voordat je deze actie uitvoert ongewijzigd is, dan zal na deze actie je werkkopie als gewijzigd aangegeven worden. Als je al lokale wijzigingen had aangebracht, dan zal met dit commando de *ongedaan-maken* wijzigingen worden samengevoegd in je werkkopie.

Op de achtergrond voert Subversion eigenlijk een samenvoeging van aangebrachte wijzigingen tot aan de geselecteerde revisie achterstevoren uit, waardoor de acties van eerdere vastleggingen ongedaan worden gemaakt.

Als je na het uitvoeren van deze actie bedenkt dat je eigenlijk de *ongedaanmaking ongedaan wilt maken*, zodat je je werkkopie terugkrijgt in de ongewijzigde situatie, dan moet je de functie **TortoiseSVN → Ongedaan maken** gebruiken vanuit de Windows Verkenner. Hiermee worden de lokale wijzigingen die door deze achterstevoren-samenvoeging waren gemaakt, teruggedraaid.

Als je alleen maar wilt zien hoe een bestand of map er uit zag bij een eerdere revisie, gebruik dan **Verversen naar revisie** of **Revisie opslaan als...**

#### Revisie wijzigingen ongedaan maken

Wijzigingen van de geselecteerde revisie ongedaan maken. De wijzigingen worden ongedaan gemaakt in je werkkopie, oftewel in het archief zal er *niets* veranderd worden. Merk op dat hiermee alleen de wijzigingen van die revisie ongedaan worden gemaakt in je lokale bestand; het bestand in je werkkopie wordt niet compleet vervangen door die van de eerdere revisie. Deze functie is erg handig als je een eerdere wijziging ongedaan wilt maken, terwijl er in de tussentijd andere ongerelateerde wijzigingen zijn doorgevoerd.

Als je werkkopie voordat je deze actie uitvoert ongewijzigd is, dan zal na deze actie je werkkopie als gewijzigd aangegeven worden. Als je al lokale wijzigingen had aangebracht, dan zal met dit commando de *ongedaan-maken* wijzigingen worden samengevoegd in je werkkopie.

Op de achtergrond voert Subversion eigenlijk een samenvoeging van die specifieke revisie achterstevoren uit, waardoor het resultaat van een vorige vastlegging ongedaan wordt gemaakt.

Je kunt de *ongedaanmaking* weer *ongedaan maken* zoals eerder beschreven is bij **Ongedaan maken tot deze revisie**.

#### Voeg revisie samen met...

Voeg de geselecteerde revisie(s) samen in een andere werkkopie. In venster kun de map selecteren van de werkkopie waarin je de samenvoeging wilt opslaan. Daarna komt er geen bevestigingsvenster meer en is er ook geen optie om de samenvoeging te testen. Het is daarom verstanding om de samenvoeging uit te voeren in een ongewijzigde werkkopie, zodat je de wijzigingen ongedaan kunt maken als het uiteindelijk niet blijkt te worden wat je had willen hebben. Deze functie komt van pas als je revisies van één tak in een andere wilt samenvoegen.

#### Ophalen...

Haal een verse kopie op van de geselecteerde map uit de betreffende revisie. Er wordt dan een venster weergegeven waarin je de URL en de revisie moet bevestigen en de locatie waar de map naartoe opgehaald moet worden.

#### Exporteren...

Exporteer het geselecteerde bestand of de map van de geselecteerde revisie. Er wordt dan een venster weergegeven waarin je de URL en de revisie moet bevestigen en de locatie op moet geven waar naartoe de map geëxporteerd moet worden.

#### Auteur wijzigen / Logboek boodschap wijzigen

Bewerk de logboekboodschap of auteur van een vorige vastlegging. Lees [Paragraaf 4.10.7, "Logboekboodschap en Auteur Wijzigen"](#) om hier meer over te weten te komen.

#### Toon revisie-eigenschappen

Bekijk en bewerk een eigenschap van een revisie, niet beperkt tot de logboekboodschap of auteur. Lees [Paragraaf 4.10.7, "Logboekboodschap en Auteur Wijzigen"](#).

#### Naar klembord kopiëren

Kopieer details uit het logboek van de geselecteerde revisie naar het klembord. Hierbij worden het revisienummer, auteur, datum, logboekboodschap en de lijst met gewijzigde objecten voor elke revisie gekopieerd.

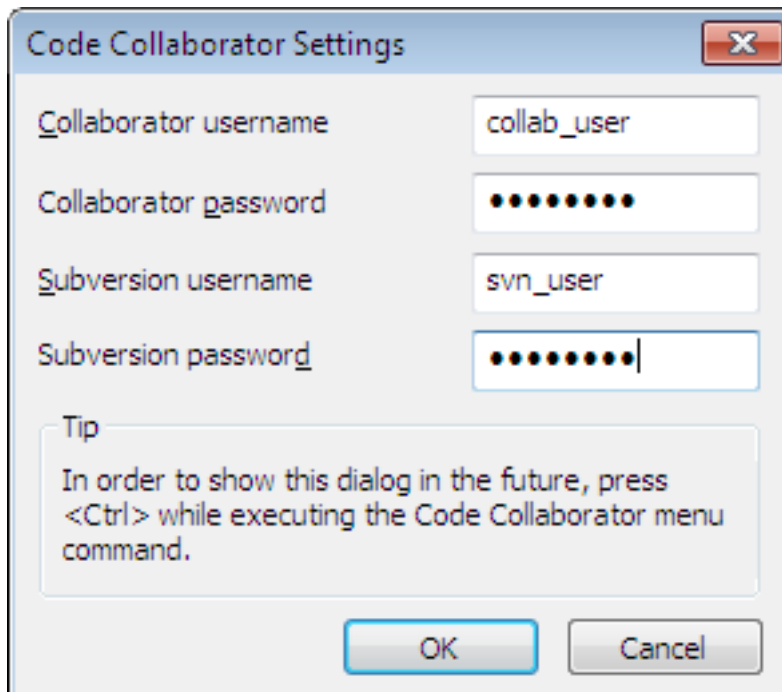
#### Doorzoek logboekberichten...

Doorzoek de logboekboodschappen op de ingevoerde tekst. Hiermee wordt er gezocht in de logboekberichten en in de samenvattingen van Subversion acties (die weergegeven worden in het onderste deel van het venster). De zoekopdrachten zijn niet hoofdlettergevoelig.

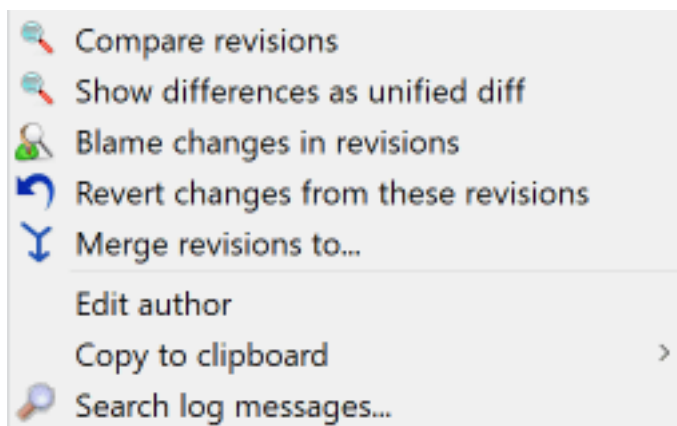
#### Create code collaborator review...

This menu is shown only if the SmartBear code collaborator tool is installed. When invoked for the first time, a dialog is shown prompting the user to enter user credentials for both code collaborator and SVN. Once the settings are stored, the settings dialog is no longer shown when the menu is invoked, unless the user holds

**Ctrl** while executing the menu item. The configuration and the chosen revision(s) are used to invoke the code collaborator graphical user interface client, which creates a new review with the selected revisions.



**Afbeelding 4.19. The Code Collaborator Settings Dialog**



**Afbeelding 4.20. Context menu van het bovenste deelvenster voor 2 geselecteerde revisies**

Als je twee revisies selecteert (als je klikt en **Ctrl** ingedrukt houdt), dan zullen er in het context menu minder opties worden weergegeven:

#### Vergelijk revisies

Vergelijk de twee geselecteerde revisies met een visueel vergelijkingsprogramma. Het standaard Diff-programma is TortoiseMerge, welke meegeleverd wordt met TortoiseSVN.

Een nieuw venster wordt er geopend, als je deze optie selecteert voor een map. Daarin worden de gewijzigde bestanden weergegeven en worden er meer opties voor het vergelijken gegeven. Lees meer over het Vergelijk revisies venster in [Paragraaf 4.11.3, “Vergelijken van mappen”](#).

#### Verklaar revisies

Verklaar de twee revisies en vergelijk de verklaarrapportages in een visueel vergelijk programma. Lees voor meer informatie [Paragraaf 4.24.2, “Verklaar Verschillen”](#).

Toon verschillen als unified diff

Bekijk de verschillen van twee verschillende revisies opgeslagen in een unified-diff bestand.

Naar klembord kopiëren

Kopieer de logboekboodschappen naar het klembord.

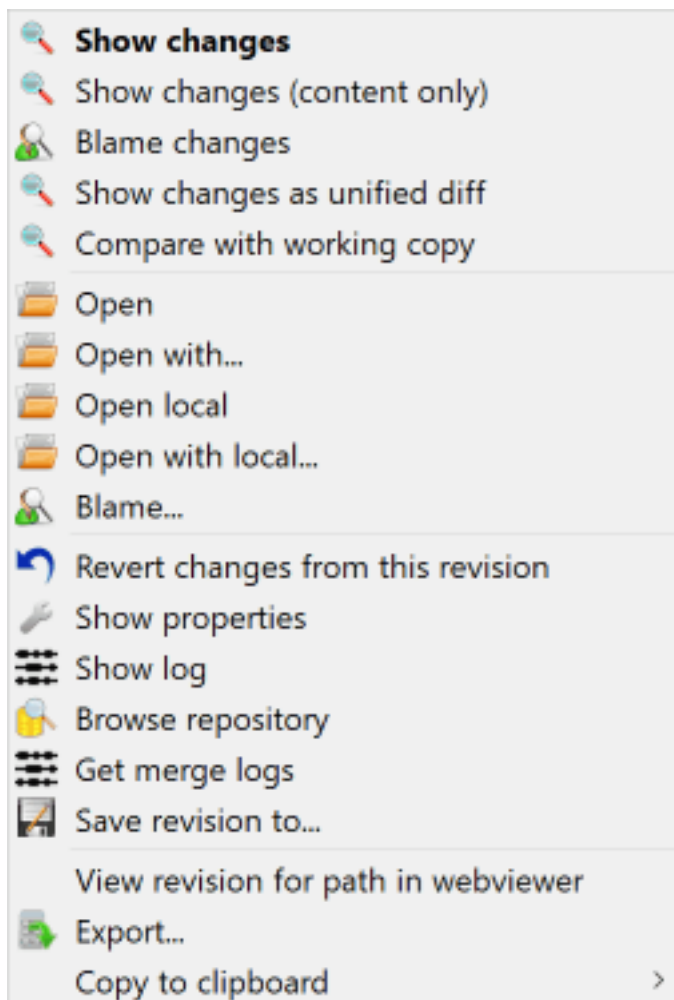
Doorzoek logboekberichten...

Doorzoek de logboekboodschappen.

Als je twee of meer revisies selecteert (als je klikt en **Ctrl** of **Shift** ingedrukt houdt), dan zal in het context menu een optie weergegeven worden waarmee alle wijzigingen van die revisies teruggedraaid kunnen worden. Dit is de makkelijkste methode om een verzameling van wijzigingen ineens ongedaan te maken.

Je kunt er ook voor kiezen om de geselecteerde revisies samen te voegen naar een andere werkkopie, zoals eerder beschreven is.

Als alle revisies van dezelfde auteur zijn, dan kun je in één keer de naam van de auteur wijzigingen voor die revisies.



**Afbeelding 4.21. Het context menu van het onderste deelvenster van het Logboekboodschappen venster**

The bottom pane of the Log dialog also has a context menu that allows you to

Toon wijzigingen

Toon de wijzigingen doorgevoerd in de geselecteerde revisie voor het geselecteerde bestand.

Verklaar wijzigingen

Verklaar de geselecteerde en vorige revisie van het geselecteerde bestand en vergelijk de verklarenrapportages met een visueel diff programma. Lees ook [Paragraaf 4.24.2, “Verklaar Verschillen”](#).

Toon wijzigingen als unified diff

Toon bestandswijzigingen in unified diff indeling. Dit context menu is alleen beschikbaar voor als *gewijzigd* gemarkeerde bestanden.

Openen / Openen met...

Open het geselecteerde bestand met het standaard programma voor dit bestandstype of met een zelf gekozen programma.

Verklaren...

Opent het verklaren venster, waarmee de geselecteerde revisie verklaard kan worden.

Revisie wijzigingen ongedaan maken

Make de wijzigingen van het geselecteerde bestand in die revisie ongedaan.

Weergeef eigenschappen

Bekijk de Subversion eigenschappen van het geselecteerde object.

Toon logboek

Toon het revisie logboek voor een enkel geselecteerd bestand.

Haal samenvoegboodschappen

Toon het revisie logboek voor een enkel geselecteerd bestand met de samengevoegde wijzigingen. Lees meer in [Paragraaf 4.10.6, “Samenvoeg Traceer Functies”](#).

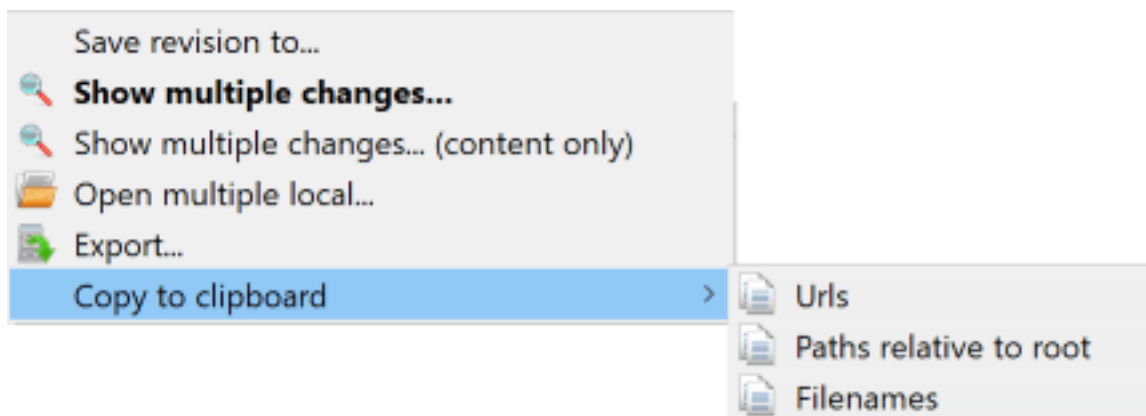
Bewaar revisie in...

Sla de geselecteerde revisie op in een bestand, zodat je een oudere versie van dat bestand hebt.

Exporteren...

Export the selected items in this revision to a folder, preserving the file hierarchy.

When multiple files are selected in the bottom pane of the Log dialog, the context menu changes to the following:



**Afbeelding 4.22. The Log Dialog Bottom Pane with Context Menu When Multiple Files Selected.**

Bewaar revisie in...

Sla de geselecteerde revisie op in een bestand, zodat je een oudere versie van dat bestand hebt.

Show multiple changes...

Show changes made in the selected revision for the selected files. Note that the show changes functionality is invoked multiple times, which may bring up multiple copies of your selected diff tool, or just add a new

comparison tab in your diff tool. If you have selected more than 15 files, you will be prompted to confirm the action.

#### Open multiple local...

This will open local working copy files that correspond to your selected files using the application that is registered for the extension. [The behavior is the one you would get double-clicking the working-copy file(s) in Windows explorer]. Depending on how your file extension is associated to an application and the capabilities of the application, this may be a slow operation. In the worst case, new instances of the application may be launched by Windows for each file that was selected.

If you hold **Ctrl** while invoking this command, the working copy files are always loaded into Visual Studio. This only works when the following conditions are met: Visual Studio must be running in the same user context while having the same process integrity level [running as admin or not] as TortoiseProc.exe. It may be desirable to have the solution containing the changed files loaded, although this is not strictly necessary. Only files that exist on disk with extensions [.cpp, .h, .cs, .rc, .resx, .xaml, .js, .html, .htm, .asp, .aspx, .php, .css and .xml] will be loaded. A maximum of 100 files can be loaded into Visual Studio at one time, and the files are always loaded as new tabs into the currently open instance of Visual Studio. The benefit of reviewing code changes in Visual Studio lies in the fact that you can then use the built-in code navigation, reference finding, static code analysis and other tools built into Visual Studio.

#### Exporteren...

Export the selected files/folder at the selected revision. This brings up a dialog for you to confirm the URL and revision, and select a location for the export.



### Tip

Misschien is het je opgevallen dat we soms refereren naar wijzigingen en soms naar verschillen. Wat is het verschil?

Subversion uses revision numbers to mean 2 different things. A revision generally represents the state of the repository at a point in time, but it can also be used to represent the changeset which created that revision, e.g. “Done in r1234” means that the changes committed in r1234 implement feature X. To make it clearer which sense is being used, we use two different terms.

Als je twee revisies N en M selecteert, dan geeft het context menu de mogelijkheid om de *verschillen* tussen de twee revisies te laten zien. In Subversion termen wordt dit aangegeven door `diff -r M:N`.

Als je alleen revisie N selecteert, dan geeft het context menu de mogelijkheid om de *wijzigingen* van die revisie te laten zien. In Subversion termen wordt dit aangegeven met `diff -r N-1:N` of `diff -c N`.

Het onderste deelvenster laat de bestanden zien die gewijzigd zijn in alle geselecteerde revisies, oftewel het context menu zal altijd de mogelijkheid geven om *wijzigingen* te laten zien.

## 4.10.4. Ophalen van uitgebreide logboek boodschappen

Het Logboek boodschappen venster geeft niet altijd alle in het verleden doorgevoerde wijzigingen weer met verschillende redenen:

- Bij een groot archief zijn er waarschijnlijk honderden of misschien wel duizenden wijzigingen. Binnenhalen van al die wijzigingen neemt meestal veel tijd in beslag, terwijl je waarschijnlijk alleen geïnteresseerd bent in de meest recente wijzigingen. Standaard worden er maximaal 100 logboek boodschappen opgehaald, maar je kunt deze waarde wijzigen bij TortoiseSVN → Instellingen (Paragraaf 4.31.1.2, “TortoiseSVN Instellingen Venster 1”).
- Als de Stoppen bij kopie/hernoemd bestand optie geselecteerd is, dan zal Toon Logboek stoppen op het punt waarop het geselecteerde bestand of de map vanaf een ander punt uit het archief gekopieerd is. Dit is handig



bij het bekijken van takken (of labels), omdat er dan gestopt wordt bij de bron van de tak en er meteen een snel overzicht is van de wijzigingen in de tak.

Normaal gesproken zal je deze optie uit laten staan. TortoiseSVN onthoudt deze instelling, zodat jouw voorkeur bewaard blijft.

Als het Toon Logboek venster wordt aangeroepen vanuit het Samenvoegen venster, dan zal deze optie standaard geselecteerd zijn. Dit komt doordat bij het samenvoegen van takken er meestal gekeken wordt naar de wijzigingen van die specifieke tak. Verder terug kijken dan de bron van de tak heeft dan vaak geen nut.

Merk op dat Subversion hernoemen uitvoert als een kopieer/verwijder combinatie. Een hernoemde map of bestand zal er ook voor zorgen dat het logboek venster stopt als de Stoppen bij kopie/hernoemd bestand optie geselecteerd is.

Als je meer logboek boodschappen wilt zien, klik dan op de **Volgende 100** knop om de volgende 100 berichten op te halen. Je kunt dit zo vaak herhalen als je wilt.

Naast deze knop is er een multi-functionele knop. Hierbij wordt de laatst gebruikte optie onthouden, zodat je die weer kunt gebruiken. Klik op het pijltje om de andere opties te zien.

Gebruik **Toon Bereik ...** als je een bepaalde reeks van revisies wilt bekijken. Er zal dan een venster getoond worden, waarin je de start en eind revisie moet ingeven.

Gebruik de **Toon Alles** knop als je *alle* logboek boodschappen vanaf de HEAD revisie teruglopend tot aan revisie 1 wilt zien.

To refresh the latest revision in case there were other commits while the log dialog was open, hit the **F5** key.

Om de log-cache te verversen, druk op de **Ctrl-F5** toetsen.

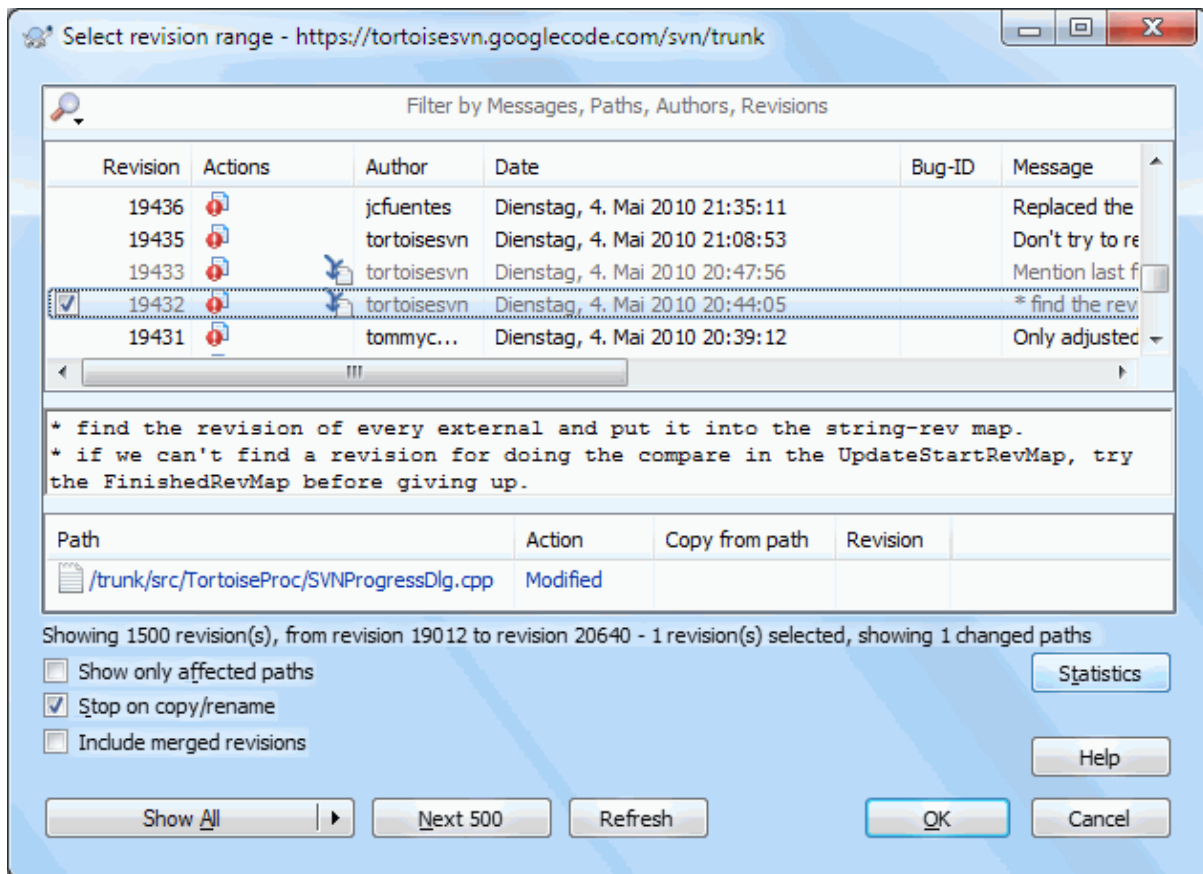
#### 4.10.5. Huidige Werkkopie Revisie

Het logboek boodschappen venster toont de berichten van de HEAD revisie, niet van de huidige werkkopie revisie. Hierdoor komt het voor dat er berichten getoond worden voor bestanden en mappen die nog niet bijgewerkt zijn in je werkkopie. Om hier een duidelijker onderscheid in te maken, wordt het bericht van de vastlegging van jouw werkkopie revisie vetgedrukt weergegeven.

Als het logboek voor een map wordt weergegeven, dan is de gemarkeerde revisie de hoogste revisie die gevonden is in die map. Hiervoor wordt de hele werkkopie doorzocht. Dit gebeurt parallel in een aparte thread, zodat het weergeven van het revisie logboek geen vertraging oploopt. Als gevolg hiervan verschijnen de markering van de mappen niet direct.

#### 4.10.6. Samenvoeg Traceer Functies

Subversion 1.5 en later slaat met behulp van eigenschappen de geschiedenis van Samenvoeg acties op. Hierdoor kun je een gedetailleerd overzicht krijgen van samengevoegde wijzigingen. Bijvoorbeeld, je ontwikkelt een nieuwe functie in een tak en voegt die later samen in de basislijn. In het logboek van de basislijn is de ontwikkeling van deze functie te zien als een enkele vastlegging, terwijl er misschien 1000 vastleggingen in de tak aan vooraf zijn gegaan.



**Afbeelding 4.23. Het logboekboodschappen venster toont de revisies van samenvoegingen**

Als je de details wilt zien welke revisies samengevoegd zijn bij die vastlegging, selecteer dan het **Samengevoegde revisies meenemen** selectievak. Hiermee zullen de logboekboodschappen opnieuw opgehaald worden en zullen de logboekboodschappen van de samengevoegde revisies tussen de andere logboekboodschappen ingevoegd worden. De samengevoegde revisies worden in grijs weergegeven, omdat deze wijzigingen aangegeven van een andere boomstructuur.

Samenvoegen is natuurlijk nooit eenvoudig. Gedurende de ontwikkelingen aan een tak zullen er soms samenvoegingen vanuit de basislijn gedaan moeten worden om ervoor te zorgen dat de tak bij blijft met de hoofdbroncode. Dat houdt in dat de samenvoeggeschiedenis van een tak ook een extra laag van samenvoeggeschiedenis zal hebben. Deze verschillende lagen worden met verschillende inspringingen weergegeven.

#### 4.10.7. Logboekboodschap en Auteur Wijzigen

Revisie eigenschappen zijn compleet anders dan de eigenschappen van Subversion objecten. Revprops zijn beschrijvende objecten die gerelateerd zijn aan een specifiek revisienummer in het archief, zoals logboekboodschap, vastleggingsdatum en naam van de vastlegger (auteur).

Soms wil je een logboekboodschap wijzigen, omdat er bijvoorbeeld een tikfout in staat, de beschrijving wilt verbeteren of een ander soort wijziging wilt doorvoeren. Misschien wil je wel de auteur van de vastlegging wijzigen of omdat je authenticatie bent vergeten in te stellen or...

Subversion lets you change revision properties any time you want. But since such changes can't be undone (those changes are not versioned) this feature is disabled by default. To make this work, you must set up a pre-revprop-change hook. Please refer to the chapter on [Hook Scripts](http://svnbook.red-bean.com/en/1.8/svn.reposadmin.create.html#svn.reposadmin.create.hooks) [http://svnbook.red-bean.com/en/1.8/svn.reposadmin.create.html#svn.reposadmin.create.hooks] in the Subversion Book for details about how to do that. Read [Paragraaf 3.3, "Server gesitueerde hook scripts"](#) to find some further notes on implementing hooks on a Windows machine.

Als je eenmaal je server het met de benodigde hooks geconfigureerd hebt, dan kun je met het context menu uit het bovenste deelvenster van het Log venster de auteur en logboekboodschap (of elke andere revprop) van een revisie wijzigen. De logboekboodschap kun je ook wijzigen via het context menu van het middelste deelvenster.



## Waarschuwing

Omdat Subversion revisie eigenschappen niet van versiebeheer zijn voorzien, worden de oude waarden van een eigenschap (bijvoorbeeld de `svn:log` eigenschap) bij een wijzigingen voor *altijd* overschreven.



## Belangrijk

Since TortoiseSVN keeps a cache of all the log information, edits made for author and log messages will only show up on your local installation. Other users using TortoiseSVN will still see the cached (old) authors and log messages until they refresh the log cache. Refer to [Paragraaf 4.10.11, “De weergave verversen”](#)

### 4.10.8. Filter op Logmeldingen

Als je de getoonde logboek berichten wilt beperken tot degene waarin jij geïnteresseerd bent, dan kun je hiervoor de instellingen boven aan het Logboekboodschappen venster gebruiken. Met de start- en einddatum instellingen kun je het datumbereik instellen. Met het zoekveld kun je de berichten laten zien die alleen een bepaalde term bevatten.

Klik op het zoeken pictogram om te selecteren welke informatie je wilt zoeken en om de *regex* mode te kunnen kiezen. Normaal gesproken heb je alleen een deel-regel zoekopdracht nodig, maar als je een flexibelere zoekmethode wilt gebruiken, dan kun je ook vaak gebruikte expressies gebruiken. Als je de muis boven een vak houdt, dan worden er in een tooltip tips gegeven voor het gebruik van regex of deel-regel functies. Het filter vergelijkt de zoekopdracht met logboek berichten en toont dan de berichten die *overeenkomen* met de zoekopdracht.

Simple sub-string search works in a manner similar to a search engine. Strings to search for are separated by spaces, and all strings must match. You can use a leading `-` to specify that a particular sub-string is not found (invert matching for that term), and you can use `!` at the start of the expression to invert matching for the entire expression. You can use a leading `+` to specify that a sub-string should be included, even if previously excluded with a `-`. Note that the order of inclusion/exclusion is significant here. You can use quote marks to surround a string which must contain spaces, and if you want to search for a literal quotation mark you can use two quotation marks together as a self-escaping sequence. Note that the backslash character is *not* used as an escape character and has no special significance in simple sub-string searches. Examples will make this easier:

```
Alice Bob -Eve
```

searches for strings containing both Alice and Bob but not Eve

```
Alice -Bob +Eve
```

searches for strings containing both Alice but not Bob, or strings which contain Eve.

```
-Case +SpecialCase
```

searches for strings which do not contain Case, but still include strings which contain SpecialCase.

```
!Alice Bob
```

searches for strings which do not contain both Alice and Bob

```
!-Alice -Bob
```

do you remember De Morgan's theorem? NOT(NOT Alice AND NOT Bob) reduces to (Alice OR Bob).

```
"Alice and Bob"
```

searches for the literal expression "Alice and Bob"

```
" "
```

searches for a double-quote anywhere in the text

```
"Alice says "hi" to Bob"
```

searches for the literal expression "Alice says "hi" to Bob".

Het beschrijven van vaak gebruikte expressie zoekopdrachten gaat buiten het beoogde bereik van deze handleiding. Op het internet kun je veel meer informatie hierover vinden en een zelfstudie is te vinden op <http://www.regular-expressions.info/>.

Merk op dat de filters alleen toegepast worden op de berichten die al zijn opgehaald. Hiermee wordt niet het ophalen van berichten uit het archief gefilterd.

Je kunt ook filteren op pad-namen in het onderste deelvenster door gebruik te maken van het **Toon alleen veranderde paden** selectievak. Veranderde paden zijn de paden die gebruikt zijn om het logboek weer te geven. Als je het logboek voor een map ophaalt, dan houdt dat alles in die map en onderliggende mappen in. Bij een bestand gaat het dan alleen om het bestand. Normaliter wordt in de paden-lijst elk ander pad getoond die veranderd zijn bij een vastlegging, maar dan in het grijs. Als dit selectievak geselecteerd is, worden die paden verborgen.

In sommige gevallen vereist je werkmethode een bepaalde indeling van de logboekboodschappen. Dit kan tot gevolg hebben dat de beschrijvende tekst van de wijziging niet zichtbaar is in de samenvatting, die weergegeven wordt in het bovenste deelvenster. De eigenschap `tsvn:logsummary` kan gebruikt worden om een deel van de logboekboodschap te extraheren en in het bovenste deelvenster te laten tonen. Lees [Paragraaf 4.18.2, "TortoiseSVN Project Eigenschappen"](#) om te weten te komen hoe je deze eigenschap moet gebruiken.



### Geen Logopmaak van de archief-verkenner

Because the formatting 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.

#### 4.10.9. Statistische informatie

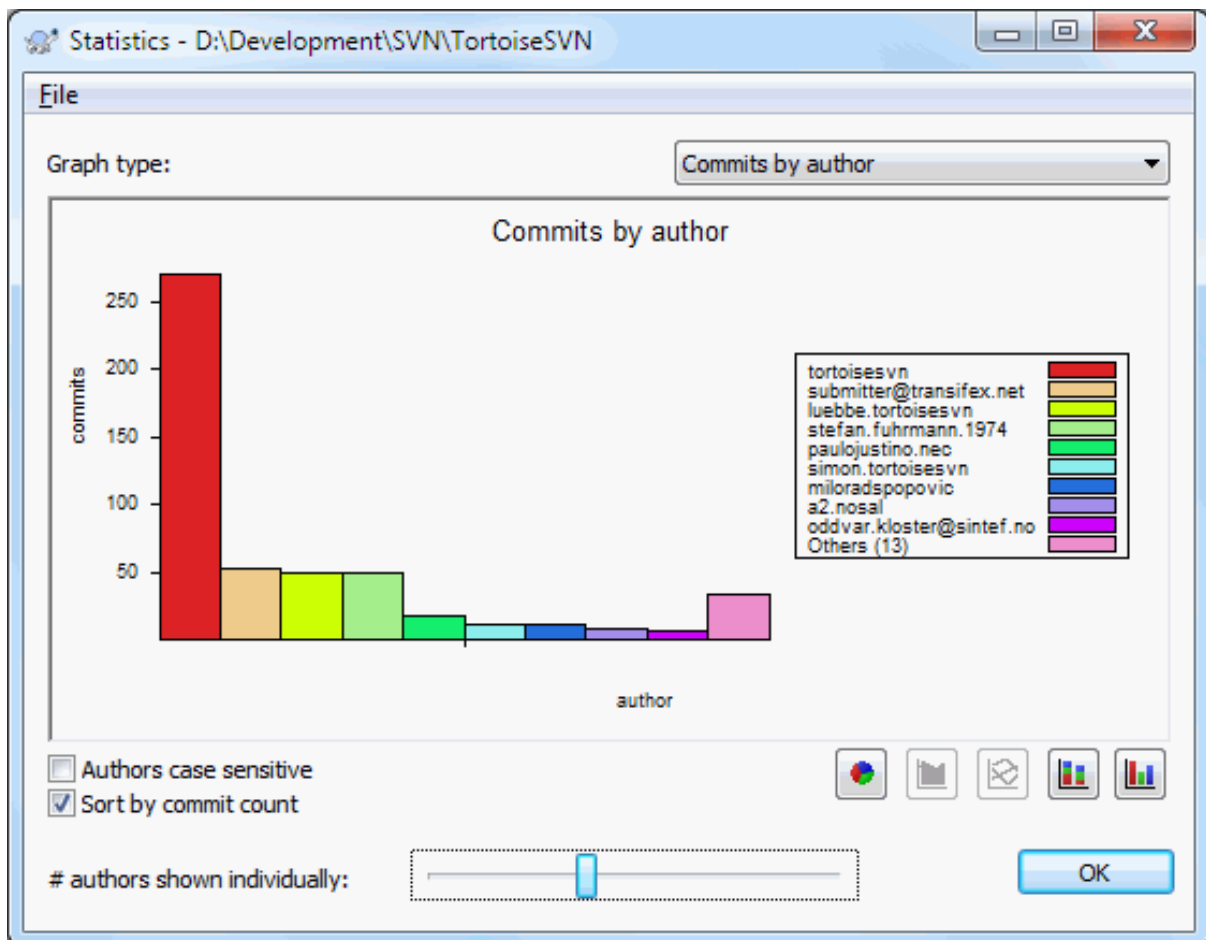
Met de **Statistics** knop kun je een scherm oproepen met interessante informatie over revisies die weergegeven worden in het logboekboodschappenvenster. In dit scherm wordt weergegeven hoeveel auteurs er aan het werk

geweest zijn, hoeveel vastleggingen zijn hebben gedaan, de voortgang per week en nog veel meer. Nu kun je in één oogopslag zien wie het hardst heeft gewerkt en wie er loopt te treuzelen. ;-)

#### 4.10.9.1. Statistiekenpagina

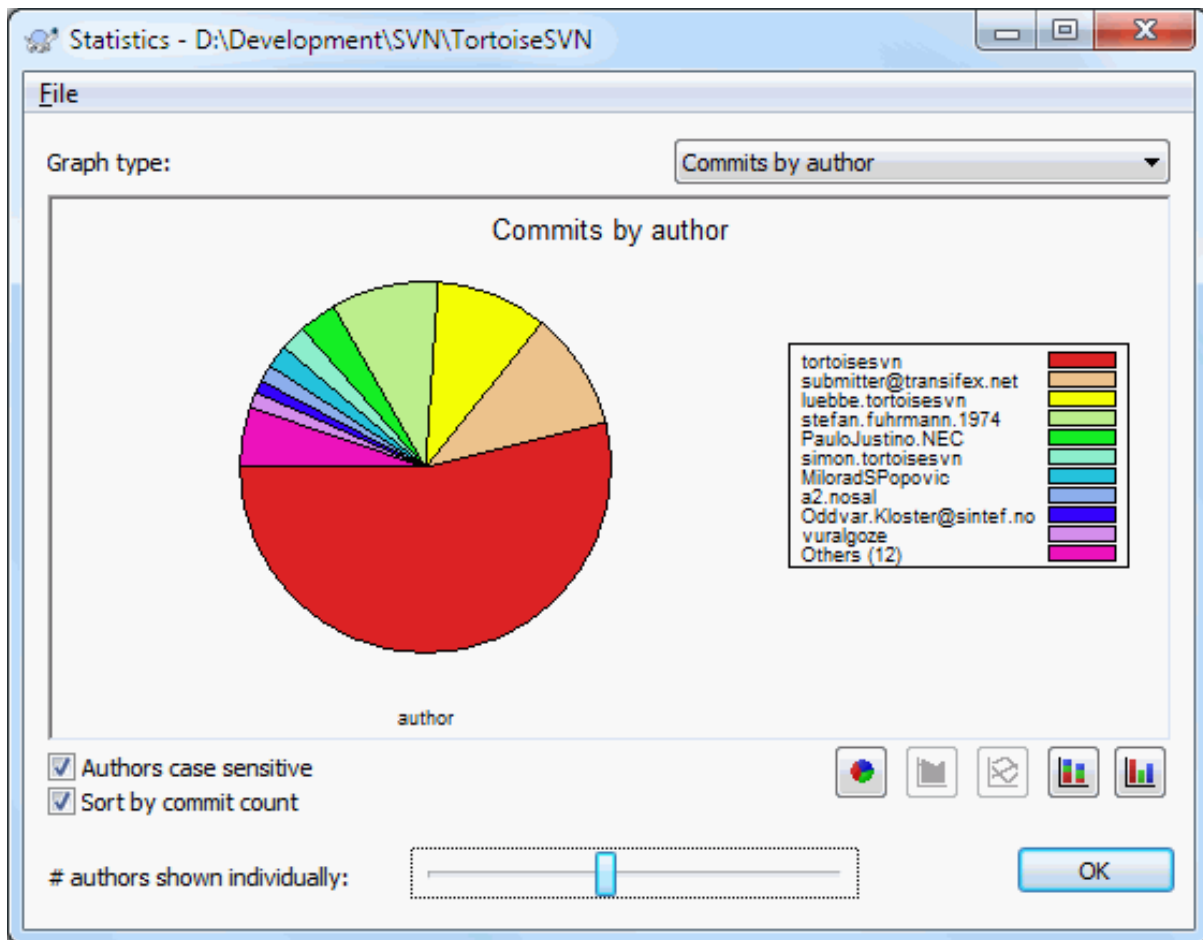
Op deze pagina vind je alle getallen die je maar kunt bedenken, maar in het bijzonder die van een bepaalde periode en aantal revisies die geselecteerd zijn en bijbehorende min/max/gemiddelde waarden.

#### 4.10.9.2. Vastleggingen per Auteur



**Afbeelding 4.24. Vastleggingen per Auteur Histogram**

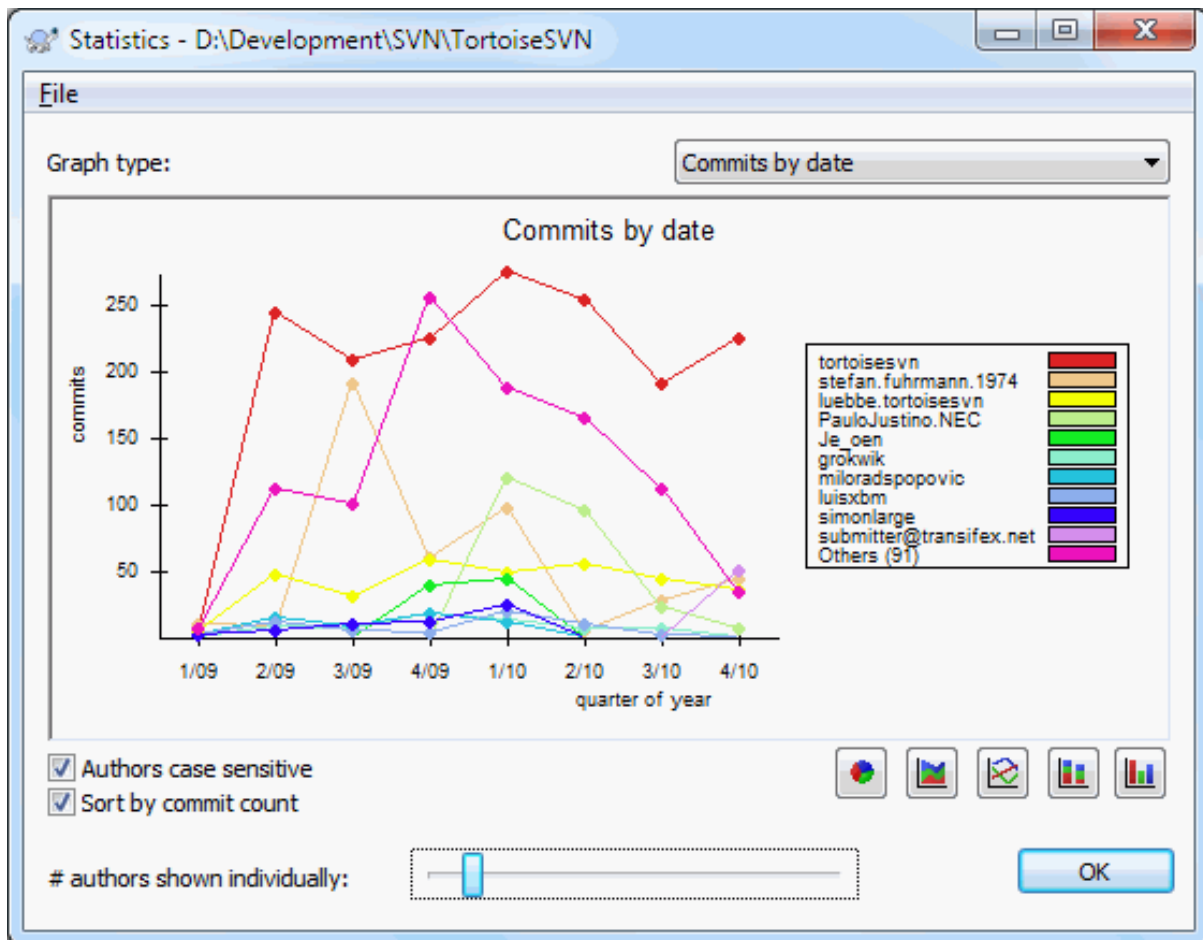
Dit figuur laat zien welke auteurs actief zijn geweest aan het project in de vorm van een eenvoudige histogram, gestapelde histogram, of taartdiagram.



**Afbeelding 4.25. Vastleggingen-per-Auteur taartdiagram**

Als er enkele zeer actieve auteurs zijn en vele minder actieve, dan wordt door het aantal kleine segmenten de grafiek snel onleesbaar. Met de schuifbalk onderaan het venster kun je de drempel bepalen (als percentage van het totaal aantal vastleggingen) waaronder activiteiten gegroepeerd moeten worden in een categorie met *Anderen*.

### 4.10.9.3. Vastleggingen per datum



**Afbeelding 4.26. Vastleggingen-per-datum Grafiek**

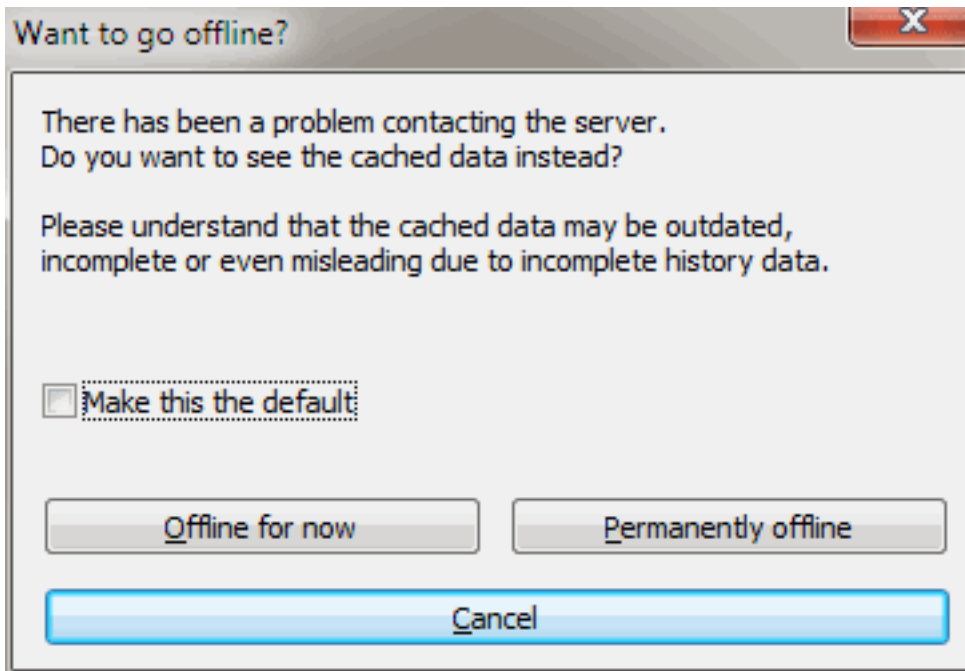
Deze grafiek is een grafische weergave van de project activiteiten in de zin van aantal vastleggingen *en* auteur. Hiermee kun je een idee krijgen wanneer er aan een project gewerkt wordt en wie er op welk moment aan werkt.

Er zullen veel lijnen in de grafiek weergegeven worden als er meerdere auteurs zijn. Er zijn hier twee weergavemodi beschikbaar: *normaal*, waarbij de activiteit van elke auteur ten opzichte van de nullijn wordt weergegeven, en *gestapeld*, waarbij de activiteit ten opzichte van de lijn eronder wordt weergegeven. Met de laatste modus voorkom je dat lijnen elkaar kruisen. Hiermee wordt de grafiek overzichtelijker, maar ook meteen lastiger om de individuele bijdrage te bekijken.

De analyse is standaard ingesteld als hoofdlettergevoelig, dus de gebruikers PeterEgan en PeteRegan worden als aparte auteurs behandeld. Echter zijn in veel gevallen de gebruikersnamen niet hoofdlettergevoelig en dus worden ze vaak inconsequent ingevoerd. In die gevallen wil je dat DavidMorgan en davidmorgan als dezelfde persoon behandeld worden. Gebruik het Exacte auteursnamen selectievak om dit in te stellen.

Merk op dat statistieken over dezelfde periode weergegeven worden zoals deze is ingesteld in het Logboek boodschappen venster. Als hierin maar één revisie wordt weergegeven, dan zullen de statistieken niet veel vertellen.

#### 4.10.10. Offline Modus



**Afbeelding 4.27. Schakel naar Offline werken dialoog**

Als de server niet bereikbaar is en je hebt log caching aangezet, dan kun je het logboek boodschappen venster en de revisie grafieken in offline modus gebruiken. Hiervoor worden de gegevens uit de cache gebruikt, waardoor het mogelijk is om door te werken ook al is de information niet bijgewerkt of misschien zelfs incompleet.

Je hebt hier drie mogelijkheden:

##### Tijdelijk offline

Maak de huidige actie af in offline modus, maar probeer het archief opnieuw te bereiken als er weer informatie opgevraagd wordt.

##### Permanent offline

Blijf in offline modus werken, totdat er een specifieke actie op het archief gevraagd wordt. Lees ook [Paragraaf 4.10.11, "De weergave verversen"](#).

##### Annuleren

Als je niet verder wilt gaan met de actie met mogelijk achterhaalde informatie, selecteer dan annuleren.

Met het **Maak dit de standaard** selectievak voorkom je dat dit venster telkens weer terug komt en selecteert altijd de optie die je eerder hebt gekozen. Je kunt je keuze later altijd nog wijzigen via TortoiseSVN → Instellingen.

#### 4.10.11. De weergave verversen

Als je wilt kijken of er nieuwe logboek boodschappen zijn op de server, dan kun je het beeld eenvoudig verversen door op **F5** te drukken. Als je gebruikt maakt van de log cache (standaard ingeschakeld), dan zal het archief op nieuwe berichten worden gecontroleerd en worden alleen de nieuwe berichten opgehaald. Als de log cache offline was, dan wordt er ook meteen geprobeerd om weer online te gaan.

Als je de log cache gebruikt en je vermoed dat de inhoud van een bericht of de auteur gewijzigd is, dan kun je **Shift-F5** of **Ctrl-F5** gebruiken om de weergegeven berichten opnieuw van de server op te halen en de log cache bij te werken. Merk op dat hiermee alleen de getoonde berichten worden verversen, de rest van de cache van het archief wordt hierdoor niet gewijzigd of gecontroleerd.



## 4.11. Tonen van Verschillen

Een van de meest voorkomende eisen in de ontwikkeling van projecten is om te kunnen zien wat er is veranderd. Je zou bijvoorbeeld de verschillen tussen twee herzieningen van hetzelfde bestand, of de verschillen tussen twee aparte bestanden willen bekijken. TortoiseSVN biedt een ingebouwd gereedschap genaamd TortoiseMerge om de verschillen tussen tekstbestanden te kunnen bekijken. Voor het vergelijken van afbeeldingen heeft TortoiseSVN ook een gereedschap genaamd TortoiseIDiff. Natuurlijk kun je hier ook je eigen favoriete programma voor gebruiken als je wilt.

### 4.11.1. Bestandsverschillen

#### Lokale wijzigingen

Als je wilt bekijken welke veranderingen *jij* gedaan hebt in je werkkopie, gebruik je simpelweg het context menu van de verkenner en selecteert TortoiseSVN → Verschil.

#### Verschil met een andere tak/label

Als je wilt zien wat er veranderd is in de basislijn (als je aan een tak werkt) of in een specifieke tak (als je werkt op de basislijn), kun je gebruik maken van het context menu van de verkenner. Je houdt gewoon de **Shift**-toets ingedrukt terwijl je met de rechtermuisknop op het bestand klikt. Vervolgens selecteer je TortoiseSVN → Verschil met URL. In het dialoogvenster dat volgt, geef je de URL in het archief aan waarmee je het lokale bestand wilt vergelijken.

Je kunt ook gebruik maken van de archief-verkenner. Selecteer hier twee bomen, twee labels, of een tak/label en de basislijn om te vergelijken. Via het daar aanwezige contextmenu kun je ze vergelijken met Vergelijk revisies. Lees meer in [Paragraaf 4.11.3, “Vergelijken van mappen”](#).

#### Verschil met een vorige revisie

Als je het verschil wilt zien tussen een bepaalde revisie en je werkkopie, gebruik je het Logboek dialoogvenster, selecteer je de gewenste revisie, en kies je vervolgens Vergelijk met werkkopie uit het contextmenu.

Als je het verschil wilt zien tussen de laatst vastgelegde revisie en je werkkopie, ervan uitgaande dat de werkkopie niet is gewijzigd, klik je met de rechtermuisknop op het bestand. Vervolgens selecteer je TortoiseSVN → Vergelijk met vorige revisie. Dit zal een vergelijking uitvoeren tussen de revisie voor de laatste-vastleg-datum (zoals vastgelegd in je werkkopie) en de werkbasis. Deze toont je de laatste wijziging uitgevoerd op het bestand welke nodig is om deze in de staat te brengen welke je in je werkkopie ziet. Het zal geen wijzigingen tonen die nieuwer zijn dan je werkkopie.

#### Verschil tussen twee eerdere revisies

Als je het verschil wil zien tussen twee revisies die al vastgelegd zijn, gebruik dan het Revisie Log dialoog en selecteer de twee revisies die je wil vergelijken (met behulp van de gebruikelijke **Ctrl**-wijziger). Selecteer daarna Vergelijk revisies uit het contextmenu.

Als je dit gedaan hebt vanuit het revisie log van een map, verschijnt er een Vergelijk Revisies dialoog welke een lijst toont met gewijzigde bestanden uit deze map. Lees meer in [Paragraaf 4.11.3, “Vergelijken van mappen”](#).

#### Alle wijzigingen die zijn gemaakt in een vastlegging

Wanneer je de wijzigingen van alle bestanden in een bepaalde revisie in één weergave wil zien, kun je hier de Unified-Diff uitvoer (GNU patch formaat) voor gebruiken. Deze toont alleen de wijzigingen met een aantal regels voor de context. Het is moeilijker te lezen als een visuele vergelijking, maar het toont alle wijzigingen bij elkaar. Selecteer vanuit het revisie log dialoog de gewenste revisie en selecteer dan Toon verschillen als unified diff uit the contextmenu.

#### Verschillen tussen bestanden

Als je de verschillen tussen twee verschillende bestanden wil bekijken, dan kun je dat rechtstreeks via de verkenner doen door beide bestanden te selecteren (gebruik makend van de standaard **Ctrl**-wijziger). Selecteer daarna uit het verkenner contextmenu TortoiseSVN → Verschil.

If the files to compare are not located in the same folder, use the command TortoiseSVN → Diff later to mark the first file for diffing, then browse to the second file and use TortoiseSVN → Diff with "path/of/ marked/file". To remove the marked file, use the command TortoiseSVN → Diff later again, but hold down the **Ctrl**-modifier while clicking on it.

#### Verschil tussen een werkkopie bestand/map en een URL

Als je de verschillen wil zien tussen een bestand in je werkkopie en een bestand in een willekeurig Subversion archief, kun je dat rechtstreeks vanuit de verkenner doen door het bestand te selecteren en terwijl je de **Shift** ingedrukt houdt met de rechter muisknop erop klikt zodat het contextmenu verschijnt. Selecteer dan TortoiseSVN → Verschil met URL. Je kan hetzelfde doen met een folder uit je werkkopie. TortoiseMerge toont deze verschillen op dezelfde manier als patch bestanden worden getoond - een lijst met gewijzigde bestanden die je in één keer kan zien.

#### Verschil met verklaringen

Als je niet alleen de verschillen, maar ook de auteur, de revisie en de wijzigingen die gemaakt zijn wil zien, kun je de verschil en verklaar rapporten vanuit het revisie log dialoog combineren. Ga voor meer informatie naar [Paragraaf 4.24.2, "Verklaar Verschillen"](#).

#### Verschillen tussen mappen

Met de ingebouwde gereedschappen die met TortoiseSVN worden meegeleverd is het niet mogelijk om verschillen tussen map hiërarchieën te bekijken, maar als je een extern gereedschap hebt welke dat ondersteund, kun je deze hier ook voor gebruiken. In [Paragraaf 4.11.6, "Externe Verschil/Samenvoeg gereedschappen"](#) worden een aantal gereedschappen beschreven die wij hebben gebruikt.

Als je een verschil gereedschap van een externe partij hebt geconfigureerd, kun je **Shift** gebruiken bij het selecteren van het Verschil commando om het alternatieve gereedschap te gebruiken. Lees [Paragraaf 4.31.5, "Externe Programma Instellingen"](#) voor meer informatie over het configureren van alternatieve verschil gereedschappen.

### 4.11.2. Opties voor regeleinden, spaties en witregels

Gedurende de loop van een project, zou je kunnen beslissen om de regeleinden te veranderen van CRLF naar LF, of je zou het inspringen van een sectie kunnen aanpassen. Dit zal jammer genoeg een groot aantal bestanden markeren als gewijzigd, ook al is er geen verandering in de betekenis van de code. De opties hieronder helpen je om deze wijzigingen te hanteren tijdens het vergelijken en toepassen van verschillen. Je komt deze instellingen tegen in de Samenvoeg en Verklaar dialogen, maar ook bij de instellingen van TortoiseMerge.

Negeer regeleinden negeert wijzigingen die uitsluitend veroorzaakt worden door een verschil in stijl van regeleinden.

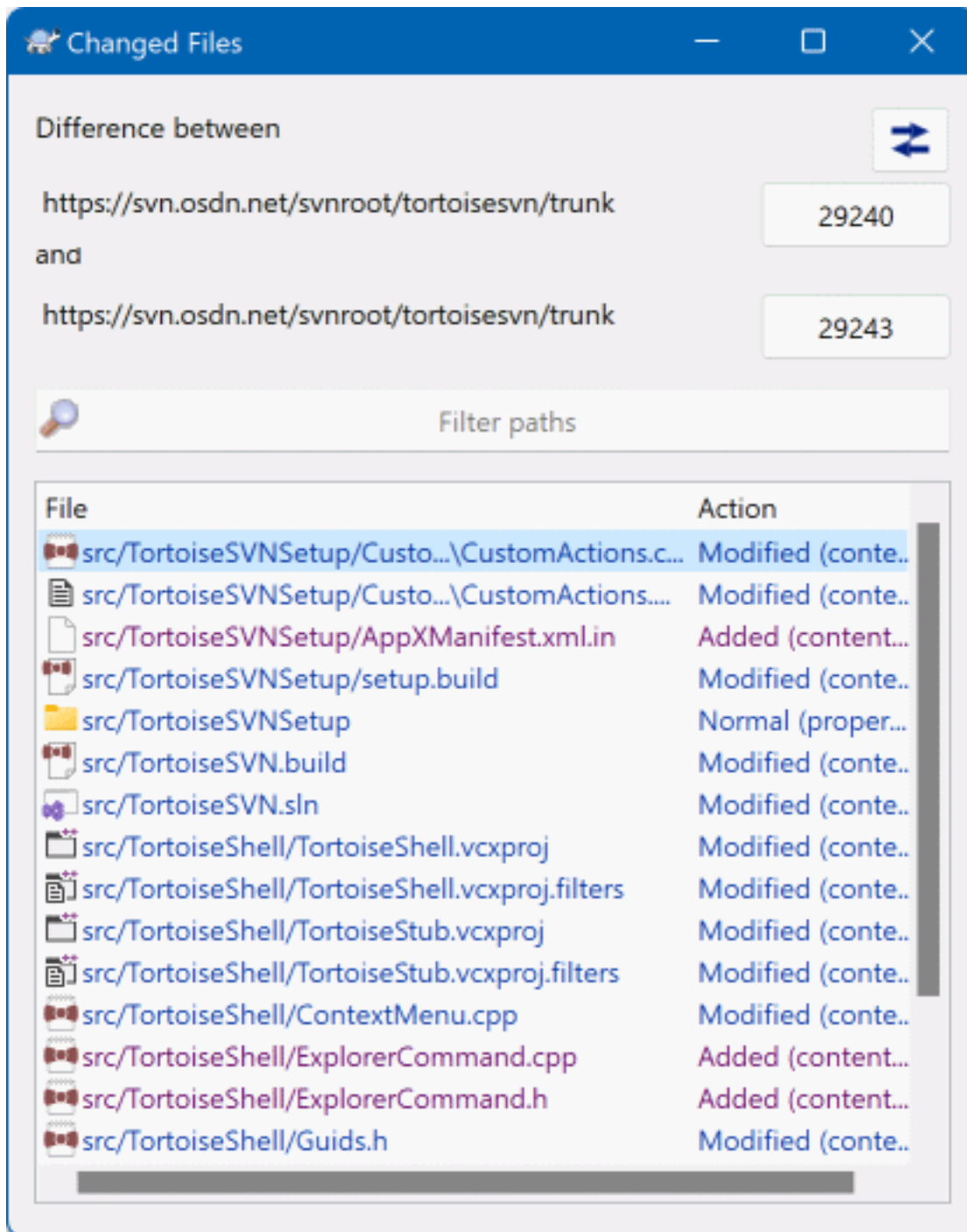
Vergelijk witregels toont alle wijzigingen in inspringen en witregels als toegevoegde/verwijderde regels.

Ignore whitespace changes excludes changes which are due solely to a change in the amount or type of whitespace, e.g. changing the indentation or changing tabs to spaces. Adding whitespace where there was none before, or removing a whitespace completely is still shown as a change.

Alle spaties en witregels negeren sluit alles uit waar alleen wijzigingen aan spaties en/of witregels zijn gemaakt.

Natuurlijk wordt elke regel met gewijzigde inhoud altijd aan het verschil toegevoegd.

### 4.11.3. Vergelijken van mappen



#### Afbeelding 4.28. De Vergelijk Revisies Dialoog

Als je twee bomen selecteert in de archief verkenner, of wanneer je twee revisies van een map selecteert in het log dialoog, kun je Contextmenu → Vergelijk revisies.

Deze dialoog toont een lijst met alle bestanden die zijn veranderd en staat je toen om deze individueel te vergelijken of te verklaren via het context menu.

Je kan een *wijzigings boom* exporteren, welke handig kan zijn als je iemand anders de boomstructuur van je project wil sturen, maar daar alleen de gewijzigde bestanden in wilt hebben. Deze operatie werkt alleen op de geselecteerde bestanden, dus dien je de gewenste bestanden - meestal betekent dit alle bestanden - te selecteren en daarna Contextmenu → Exporteer selectie naar.... Je krijgt de vraag op welke locatie je deze wijzigingsboom wilt opslaan.

Je kan de *lijst* met gewijzigde bestanden ook exporteren naar een tekstbestand via Contextmenu → Sla lijst van geselecteerde bestanden op in....

Als je de lijst met bestanden *en* de bijbehorende acties (gewijzigd, toegevoegd, verwijderd) wil exporteren, kun je dat doen via Contextmenu → Naar klembord kopiëren.

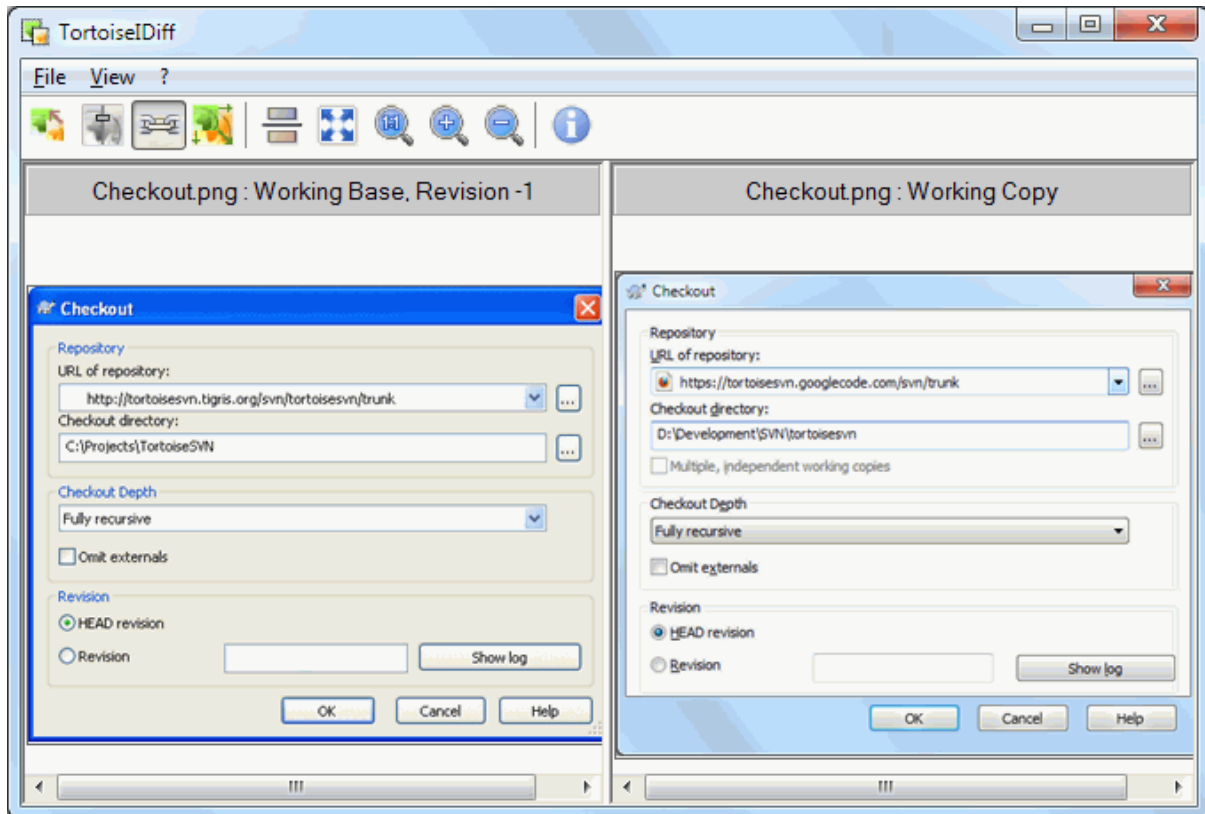
Met de knop bovenaan kun je de richting waarin vergeleken wordt aanpassen. Je kan de wijzigingen zien die nodig zijn om van A naar B te komen, maar als je dat liever hebt, van B naar A.

De knoppen met de revisie nummers erop kunnen gebruikt worden om het revisie bereik te veranderen. Als je dit bereik verandert, zal de lijst met items die verschillen tussen deze twee revisies automatisch ververs worden.

Als de lijst met bestandsnamen erg lang is, kun je het zoekvak gebruiken om de lijst te reduceren tot bestandsnamen die een specifieke tekst bevatten. Bedenk hierbij dat een simpele manier van zoeken op tekst wordt gebruikt, dus als je de lijst wil beperken tot C bron bestanden, zul je moeten zoeken op `.c` in plaats van `*.c`.

#### 4.11.4. Vergelijk afbeeldingen met TortoiseIDiff

Er zijn veel gereedschappen beschikbaar waarmee verschillen tussen tekst bestanden bekeken kunnen worden, inclusief onze eigen TortoiseMerge, maar vaak willen we ook kunnen zien hoe een afbeelding veranderd is. Daarom hebben we TortoiseIDiff gemaakt.



#### Afbeelding 4.29. De beeld-verschil viewer

TortoiseSVN → Verschil uitvoeren op één van de meest voorkomende formaten voor afbeeldingen zal TortoiseIDiff starten om de verschillen tussen afbeeldingen te tonen. Standaard zullen de afbeeldingen naast elkaar worden getoond, maar je kan via het View menu of de werkbalk instellen dat ze boven elkaar getoond worden. Als je dat wilt, kun je ze zelfs over elkaar leggen en doen of je een lightbox gebruikt.

Natuurlijk kan je ook in en uit zoomen en de afbeelding verplaatsen. De afbeelding kan ook eenvoudig verplaatst worden door te slepen met de linker muisknop. Als je de optie **Koppel afbeeldingsposities** selecteert, dan worden de verplaats regelaars (schuifbalken, muiswiel) voor beide afbeeldingen gekoppeld.

Een afbeeldingsinformatie scherm toont details over de afbeelding, zoals de grootte in pixels, resolutie en kleurdiepte. Als dit scherm in de weg zit, gebruik dan **Beeld** → **Afbeelding Info** om het te verbergen. Je krijgt dezelfde informatie in een tooltip wanneer je de muis over de titel van de afbeelding laat zweven.

Als de afbeeldingen over elkaar zijn gelegd, wordt de intensiteit van de afbeeldingen (alpha mix) bestuurd met de schuifbalk aan de linkerkant. Je kan op een willekeurige plek op de schuifbalk klikken om de mix direct te veranderen. Ook kan je de balk verschuiven om de mix interactief te veranderen.

De knop boven de schuifbalk schakelt tussen 0% mix en 100% mix. Als je dubbel klikt op de knop, zal de mix elke seconde automatisch schakelen totdat je nog een keer op de knop klikt. Dit kan handig zijn als je verschillende kleine wijzigingen zoekt.

Soms wil je een verschil zien in plaats van een mix. Je zou bijvoorbeeld van twee herzieningen van een afbeelding voor een printplaat willen zien welke paden zijn gewijzigd. Als je de alpha-mix modus uitzet, zal het verschil getoond worden als een *XOR* van de kleur van de pixel waarden. Ongewijzigde gebieden zullen getoond worden als wit en veranderingen zullen worden gekleurd.

### 4.11.5. Diffing Office Documents

When you want to diff non-text documents you normally have to use the software used to create the document as it understands the file format. For the commonly used Microsoft Office and Open Office suites there is indeed some support for viewing differences and TortoiseSVN includes scripts to invoke these with the right settings when you diff files with the well-known file extensions. You can check which file extensions are supported and add your own by going to TortoiseSVN → Settings and clicking Advanced in the External Programs section.



#### Problemen met Office 2010

If you installed the *Click-to-Run* version of Office 2010 and you try to diff documents you may get an error message from Windows Script Host something like this: “ActiveX component can't create object: word.Application”. It seems you have to use the MSI-based version of Office to get the diff functionality.

### 4.11.6. Externe Verschil/Samenvoeg gereedschappen

Als de gereedschappen die wij aanbieden niet doen wat je nodig hebt, kun je één van de vele beschikbare open-source of commerciële programma's gebruiken. Iedereen heeft zo zijn eigen favoriet, en deze lijst is verre van compleet, maar hier zijn er een aantal die je zou kunnen overwegen:

#### WinMerge

[WinMerge](https://winmerge.org/) [https://winmerge.org/] is a great open-source diff tool which can also handle directories.

#### Perforce Merge

Perforce is a commercial RCS, but you can download the diff/merge tool for free. Get more information from [Perforce](https://www.perforce.com/perforce/products/merge.html) [https://www.perforce.com/perforce/products/merge.html].

#### KDiff3

KDiff3 is een gratis verschil gereedschap welke ook mappen kan vergelijken. Je kan het [hier](http://kdiff3.sf.net/) [http://kdiff3.sf.net/] downloaden.

#### SourceGear DiffMerge

SourceGear Vault is a commercial RCS, but you can download the diff/merge tool for free. Get more information from [SourceGear](https://www.sourcegear.com/diffmerge/) [https://www.sourcegear.com/diffmerge/].

#### ExamDiff

ExamDiff Standard is freeware. Het kan omgaan met bestanden, maar niet met mappen. ExamDiff Pro is shareware en voegt een aantal 'goodies' toe waaronder het vergelijken van mappen en een bewerken functionaliteit. Beide smaken, versie 3.2 en hoger, kunnen omgaan met unicode. Je kan ze downloaden van [PrestoSoft](http://www.prestosoft.com/) [http://www.prestosoft.com/].

### Beyond Compare

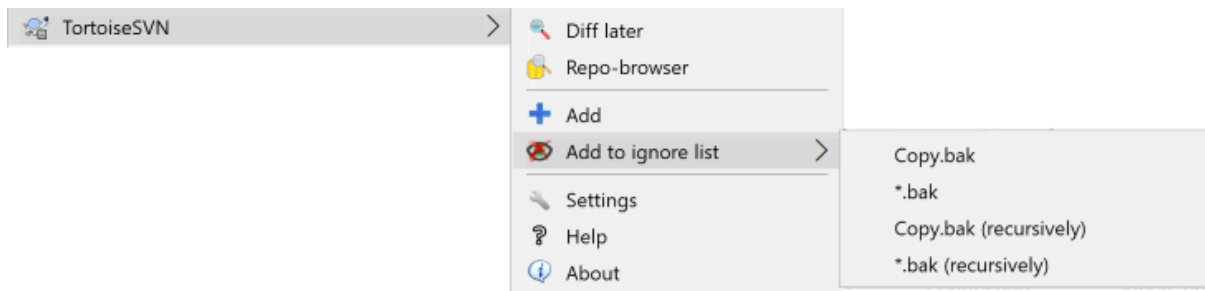
Similar to ExamDiff Pro, this is an excellent shareware diff tool which can handle directory diffs and unicode. Download it from [Scooter Software](https://www.scootersoftware.com/) [https://www.scootersoftware.com/].

### Araxis Merge

Araxis Merge is a useful commercial tool for diff and merging both files and folders. It does three-way comparison in merges and has synchronization links to use if you've changed the order of functions. Download it from [Araxis](https://www.araxis.com/merge/index.html) [https://www.araxis.com/merge/index.html].

Lees [Paragraaf 4.31.5, "Externe Programma Instellingen"](#) voor informatie over het gebruiken van deze gereedschappen binnen TortoiseSVN."

## 4.12. Nieuwe Bestanden En Mappen Toevoegen



### Afbeelding 4.30. Verkenner context menu voor bestanden zonder versiebeheer

Als je tijdens het ontwikkelproces nieuwe bestanden en/of mappen hebt gemaakt, moeten die ook onder broncodebeheer worden gebracht. Selecteer de bestanden en/of de mappen, en ga naar TortoiseSVN → Toevoegen.

Nadat je de bestanden of mappen onder versiebeheer hebt gebracht, verschijnt er een icoontje in de vorm van een plusje over het normale icoontje in Windows Verkenner. Dit betekent dat je de werkkopie van het bestand moet vastleggen voordat de bestanden/mappen beschikbaar zijn voor andere ontwikkelaars. Een bestand of map toevoegen heeft *geen* effect op het archief!



### Vele toevoegingen

Je kunt het commando 'Toevoegen' ook gebruiken op reeds geversioneerde mappen. In dat geval zal het toevoegen-dialoogscherm alle niet-geversioneerde bestanden in die geversioneerde mappen weergeven. Dit is handig als je veel nieuwe bestanden hebt en die allemaal in één keer wilt toevoegen.

Bestanden van buiten je werkkopie kun je daaraan toevoegen door middel van de volgende drag-and-drop-handeling:

1. selecteer de bestanden die je wilt toevoegen
2. right drag them to the new location inside the working copy
3. laat de rechter muisknop los
4. ga naar Contextmenu → SVN Kopiëren en toevoegen aan deze werkkopie. De bestanden worden dan naar de werkkopie gekopieerd en onder versiebeheer gebracht.

Ook binnen een werkkopie kun je bestanden toevoegen. Dit doe je door de bestanden (of mappen) met de linkermuisknop op te pakken, naar het vastleggen-dialoogscherm te slepen en ze vervolgens los te laten.

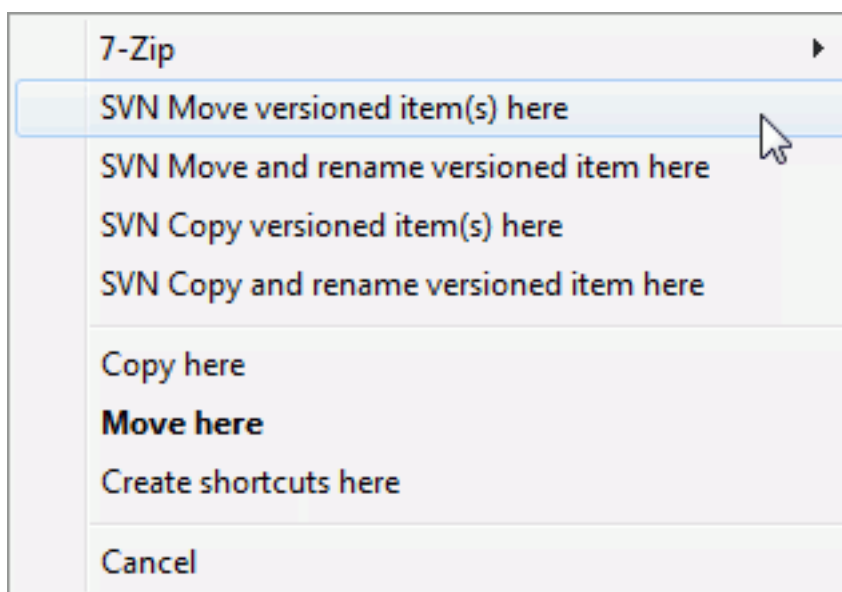


Het toevoegen van bestanden en mappen kun je, als je bijvoorbeeld per ongeluk een verkeerd bestand hebt toegevoegd, ongedaan maken voordat je je wijzigingen vastlegt. Dit doe je door het volgende uit te voeren: TortoiseSVN → Toevoeging opheffen....

## 4.13. Kopiëren/Verplaatsen/Hernoemen Bestanden en Mappen

It often happens that you already have the files you need in another project in your repository, and you simply want to copy them across. You could simply copy the files and add them, but that would not give you any history. And if you subsequently fix a bug in the original files, you can only merge the fix automatically if the new copy is related to the original in Subversion.

The easiest way to copy files and folders from within a working copy is to use the right drag menu. When you right drag a file or folder from one working copy to another, or even within the same folder, a context menu appears when you release the mouse.



**Afbeelding 4.31. Rechts-slepen-menu voor een map met versiebeheer**

Now you can copy existing versioned content to a new location, possibly renaming it at the same time.

U kunt ook bestanden met versiebeheer binnen een werkkopie kopiëren of verplaatsen, of tussen twee werkkopieën, met behulp van de gebruikelijke knip-en-plak methode. Gebruik de standaard Windows Kopiëren of Knippen om een of meerdere geversioneerde items naar het klipbord te kopiëren. Als het klipbord zulke geversioneerde items bevat, kunt u vervolgens TortoiseSVN → Plakken gebruiken (noot: niet de standaard Windows Plakken) om de items naar de nieuwe werkkopie locatie te kopiëren of verplaatsen.

U kunt bestanden en mappen van uw werkkopie naar een andere locatie in het archief kopiëren door middel van TortoiseSVN → Tak/Label. Raadpleeg [Paragraaf 4.20.1, “Aanmaken van Tak of Label”](#) voor meer informatie.

U kunt een oudere versie van een bestand of map vinden in het log dialoog en deze direct vanuit het log dialoog kopiëren naar een nieuwe locatie in het archief met behulp van Context menu → Maak tak/label van revisie. Raadpleeg [Paragraaf 4.10.3, “Extra Informatie Opvragen”](#) voor meer informatie.

U kunt ook de archief-verkenner gebruiken om de gewenste inhoud te vinden, en deze direct vanuit het archief naar uw werkkopie kopiëren, of tussen twee locaties binnen het archief te kopiëren. Raadpleeg [Paragraaf 4.25, “De Archief-verkenner”](#) voor meer informatie.

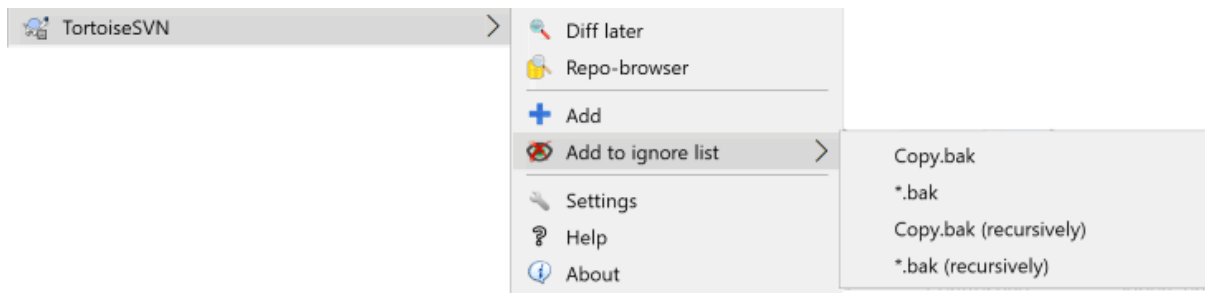


## Kan niet tussen archieven kopiëren

Hoewel je bestanden en mappen kunt kopiëren of verplaatsen *binnen* in een archief, kun je bestanden en mappen *niet* kopiëren tussen twee archieven zonder dat de geschiedenis verloren gaat. Zelfs niet als de archieven op dezelfde server staan. Het enige wat je kunt doen is de inhoud in de huidige staat kopiëren en als nieuwe inhoud aan het tweede archief toevoegen.

Als u er niet zeker van bent of twee URL's op dezelfde server naar dezelfde of verschillende archieven verwijzen, gebruik dan de archiefverkenner om één URL te openen en bepaal hiermee waar de bron van het archief is. Als u beide locaties kunt herkennen in de één archiefverkennervenster, dan bevinden ze zich in hetzelfde archief.

## 4.14. Negeren bestanden en mappen



**Afbeelding 4.32. Verkenner context menu voor bestanden zonder versiebeheer**

In de meeste projecten zijn er ook bestanden of mappen die je niet onder controle van subversion wilt brengen. Denk bijvoorbeeld aan compiler bestanden, `*.obj`, `*.lst`, of output bestanden en backup versies. Als je wijzigingen commit, laat TortoiseSVN jou de bestanden zien die (nog) niet meegenomen zullen worden. Deze kunnen je dialoog box vullen. Natuurlijk je de weergave ervan uitzetten, maar dan zou je belangrijke bestanden kunnen gaan vergeten.

De beste manier om deze problemen te vermijden is om een bestanden en/of extensies aan de negerlijst toe te voegen. Zo kunnen ze nooit per ongeluk gecommit worden of de weergave in de commit dialog box overvol maken zodat je andere belangrijke bestanden voor versie beheer zou vergeten.

If you right click on a single unversioned file, and select the command TortoiseSVN → Add to Ignore List from the context menu, a submenu appears allowing you to select just that file, or all files with the same extension. Both submenus also have a (recursively) equivalent. If you select multiple files, there is no submenu and you can only add those specific files/folders.

If you choose the (recursively) version of the ignore context menu, the item will be ignored not just for the selected folder but all subfolders as well. However this requires SVN clients version 1.8 or higher.

Wil je items van je neger-lijst afhalen, selecteer dan die bestanden of mappen, rechts klik op de selectie en kies TortoiseSVN → Verwijder van neger-lijst. Voor mappen kun je ook de `svn:ignore` eigenschap direct zetten. Daarmee kun je algemene patronen specificeren van te negeren bestanden, zoals hieronder beschreven zal worden. Lees [Paragraaf 4.18, "Project Instellingen"](#) voor meer informatie over het zetten van eigenschappen op mappen. Let op: iedere neger-patroon moet op zijn eigen regel komen te staan. Scheiden met spaties werkt niet.



## De Globale Negeerlijst

Een andere methode om bestanden te negeren is door ze toe te voegen aan de *globale neger lijst*. Het grote verschil is dat de globale neger lijst een eigenschap van de gebruiker is. Deze lijst wordt



toegepast op *alle* Subversion projecten, maar dan alleen die op de PC van de gebruiker. Over het algemeen is het beter om de `svn:ignore` eigenschap te gebruiken waar mogelijk, omdat het toegepast kan worden op specifieke project gebieden en het wordt gebruikt bij iedereen die het project ophaalt. Lees [Paragraaf 4.31.1, “Algemene instellingen”](#) voor meer informatie.



## Negeren van geversioneerde items

Bestanden en mappen die al onder versiebeheer vallen kunnen nooit genegeerd worden. Dat zou subversion tegenspreken. Mocht je een bestand per ongeluk hebben toegevoegd, lees dan [Paragraaf B.8, “Negeer bestanden die al opgenomen zijn in het versiebeheer”](#) om dit ongedaan te maken.

### 4.14.1. Patroonherkenning in negerlijsten

Subversion's neger patronen maken gebruik van globbing bij bestandsnamen, een techniek die van origine gebruikt is in Unix om bestanden aan te wijzen met meta-karakters als wildcards. De volgende karakters hebben een speciale betekenis:

\*

Matcht elke string van karakters, inclusief de lege string.

?

Matcht elk enkel letterteken.

[...]

Zoekt overeenkomsten met elk van de karakters welke zijn omsloten door blokhaken. Binnen de blokhaken, een karakterpaar gescheiden door “-” komen overeen met ieder karakter dat lexicaal tussen beide karakters in ligt. Bijvoorbeeld [AGm-p] komt overeen met elk van de karakters A, G, m, n, o of p.

Patroonherkenning is hoofdletter gevoelig, dus let op onder Windows. Je kunt ongevoeligheid voor hoofd- of kleine letters afdwingen door expliciet en uitpuddend letterparen te specificeren, bv. om `<literal>*.tmp</literal>` te negeren gebruik je het patroon `<literal>*. [Tt][Mm][Pp]</literal>`.

Voor de officiële definitie van globbing, raadpleeg de IEEE specificaties voor de 'shell command language' [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)].

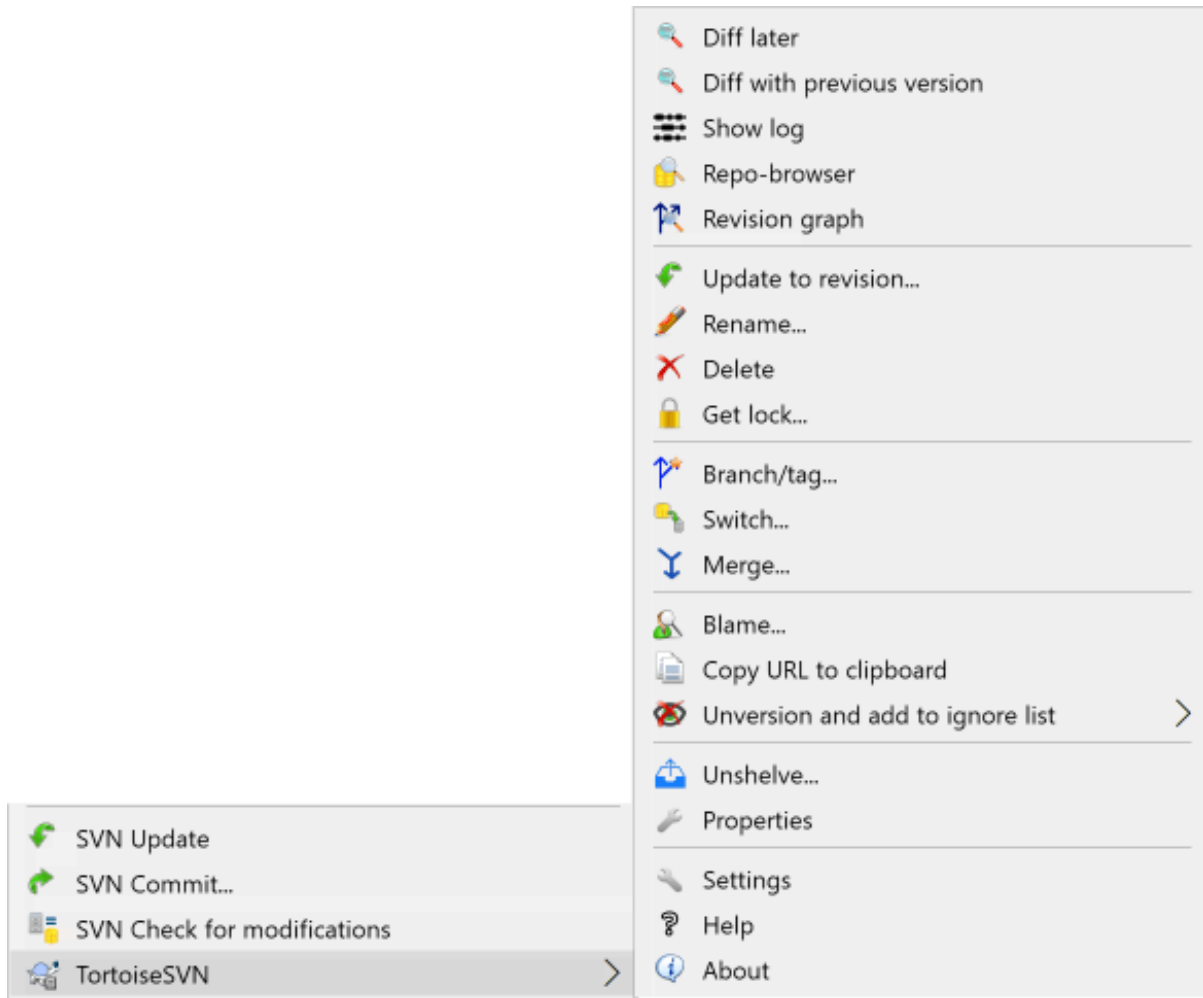


## Geen Paden in de Globale Negeerlijst

Je moet eigenlijk geen paden in je patronen opnemen. Het patroonherkenningsproces is bedoeld om te filteren op platte bestands- en mapnamen. Als je alle CVS mappen wilt negeren, voeg dan allen CVS toe aan de neger lijst. Het is niet nodig om `CVS */CVS` te specificeren, zoals je gewend was van eerder versies. Als je alle `tmp` mappen wilt negeren als ze in een `prog` map aanwezig zijn, maar niet in een `doc` map, dan zou je de `svn:ignore` eigenschap moeten gebruiken. Er is geen betrouwbare manier om dit gedrag te bereiken met globale neger patronen.

## 4.15. Verwijderen, Verplaatsen en Hernoemen

Subversion allows renaming and moving of files and folders. So there are menu entries for delete and rename in the TortoiseSVN submenu.



**Afbeelding 4.33. Verkenner context menu voor bestanden met versiebeheer**

#### 4.15.1. Verwijderen van bestanden en mappen

Use TortoiseSVN → Delete to remove files or folders from Subversion.

When you TortoiseSVN → Delete a file or folder, it is removed from your working copy immediately as well as being marked for deletion in the repository on next commit. The item's parent folder shows a “modified” icon overlay. Up until you commit the change, you can get the file back using TortoiseSVN → Revert on the parent folder.

Als je een item wilt verwijderen uit het archief, maar wel lokaal wil behouden als een bestand/map zonder versiebeheer, gebruik dan Uitgebreide Context Menu → Verwijderen (lokaal behouden). Hiervoor dien je de **Shift** toets ingedrukt te houden terwijl je rechts klikt op het item in het lijst venster van de verkenner (rechter venster) om het uitgebreide context menu te kunnen zien.

If an item is deleted via the explorer instead of using the TortoiseSVN context menu, the commit dialog shows those items as missing and lets you remove them from version control too before the commit. However, if you update your working copy, Subversion will spot the missing item and replace it with the latest version from the repository. If you need to delete a version-controlled file, always use TortoiseSVN → Delete so that Subversion doesn't have to guess what you really want to do.



## Een verwijderd bestand of map terughalen

If you have deleted a file or a folder and already committed that delete operation to the repository, then a normal TortoiseSVN → Revert can't bring it back anymore. But the file or folder is not lost at all. If you know the revision the file or folder got deleted (if you don't, use the log dialog to find out) open the repository browser and switch to that revision. Then select the file or folder you deleted, right click and select Context Menu → Copy to... as the target for that copy operation select the path to your working copy.

### 4.15.2. Verplaatsen van bestanden en mappen

Wanneer je een bestand of map gewoonweg wil hernoemen, gebruik dan Context Menu → Hernoemen.... Voer de nieuwe naam voor het item in en je bent klaar.

Wanneer je bestanden binnen je werkkopie wilt verplaatsen, wellicht naar een andere submap, gebruik dan de rechter-muisknop slepen en plaatsen methode: <placeholder-1/>



## Vastleggen van de bovenliggende map

Omdat het hernoemen en verplaatsen uitgevoerd wordt als een verwijdering gevolgd door een toevoeging, moet je de bovenliggende map vastleggen zodat het verwijderde gedeelte van de hernoeming/verplaatsing in het vastleggen dialoog verschijnt. Als je het verwijderde gedeelte niet vastlegt, zal het in het archief achterblijven en wanneer je collega's verversen, zal het oude bestand niet verwijderd zijn/worden. Zij zullen dus *beide* bestanden krijgen.

Je *moet* een naamswijziging in een map eerst vastleggen voordat je één van de bestanden in deze map gaat wijzigen, anders kan je werkkopie flink verknoeid worden.

Another way of moving or copying files is to use the Windows copy/cut commands. Select the files you want to copy, right click and choose Context Menu → Copy from the explorer context menu. Then browse to the target folder, right click and choose TortoiseSVN → Paste. For moving files, choose Context Menu → Cut instead of Context Menu → Copy.

Je kan ook de archief-verkenner gebruiken om objecten te verplaatsen. Lees [Paragraaf 4.25, "De Archief-verkenner"](#) voor meer informatie.



## SVN Verplaats Geen Externen

Je moet *niet* de TortoiseSVN Verplaatsen of Hernoemen commando's gebruiken op een map welke gemaakt is door gebruik te maken van `svn:externals`. Deze actie zal ervoor zorgen dat het externe item verwijderd wordt uit de bovenliggende map wat waarschijnlijk vele andere mensen van streek maakt. Als je een externe map moet verplaatsen, gebruik dan een gewone shell verplaatsing. Wijzig daarna de `svn:externals` eigenschappen van de bovenliggende mappen van de bron en het doel.

### 4.15.3. Omgaan met conflicten in hoofdlettergebruik in bestandsnamen

Als het archief al twee bestanden bevat met dezelfde naam, maar verschillend in hoofd-/kleine letters (bijvoorbeeld TEST.TXT en test.txt), zal het niet mogelijk zijn om de bovenliggende map te verversen of op te halen op een Windows client. Terwijl Subversion hoofdlettergevoelige bestandsnamen ondersteunt, doet Windows dit niet.

Dit kan soms gebeuren wanneer twee mensen bestanden vastleggen, vanuit verschillende werkkopiën, welke toevallig dezelfde naam hebben, maar verschillen in hoofd-/kleine letter. Het kan ook gebeuren wanneer bestanden worden vastgelegd vanuit een hoofdlettergevoelig bestandssysteem, zoals Linux.

In dat geval zul je moeten beslissen welk van deze bestanden je wilt behouden en de andere zul je moeten verwijderen uit het archief (of hernoemen).



### Het voorkomen van twee bestanden met dezelfde naam

There is a server hook script available at: <https://svn.apache.org/repos/asf/subversion/trunk/contrib/hook-scripts/> that will prevent checkins which result in case conflicts.

#### 4.15.4. Herstellen van Bestandhernoeringen

Soms zal je vriendelijke IDE bestanden voor je hernoemen als onderdeel van een refactoring en dit natuurlijk niet doorgeven aan Subversion. Als je dan probeert je wijzigingen vast te leggen, zal de oude bestandsnaam te zien zijn als ontbrekend, terwijl de nieuwe getoond wordt als een bestand zonder versiebeheer. Je kan natuurlijk eenvoudig de nieuwe bestandsnaam aanvinken om het toe te voegen, maar omdat Subversion niet weet dat de twee bestanden gerelateerd zijn, verlies je wel alle achterliggende geschiedenis.

Een beter manier is om Subversion te vertellen dat deze wijziging feitelijk een hernoeming is, wat gedaan kan worden in de Vastleggen en Kijk of er updates zijn dialogen. Selecteer eenvoudig zowel de oude (ontbrekende) als de nieuwe (zonder versiebeheer) bestandsnaam en gebruik Context Menu → Repareer Verplaatsing om de twee bestanden te koppelen als een hernoeming.

#### 4.15.5. Verwijderen van Bestanden Zonder Versiebeheer

Normaal gesproken zul je je negerlijst zo opzetten dat alle gegenereerde bestanden door Subversion genegeerd worden. Maar wat als je al deze bestanden wilt verwijderen om een schone build te kunnen produceren. Meestal zal je dit in je makefile zetten, maar als je de makefile aan het debuggen bent, of wanneer je het bouwsysteem veranderd, is het handig om een manier te hebben om het dek schoon te vegen.

TortoiseSVN voorziet in zo'n optie via Uitgebreide Context Menu → Verwijder objecten zonder versiebeheer.... Je moet de **Shift** toets ingedrukt houden als je rechts klikt op een map in het lijstvenster (rechter venster) van de verkenner om deze optie te kunnen zien in het uitgebreide context menu. Er wordt een scherm getoond waarin alle bestanden zonder versiebeheer in je gehele werkkopie in een lijst weergegeven worden. Je kan hieruit de bestanden die verwijderd moeten worden selecteren of deselecteren.

Als zulke items worden verwijderd wordt de prullenbak gebruikt, zodat als u hier een vergissing maakt en een bestand verwijdert dat zou moeten zijn geversioneerde, kunt u het nog terugzetten.

#### 4.16. Wijzigingen ongedaan maken

Als je de wijzigingen die je sinds de laatste update in een bestand hebt gemaakt, ga je als volgt te werk. Selecteer het bestand, klik er met de rechter muisknop op zodat het contextmenu opspringt, en klik vervolgens op het commando TortoiseSVN → Ongedaan maken

If you also want to clear all the changelists that are set, check the box at the bottom of the dialog.

If you want to undo a deletion or a rename, you need to use Revert on the parent folder as the deleted item does not exist for you to right click on.

Als je het toevoegen van een bestand of map ongedaan wilt maken, dien je de volgende actie in het contextmenu uit te voeren: TortoiseSVN → Toevoegen opheffen.... Deze actie verschilt niet van de andere acties waarmee je handelingen ongedaan kunt maken, maar heeft een andere naam gekregen om het doel ervan te verhelderen.

Je kunt de kolommen in dit dialoogscherf configureren op dezelfde manier als die van het dialoogscherf Kijk of er updates zijn. Ga naar [Paragraaf 4.7.3, "Lokale en Server Status"](#) voor meer informatie.

Omdat terugzetten soms wordt gebruikt om een werkkopie op te schonen is er een extra knop welke de mogelijkheid biedt om ook niet-geversioneerde items te verwijderen. Als u op deze knop klikt wordt er een

andere dialoog getoond welke alle niet-geversioneerde items in een lijst toont, welke u dan voor verwijdering kunt selecteren.



### Reeds vastgelegde wijzigingen ongedaan maken

Ongedaan maken maakt alleen lokale wijzigingen ongedaan. Wijzigingen die al zijn vastgelegd, worden *niet* ongedaan gemaakt. Als je wijzigingen die in een bepaalde revisie zijn vastgelegd, ongedaan wilt maken, lees dan [Paragraaf 4.10, "Logboekboodschappen Venster"](#).

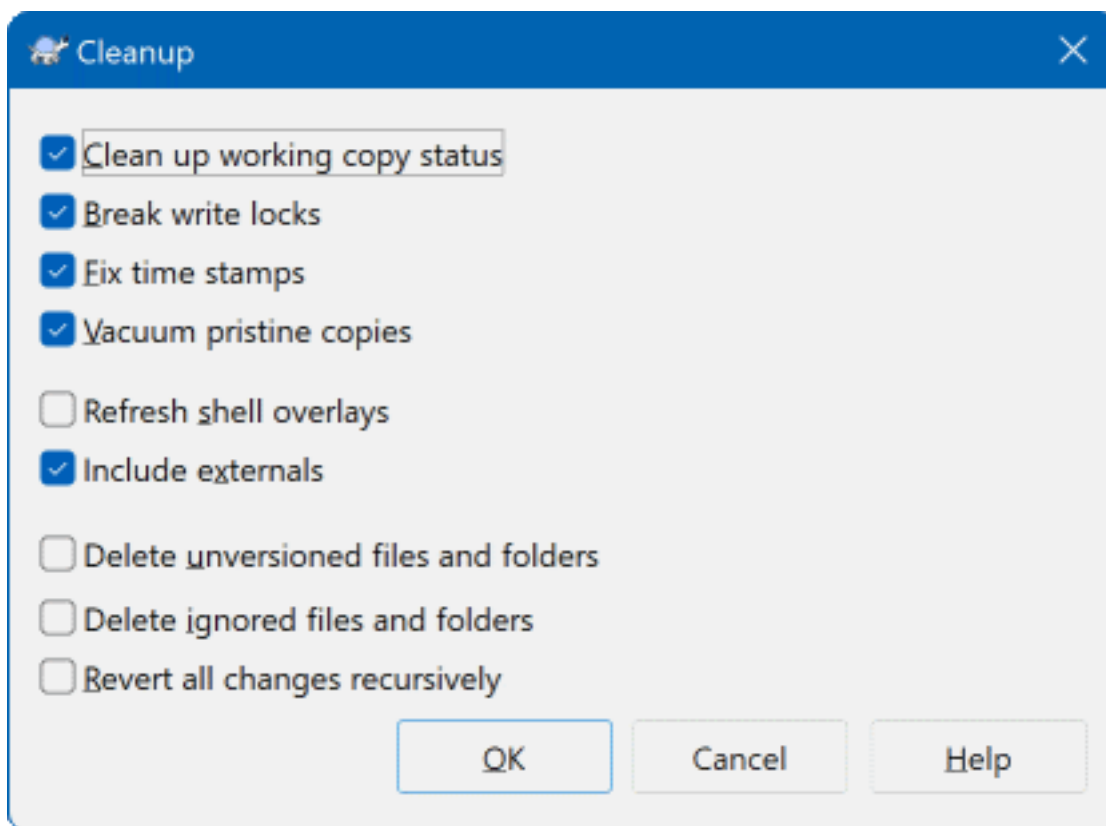


### Herstellen is langzaam

Als je wijzigingen ongedaan maakt, kan het gebeuren dat dit veel langer duurt dan verwacht. De reden hiervoor is dat de gewijzigde versie van het bestand wordt weggegooid en in de prullenbak terecht komt, zodat je je wijzigingen weer terug kunt halen als je TortoiseSVN per ongeluk instrueerde om die ongedaan te maken. Het kan echter voorkomen dat de prullenbak vol zit; in dat geval heeft Windows veel tijd nodig om een plaats voor het weggegooid bestand te vinden. Je kunt de lange wachttijd dus terugbrengen door de prullenbak te legen, of door de eigenschap **Gebruik prullenbak bij herstellen** in de instellingen van TortoiseSVN te deactiveren.

## 4.17. Schonen

If a Subversion command cannot complete successfully, perhaps due to server problems, your working copy can be left in an inconsistent state. In that case you need to use TortoiseSVN → Cleanup on the folder. It is a good idea to do this at the top level of the working copy.



**Afbeelding 4.35. The Cleanup dialog**

In the cleanup dialog, there are also other useful options to get the working copy into a clean state.

#### Clean up working copy status

As stated above, this option tries to get an inconsistent working copy into a workable and usable state. This doesn't affect any data you have but only the internal states of the working copy database. This is the actual `CleanUp` command you know from older TortoiseSVN clients or other SVN clients.

#### Break write locks

If checked, all write locks are removed from the working copy database. For most situations, this is required for the cleanup to work!

Only uncheck this option if the working copy is used by other users/clients at the time. But if the cleanup then fails, you have to check this option for the cleanup to succeed.

#### Fix time stamps

Adjusts the recorded time stamps of all files, speeding up future status checks. This can speed up all dialogs that show working copy file lists, for example the Commit dialog.

#### Vacuum pristine copies

Removes unused pristine copies and compresses all remaining pristine copies of working copy files.

#### Refresh shell overlays

Sometimes the shell overlays, especially on the tree view on the left side of the explorer don't show the current status, or the status cache failed to recognize changes. In this situation, you can use this command to force a refresh.

#### Include externals

If this is checked, then all actions are done for all files and folders included with the `svn:externals` property as well.

#### Delete unversioned files and folders, Delete ignored files and folders

This is a fast and easy way to remove all generated files in your working copy. All files and folders that are not versioned are moved to the trash bin.

Note: you can also do the same from the TortoiseSVN → Revert dialog. There you also get a list of all the unversioned files and folders to select for removal.

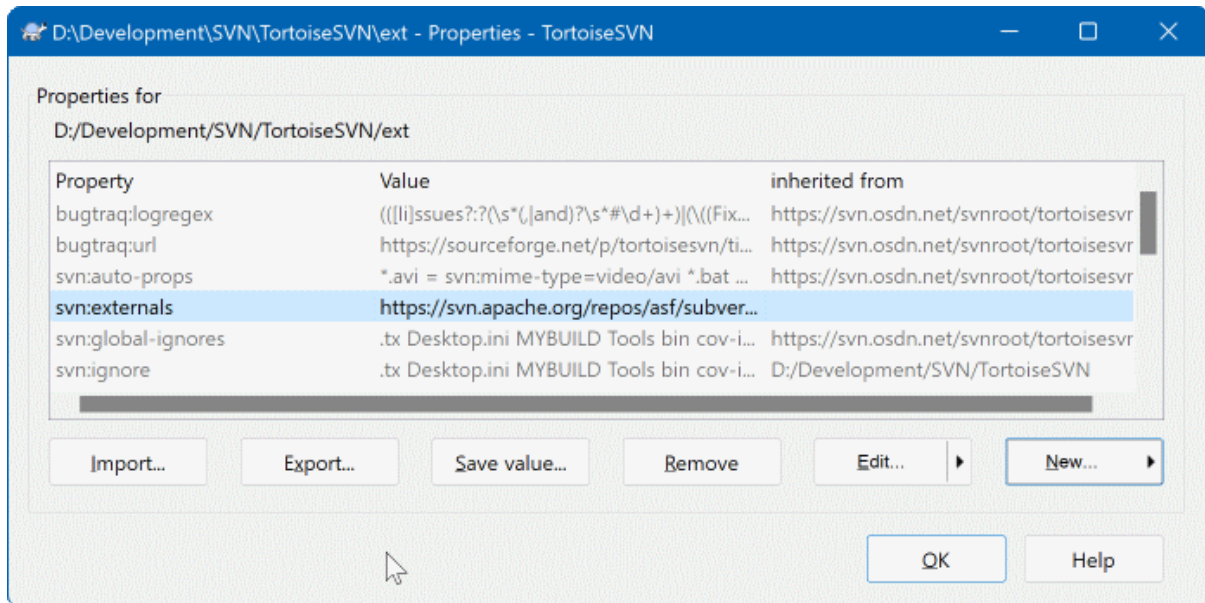
#### Revert all changes recursively

This command reverts all your local modifications which are not committed yet.

Note: it's better to use the TortoiseSVN → Revert command instead, because there you can first see and select the files which you want to revert.

## 4.18. Project Instellingen

### 4.18.1. Subversion Eigenschappen



**Afbeelding 4.36. Subversion kenmerken pagina**

U kunt de Subversion eigenschappen lezen en zetten vanuit het Windows eigenschappen dialoog, maar ook vanuit TortoiseSVN → eigenschappen en binnen TortoiseSVN's status lijsten, vanuit Context menu → eigenschappen.

U kunt uw eigen eigenschappen toevoegen, of enige eigenschappen met een speciale betekenis in Subversion. Deze beginnen met `svn:`. `svn:externals` is zo'n eigenschap; raadpleeg hoe externen af te handelen in [Paragraaf 4.19, "Externe Objecten"](#).

#### 4.18.1.1. svn:keywords

Subversion ondersteunt CVS-achtige trefwoord uitbreidingen welke kunnen worden gebruikt om bestandsnaam en revisie informatie in het bestand zelf in te bedden. Trefwoorden welke momenteel worden ondersteund zijn:

##### \$Date\$

Datum van laatste vastlegging. Dit is gebaseerd op informatie verkregen tijdens het bijwerken van uw werkkopie. Het controleert het archief *niet* op meer recente wijzigingen.

##### \$Revision\$

Revisie van laatst bekende vastlegging.

##### \$Author\$

Auteur welke de laatst bekende vastlegging heeft gemaakt.

##### \$HeadURL\$

De volledige URL van dit bestand in het archief.

##### \$Id\$

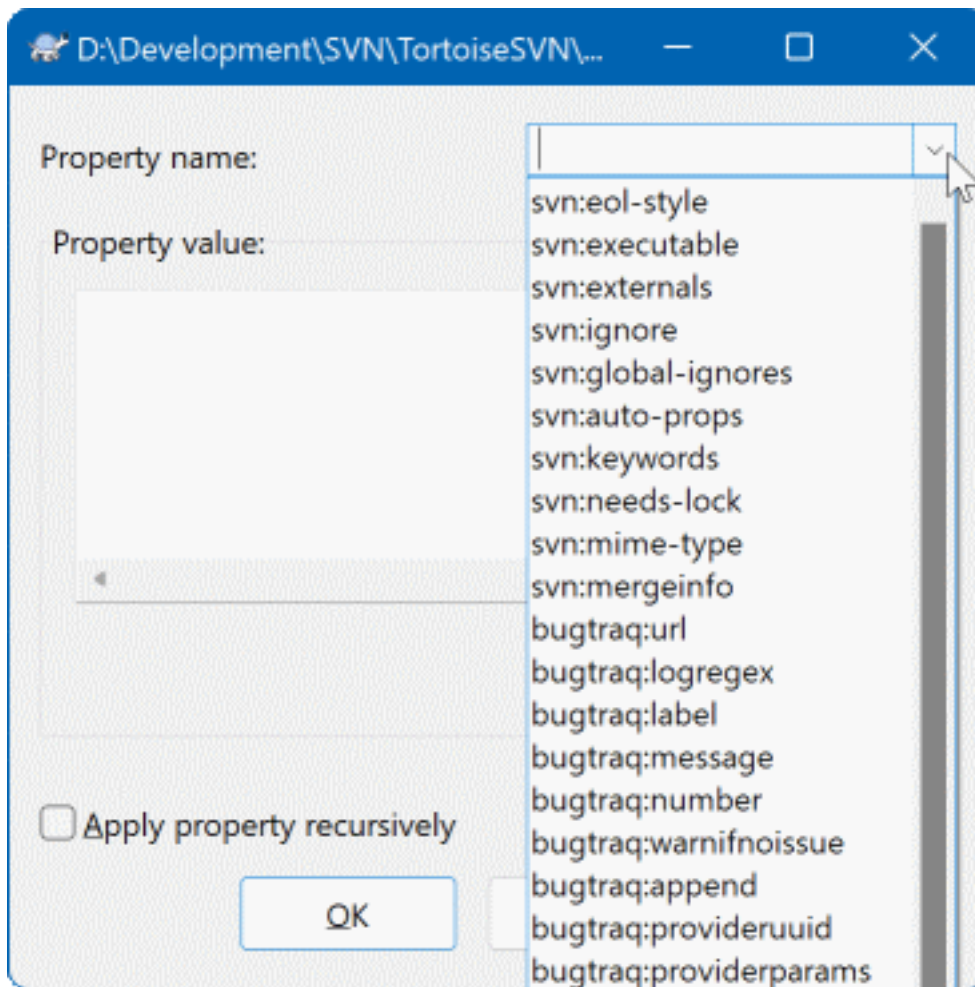
Een gekomprimeerde combinatie van de vorige vier trefwoorden.

To find out how to use these keywords, look at the [svn:keywords section](http://svnbook.red-bean.com/en/1.8/svn.advanced.props.special.keywords.html) [http://svnbook.red-bean.com/en/1.8/svn.advanced.props.special.keywords.html] in the Subversion book, which gives a full description of these keywords and how to enable and use them.

For more information about properties in Subversion see the [Special Properties](http://svnbook.red-bean.com/en/1.8/svn.advanced.props.html) [http://svnbook.red-bean.com/en/1.8/svn.advanced.props.html].



#### 4.18.1.2. Toevoegen en Aanpassen van Eigenschappen



**Afbeelding 4.37. Toevoegen van eigenschappen**

To add a new property, first click on **New...** Select the required property name from the menu, and then fill in the required information in the specific property dialog. These specific property dialogs are described in more detail in [Paragraaf 4.18.3, "Eigenschappen bewerken"](#).

To add a property that doesn't have its own dialog, choose **Advanced** from the **New...** menu. Then either select an existing property in the combo box or enter a custom property name.

If you want to apply a property to many items at once, select the files/folders in explorer, then select **Context menu** → **properties**.

Als je de eigenschappen wilt gebruiken voor *alle* bestanden en mappen in de onderliggende mappen van de huidige map, selecteer dan de optie **Rekursief**.

Als je een bestaande eigenschap wilt bewerken, selecteer dan die eigenschap uit de lijst van bestaande eigenschappen en klik dan op **Wijzigen...**

Als je een bestaande eigenschap wilt verwijderen, selecteer dan de eigenschap uit de lijst van bestaande eigenschappen en klik dan op **Verwijderen**.

De `svn:externals` eigenschap kan worden gebruikt om andere projecten uit hetzelfde archief of uit een compleet ander archief binnen te halen. Voor meer informatie, lees [Paragraaf 4.19, "Externe Objecten"](#).





## Bewerk eigenschappen in de HEAD-revisie

Because properties are versioned, you cannot edit the properties of previous revisions. If you look at properties from the log dialog, or from a non-HEAD revision in the repository browser, you will see a list of properties and values, but no edit controls.

### 4.18.1.3. Exporteren en Importeren van Eigenschappen

Je zult er achter komen dat je vaak dezelfde set van eigenschappen toepast, bijvoorbeeld `bugtraq:logregex`. Om het kopiëren van eigenschappen van het ene project naar het andere te vereenvoudigen, kun je gebruik maken van de Export/Import functionaliteit.

Kies voor het bestand of de map waar de eigenschap al ingesteld is TortoiseSVN → Eigenschappen en selecteer de eigenschappen die je wilt exporteren en klik op **Exporteer**. Je moet vervolgens de bestandsnaam opgeven voor het bestand waarin je namen en waardes wilt opslaan.

Kies vanuit de map(pen) waarop je deze eigenschappen wilt toepassen TortoiseSVN → Eigenschappen en klik op **Importeren...** Er zal dan gevraagd worden vanuit welk bestand er geïmporteerd moet worden. Blader naar de locatie waar je eerder het export bestand hebt opgeslagen en selecteer deze. De eigenschappen zullen niet-recursief worden toegepast op de mappen.

Als je eigenschappen recursief aan een mapstructuur wilt toevoegen, volg dan de hierboven beschreven stappen, selecteer in het eigenschappen scherm elke aparte eigenschap en klik op **Bewerken...** Vink de optie **Eigenschap recursief toepassen** aan en klik op **OK**.

De indeling van het import bestand is binair en een eigen indeling voor TortoiseSVN. Het enige nut van deze bestanden in het overzetten van eigenschappen met de Importeren en Exporteren functie. Er is dus geen reden om deze bestanden te kunnen bewerken.

### 4.18.1.4. Binaire eigenschappen

TortoiseSVN kan omgaan met binaire waardes van eigenschappen als bestanden worden gebruikt. Om een binaire waarde van een eigenschap te lezen, dan moet je deze **Opslaan...** in een bestand. Gebruik een Hex-editor of ander geschikt programma om een bestand aan te maken met de benodigde inhoud en kies dan **Laden...** om de inhoud van dit bestand op te halen.

Ondanks het feit dat binaire eigenschappen niet vaak gebruikt worden, kunnen ze in sommige gevallen erg handig zijn. Bijvoorbeeld als je hele grote grafische bestanden opslaat of de applicatie die gebruikt wordt om een bestand te laden erg groot is, dan wil je misschien een thumbnail opslaan als eigenschap, zodat je deze snel vooraf kunt bekijken.

### 4.18.1.5. Automatisch eigenschappen zetten

Je kunt Subversion en TortoiseSVN zo instellen dat ze automatisch eigenschappen instellen voor bestanden en mappen als deze worden toegevoegd aan het archief. Er zijn twee manieren om dit te doen.

You can edit the Subversion configuration file to enable this feature on your client. The **General** page of TortoiseSVN's settings dialog has an edit button to take you there directly. The config file is a simple text file which controls some of Subversion's workings. You need to change two things: firstly in the section headed `miscellany` uncomment the line `enable-auto-props = yes`. Secondly you need to edit the section below to define which properties you want added to which file types. This method is a standard Subversion feature and works with any Subversion client. However it has to be defined on each client individually - there is no way to propagate these settings from the repository.

Een alternatief is om de `tsvn:auto-props` eigenschap in te stellen op mappen, zoals omschreven in de volgende sectie. Deze methode werkt alleen met TortoiseSVN, maar zal wel propageren naar alle werkkopieën bij het verversen.

As of Subversion 1.8, you can also set the property `svn:auto-props` on the root folder. The property value is automatically inherited by all child items.

Whichever method you choose, you should note that auto-props are only applied to files at the time they are added to the working copy. Auto-props will never change the properties of files which are already versioned.

Als je er absoluut zeker van wilt zijn dat nieuwe bestanden de juist eigenschappen toegepast krijgen, moet je een pre-commit hook instellen, welke de vastlegging weigert als de benodigde eigenschappen niet ingesteld zijn.



### Vastleggen eigenschappen

Subversion eigenschappen zijn voorzien van versiebeheer. Als je een eigenschap wijzigt of toevoegt, moet je je wijzigingen vastleggen in het archief.



### Conflicten op eigenschappen

Als er een conflict optreedt bij het vastleggen van de wijzigingen, omdat een andere gebruiker dezelfde eigenschap heeft gewijzigd, maakt Subversion een `.prej>/filename>` bestand aan. Verwijder dit bestand nadat je het conflict opgelost hebt.

## 4.18.2. TortoiseSVN Project Eigenschappen

TortoiseSVN heeft een paar speciale eigenschappen van zichzelf, en deze beginnen met `tsvn:`.

- `tsvn:logminsize` stelt de minimum lengte in voor een logboek bericht bij een vastlegging. Als je minder karakters gebruikt dan gespecificeerd, dan kan de vastlegging niet afgerond worden. Deze functie is erg nuttig om je te helpen herinneren dat je een bericht achter laat met een goede beschrijving voor elke blokkade die je plaatst. Als deze eigenschap niet ingesteld is of op nul is ingesteld, dan worden lege logboek berichten toegestaan.

`tsvn:lockmsgminsize` stelt de minimum lengte in voor een blokkade bericht. Als je minder karakters gebruikt dan gespecificeerd, dan wordt de blokkade niet geplaatst. Deze functie is erg nuttig om je te helpen herinneren dat je een bericht achter laat met een goede beschrijving voor elke blokkade die je plaatst. Als deze eigenschap niet ingesteld is of op nul is ingesteld, dan worden lege blokkade berichten toegestaan.

- `tsvn:logwidthmarker` wordt gebruikt voor projecten waarbij het logboek bericht opgemaakt moet worden met een maximum breedte (meestal 80 karakters) voor een regeleinde. Het instellen van deze eigenschap op een waarde die niet nul is, resulteert in twee dingen in het logboek bericht venster: het plaatst een markering om de maximale breedte aan te geven en tekstterugloop wordt ugeschakeld, zodat je kunt zien of de tekst die je hebt ingevoerd te lang is. Noot: deze functie werkt alleen goed als je een lettertype hebt geselecteerd met een vaste tekenbreedte.
- `tsvn:logtemplate` wordt gebruikt voor project waarbij er regels zijn voor de indeling van het logboek bericht. De eigenschap bevat een meerregelige tekst die opgenomen wordt in het vastleg invoerveld als je een vastlegging start. Je kunt het vervolgens bewerken om de benodigde informatie in te vullen. Noot: als je ook `tsvn:logminsize` gebruikt, zorg er dan voor dat de minimale lengte hoger staat ingesteld dan het aantal karakters van de template, anders heeft die eigenschap helemaal geen nut meer.

There are also action specific templates which you can use instead of `tsvn:logtemplate`. The action specific templates are used if set, but `tsvn:logtemplate` will be used if no action specific template is set.

The action specific templates are:

- `tsvn:logtemplatecommit` wordt gebruikt voor alle commits van een werkkopie.
- `tsvn:logtemplatebranch` wordt gebruikt wanneer je een branch of tag aanmaakt, of wanneer je bestanden of mappen direct in de repository browser kopieert.
- `tsvn:logtemplateimport` wordt gebruikt voor imports.
- `tsvn:logtemplatedelete` wordt gebruikt wanneer je items direct in de repository browser verwijdert.

- `tsvn:logtemplatemove` wordt gebruikt wanneer items in de repository browser worden hernoemd of verplaatst.
- `tsvn:logtemplatemkdir` wordt gebruikt wanneer mappen in de repository browser worden aangemaakt.
- `tsvn:logtemplatepropset` wordt gebruikt bij het bewerken van eigenschappen in de repository browser.
- `tsvn:logtemplatelock` wordt gebruikt bij het aanvragen van een vergrendeling.
- Subversion allows you to set “autoprops” which will be applied to newly added or imported files, based on the file extension. This depends on every client having set appropriate autoprops in their Subversion configuration file. `tsvn:autoprops` can be set on folders and these will be merged with the user's local autoprops when importing or adding files. The format is the same as for Subversion autoprops, e.g. `*.sh = svn:eol-style=native;svn:executable` sets two properties on files with the `.sh` extension.

Als er een conflict is tussen de lokale autoprops en de `tsvn:autoprops`, dan krijgen de instellingen van het project voorrang, omdat die specifiek voor het project zijn.

As of Subversion 1.8, you should use the property `svn:auto-props` instead of `tsvn:autoprops` since this has the very same functionality but works with all svn clients and is not specific to TortoiseSVN.

- In het Vastleg scherm kun je de lijst met gewijzigde bestanden en de status (toegevoegd, gewijzigd, etc) invoegen. `tsvn:logfilelistenglish` definieert of de status van de bestanden in het Engels of in de lokale taal opgenomen moet worden. Deze eigenschap krijgt, als hiervoor geen andere waarde is ingesteld, de standaard waarde `true`.
- TortoiseSVN can use a spell checker. On Windows 10, the spell checker of the OS is used. On earlier Windows versions, it can use spell checker modules which are also used by OpenOffice and Mozilla. If you have those installed this property will determine which spell checker to use, i.e. in which language the log messages for your project should be written. `tsvn:projectlanguage` sets the language module the spell checking engine should use when you enter a log message. You can find the values for your language on this page: [MSDN: Language Identifiers](https://docs.microsoft.com/en-us/windows/desktop/intl/language-identifier-constants-and-strings) [https://docs.microsoft.com/en-us/windows/desktop/intl/language-identifier-constants-and-strings].

Je kunt deze waarde als decimale of hexadecimale, indien vooraf gegaan door `0x`, waarde ingevoerd worden. Bijvoorbeeld Engels (VS) kan ingevoerd worden als `0x0409` of als `1033`.

- De `tsvn:logsummary` eigenschap wordt gebruikt om een deel uit het logboek bericht te extraheren, welke dan getoond kan worden in het logboek venster als samenvatting.

De waarde van de `tsvn:logsummary` eigenschap moet een regelige reguliere expressie bevatten met één Regex-groep. De inhoud van die groep wordt als samenvatting gebruikt.

Een voorbeeld: `\[SUMMARY\]:\s+(.*)` Zal alles pakken na “[SUMMARY]” in het logbericht en zal dat gebruiken als de samenvatting.

- De eigenschap `tsvn:logregex` definieert een reguliere expressie, die overeenkomende revisies in een logboek bericht zoekt. Dit wordt gebruikt in het logboek venster om van zulke referenties koppelingen te maken. Als er op zo'n koppeling geklikt wordt, springt het scherm naar die revisie (als die revisie weergegeven wordt in het logboek venster of als het beschikbaar is in de log cache) of wordt er een nieuw scherm geopend met die revisie.

De reguliere expressie moet overeenkomen met de hele referentie, niet alleen het revisie nummer. Het revisie nummer wordt automatisch uit de gevonden referentie geëxtraheerd.

Als deze eigenschap niet ingesteld is, wordt een standaard reguliere expressie gebruikt om de revisie referenties te koppelen.

- There are several properties available to configure client-side hook scripts. Each property is for one specific hook script type.

The available properties/hook-scripts are

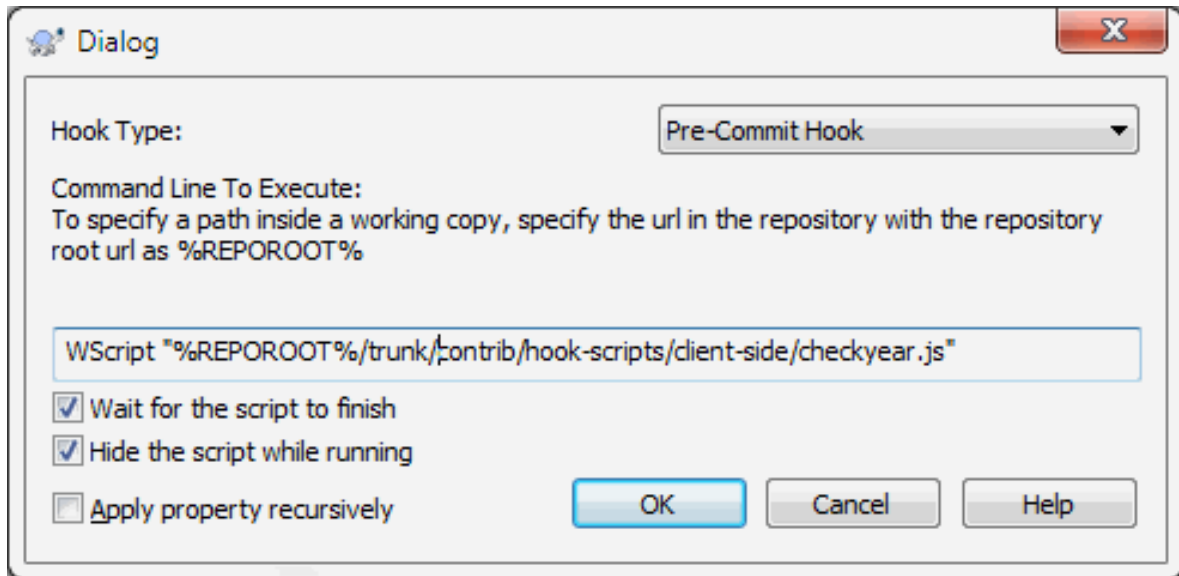
- `tsvn:startcommithook`
- `tsvn:precommithook`
- `tsvn:postcommithook`
- `tsvn:startupdatehook`
- `tsvn:preupdatehook`
- `tsvn:postupdatehook`
- `tsvn:prelockhook`
- `tsvn:postlockhook`

The parameters are the same as if you would configure the hook scripts in the settings dialog. See [Paragraaf 4.31.8, “Hook Scripts op de Client”](#) for the details.

Since not every user has his or her working copy checked out at the same location with the same name, you can configure a script/tool to execute that resides in your working copy by specifying the URL in the repository instead, using `%REPOROOT%` as the part of the URL to the repository root. For example, if your hook script is in your working copy under `contrib/hook-scripts/client-side/checkyear.js`, you would specify the path to the script as `%REPOROOT%/trunk/contrib/hook-scripts/client-side/checkyear.js`. This way even if you move your repository to another server you do not have to adjust the hook script properties.

Instead of `%REPOROOT%` you can also specify `%REPOROOT+%`. The `+` is used to insert any number of folder paths necessary to find the script. This is useful if you want to specify your script so that if you create a branch the script is still found even though the url of the working copy is now different. Using the example above, you would specify the path to the script as `%REPOROOT+%/contrib/hook-scripts/client-side/checkyear.js`.

The following screenshot shows how the script to check for current copyright years in source file headers is configured for TortoiseSVN.



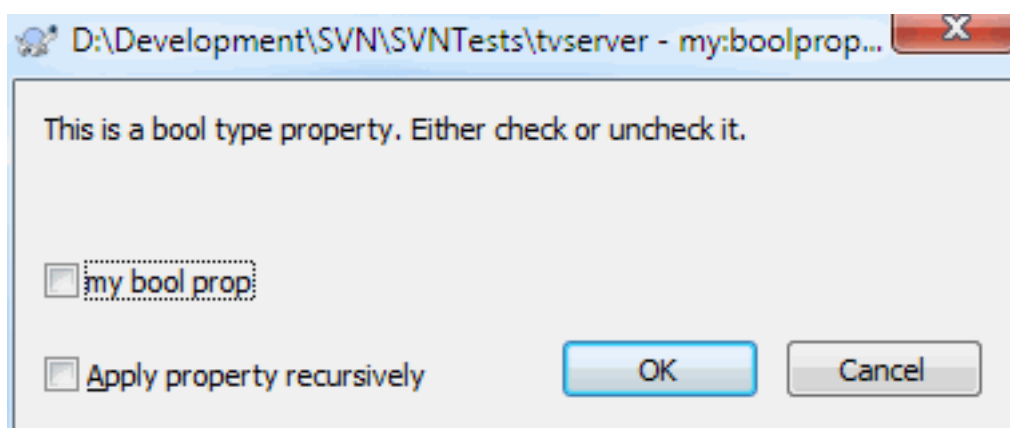
**Afbeelding 4.38. Property dialog for hook scripts**

- Als je een nieuwe eigenschap wilt toevoegen, dan kun je er eentje uit de lijst van de combo box kiezen of je kunt de gewenste naam voor de eigenschap invoeren. Als je project gebruik maakt van enkele eigen eigenschappen en je wilt dat deze in ook de lijst getoond worden (bijv. om typefouten te voorkomen), dan kun je een lijst met je eigen eigenschappen aanmaken door gebruik te maken van `tsvn:userfileproperties` en `tsvn:userdirproperties`. Pas deze eigenschappen toe op een map. Pas deze eigenschappen toe op een map. Als je dan de eigenschappen van onderliggende objecten bewerkt, dan worden je voorgedefinieerde eigenschappen weergegeven in de lijst met de namen van de beschikbare eigenschappen.

You can also specify whether a custom dialog is used to add/edit your property. TortoiseSVN offers four different dialog, depending on the type of your property.

bool

If your property can only have two states, e.g., true and false, then you can configure your property as a bool type.



**Afbeelding 4.39. Property dialog boolean user types**

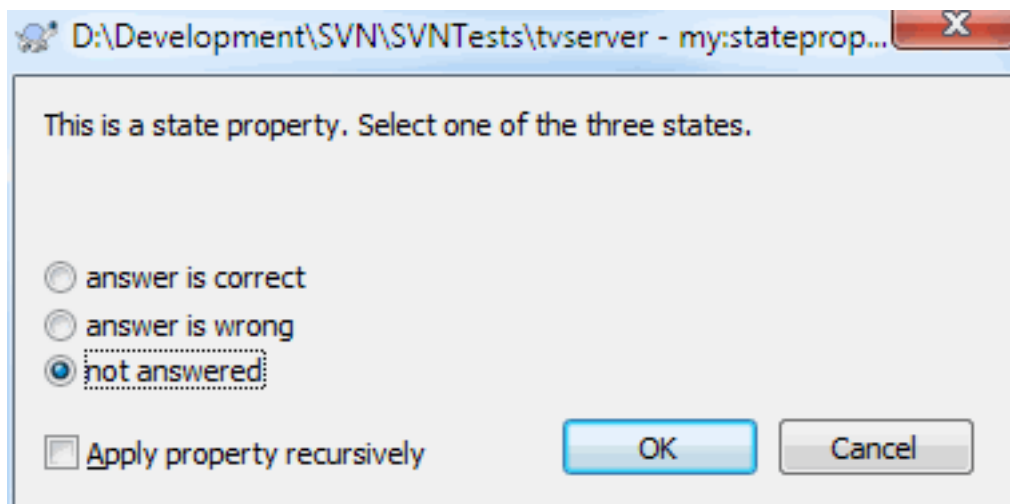
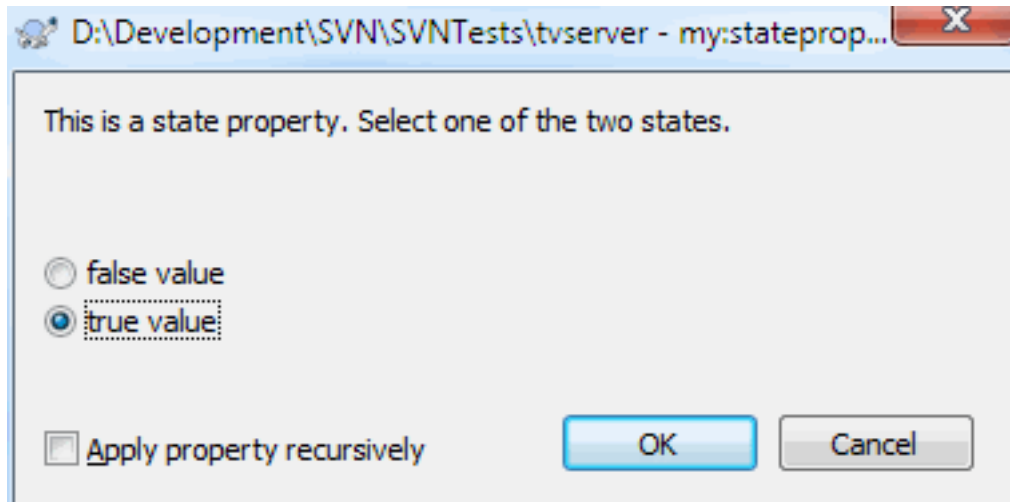
Specify your property like this:

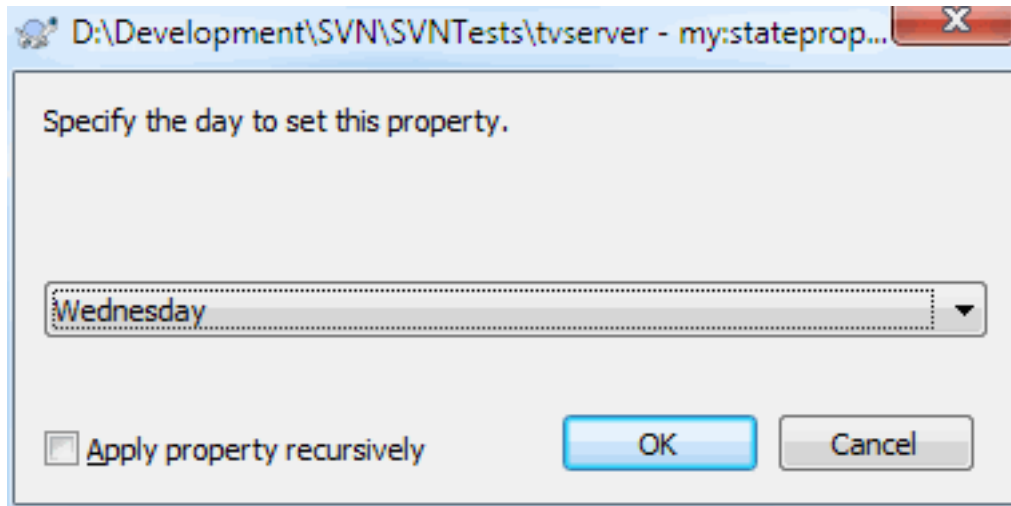
```
propertyname=bool;labeltext(YESVALUE;NOVALUE;Checkboxtext)
```

the `LabelText` is the text shown in the dialog above the checkbox where you can explain the purpose and use of the property. The other parameters should be self explanatory.

state

If your property represents one of many possible states, e.g., `yes`, `no`, `maybe`, then you can configure your property as a `state`





**Afbeelding 4.40. Property dialog state user types**

property like this:

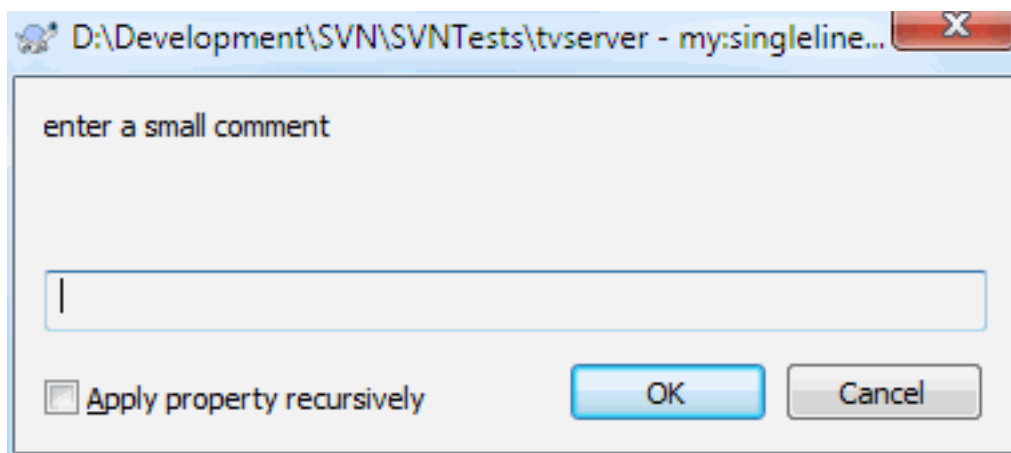
```
propertyname=state;labeltext(DEFVAL;VAL1;TEXT1;VAL2;TEXT2;VAL3;TEXT3;...)
```

The parameters are the same as for the `bool` property, with `DEFVAL` being the default value to be used if the property isn't set yet or has a value that's not configured.

For up to three different values, the dialog shows up to three radio buttons. If there are more values configured, it uses a combo box from where the user can select the required state.

`singleline`

For properties that consist of one line of text, use the `singleline` property type:



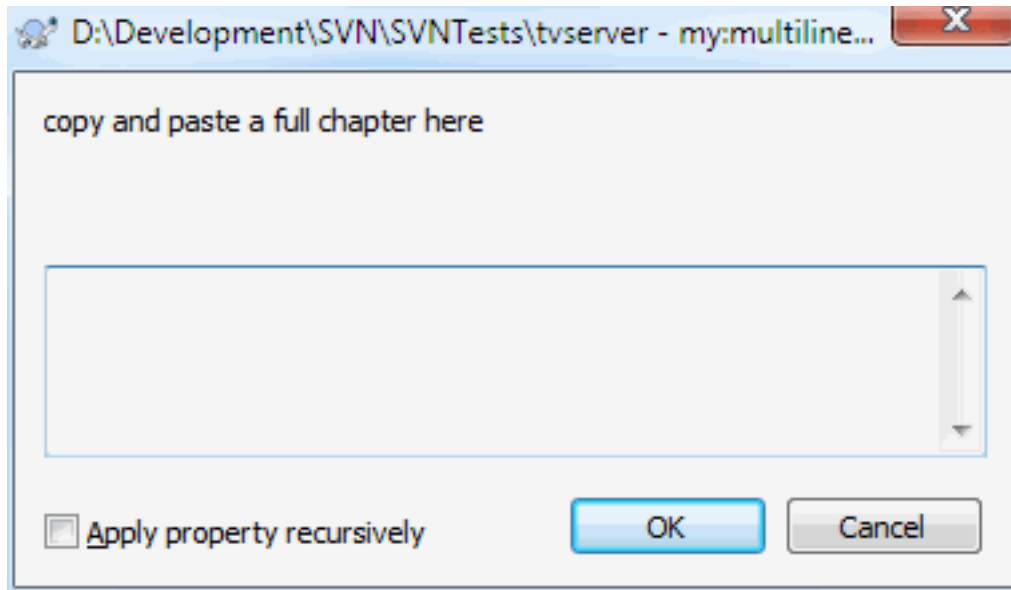
**Afbeelding 4.41. Property dialog single-line user types**

```
propertyname=singleline;labeltext(regex)
```

the `regex` specifies a regular expression which is used to validate (match) the text the user entered. If the text does not match the regex, then the user is shown an error and the property isn't set.

`multiline`

For properties that consist of multiple lines of text, use the `multiline` property type:



**Afbeelding 4.42. Property dialog multi-line user types**

```
propertyname=multiline;labeltext(regex)
```

the `regex` specifies a regular expression which is used to validate (match) the text the user entered. Don't forget to include the newline (`\n`) character in the regex!

The screenshots above were made with the following `tsvn:userdirproperties`:

```
my:boolprop=bool;This is a bool type property. Either check or uncheck it.(true/false)
my:stateprop1=state;This is a state property. Select one of the two states.(true/true)
my:stateprop2=state;This is a state property. Select one of the three states.(maybe/tr
my:stateprop3=state;Specify the day to set this property.(1;1;Monday;2;Tuesday;3;Wedne
my:singlelineprop=singleline;enter a small comment(.*)
my:multilineprop=multiline;copy and paste a full chapter here(.*)
```

TortoiseSVN kan worden geïntegreerd met enkele fout zoek tools. Hiervoor worden de project-eigenschappen gebruikt welke beginnen met `bugtraq:`. Lees [Paragraaf 4.29, "Integratie met Bug Tracking Systemen / Issue Trackers"](#) voor meer informatie.

Het kan ook worden geïntegreerd met enkele web-gebaseerde archief verkenners door gebruik te maken van de project-eigenschappen welke beginnen met `webviewer:`. Lees [Paragraaf 4.30, "Integratie met Web-gebaseerde Archief Viewers"](#) voor meer informatie.



### Stel de project eigenschappen in voor folders

These special project properties must be set on *folders* for the system to work. When you use a TortoiseSVN command which uses these properties, the properties are read from the folder you clicked on. If the properties are not found there, TortoiseSVN will search upwards through the folder tree to find them until it comes to an unversioned folder, or the tree root (e.g. `C:\`) is found. If you can be sure that each user checks out only from e.g. `trunk/` and not some sub-folder, then it is sufficient to set the properties on `trunk/`. If you can't be sure, you should set the properties recursively on each sub-folder. If you set the same property but you use different values at different depths in your project hierarchy then you will get different results depending on where you click in the folder structure.



Bij eigenschappen welke *alleen* op projecten van toepassing zijn, bijv. `tsvn:`, `bugtraq:` en `webviewer:`, kunt u gebruik maken van de **Recursief vinkbox** om de eigenschap op alle sub-mappen binnen de hiërarchie in te stellen, zonder het ook op de bestanden in te stellen.

Als u met TortoiseSVN nieuwe sub-mappen aan een werkkopie toevoegt, zullen alle project eigenschappen welke in de bovenliggende map aanwezig zijn, automatisch aan de nieuwe sub-map worden toegevoegd.



## Beperkingen Van De Archief-verkenner

Het op afstand ophalen van eigenschappen gaat erg langzaam. Enkele functies, zoals boven beschreven, zullen vanuit de archief-verkenner niet zo werken als in een werkkopie.

- Als je een eigenschap toevoegt met de archief-verkenner, dan worden alleen de standaard `svn:` eigenschappen aangeboden in een voorgedefinieerde lijst. Elke andere naam van een eigenschap moet handmatig ingevoerd worden.
- Eigenschappen kunnen niet recursief ingesteld of verwijderd worden met de archief-verkenner.
- Project eigenschappen zullen *niet* automatisch propageren als er een map dieper in een mapstructuur is toegevoegd met de archief-verkenner.
- `tsvn:autoprops` zal *geen* eigenschappen voor bestanden instellen als deze bestanden met de archief-verkenner worden toegevoegd.



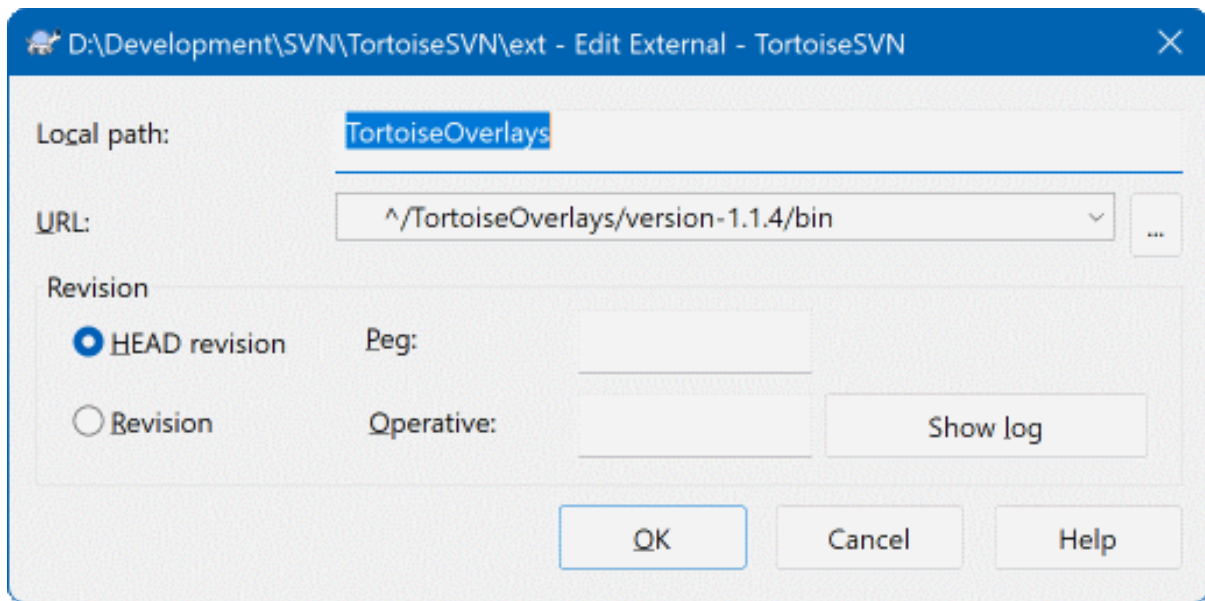
## Let op

Ondanks dat TortoiseSVN's project eigenschappen erg nuttig zijn, werken ze alleen met TortoiseSVN en sommige nieuwe eigenschappen zullen allen samenwerken met nieuwe versies van TortoiseSVN. Als er mensen met verschillende Subversion clients werken aan een project, of misschien een oude versie van TortoiseSVN gebruiken, dan wil je misschien archief hooks gebruiken om een project beleid te forceren. Project eigenschappen kunnen alleen helpen om een beleid te implementeren, eigenschappen kunnen een beleid niet afdwingen.

### 4.18.3. Eigenschappen bewerken

Some properties have to use specific values, or be formatted in a specific way in order to be used for automation. To help get the formatting correct, TortoiseSVN presents edit dialogs for some particular properties which show the possible values or break the property into its individual components.

### 4.18.3.1. Externe inhoud



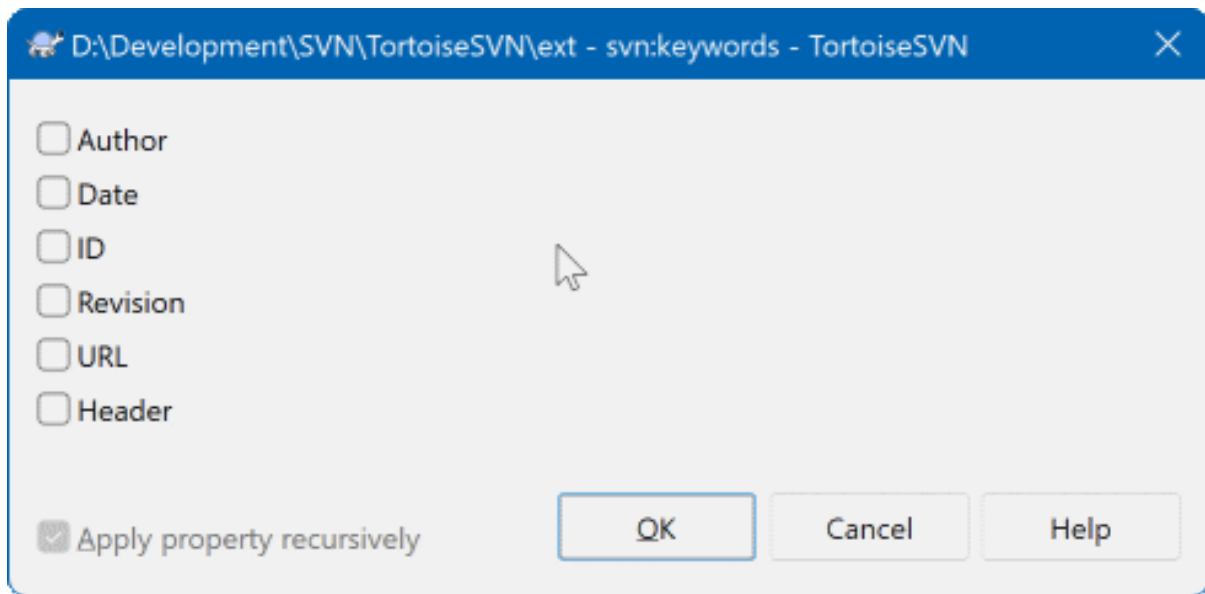
**Afbeelding 4.43. svn:externals eigenschappen pagina**

The `svn:externals` property can be used to pull in other projects from the same repository or a completely different repository as described in [Paragraaf 4.19, “Externe Objecten”](#).

You need to define the name of the sub-folder that the external folder is checked out as, and the Subversion URL of the external item. You can check out an external at its HEAD revision, so when the external item changes in the repository, your working copy will receive those changes on update. However, if you want the external to reference a particular stable point then you can specify the specific revision to use. IN this case you may also want to specify the same revision as a peg revision. If the external item is renamed at some point in the future then Subversion will not be able to update this item in your working copy. By specifying a peg revision you tell Subversion to look for an item that had that name at the peg revision rather than at HEAD.

The button Find HEAD-Revision fetches the HEAD revision of every external URL and shows that HEAD revision in the rightmost column. After the HEAD revision is known, a simple right click on an external gives you the command to peg the selected externals to their explicit HEAD revision. In case the HEAD revision is not known yet, the right click command will fetch the HEAD revision first.

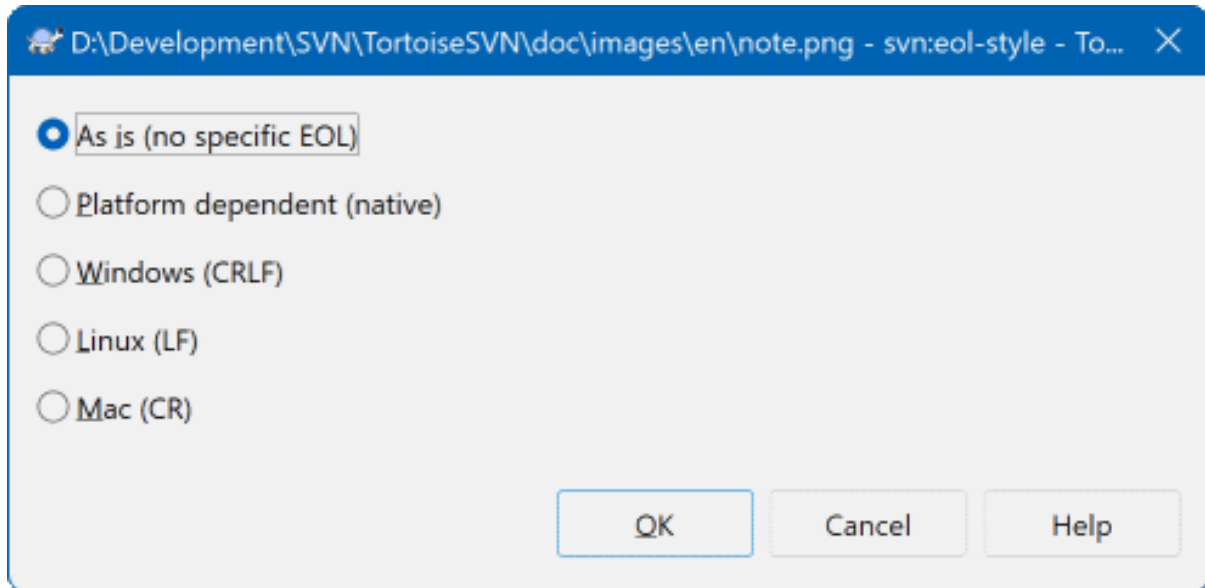
### 4.18.3.2. SVN Sleutelwoorden



**Afbeelding 4.44. svn:keywords eigenschappen pagina**

Selecteer de sleutelwoorden die je wenst uit te breiden in jouw bestand.

### 4.18.3.3. Regeleinde-stijl

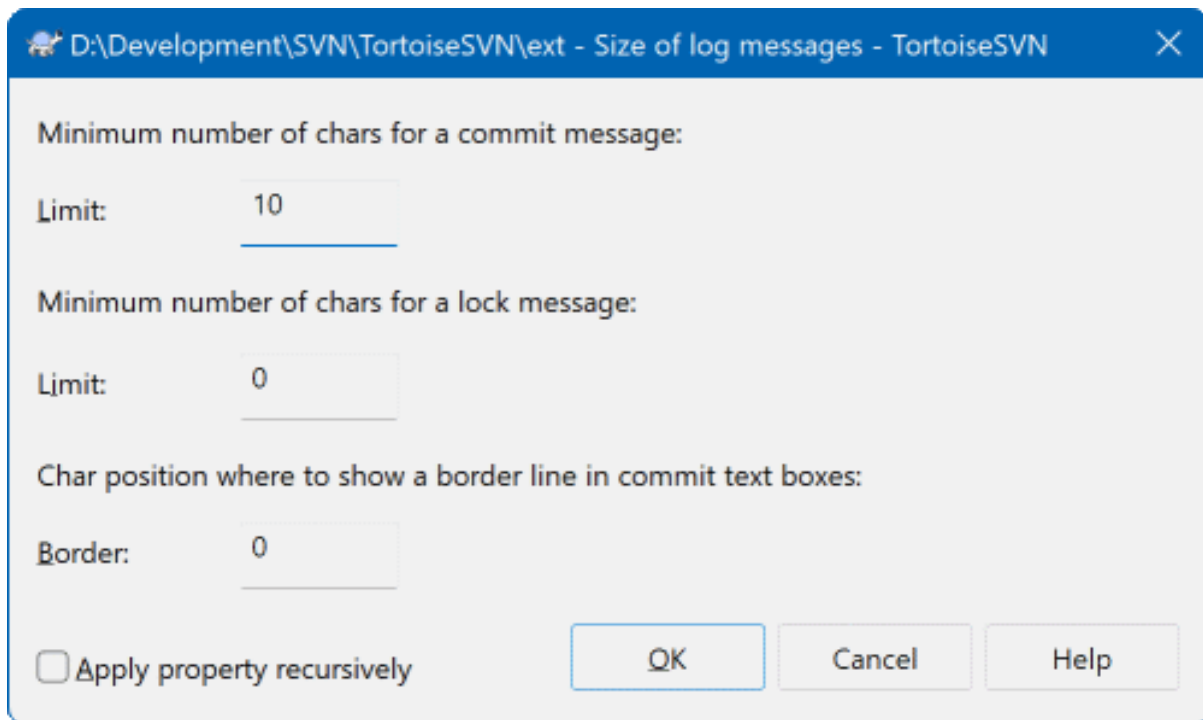


**Afbeelding 4.45. svn:eol-style eigenschappen pagina**

Select the end-of-line style that you wish to use and TortoiseSVN will use the correct property value.



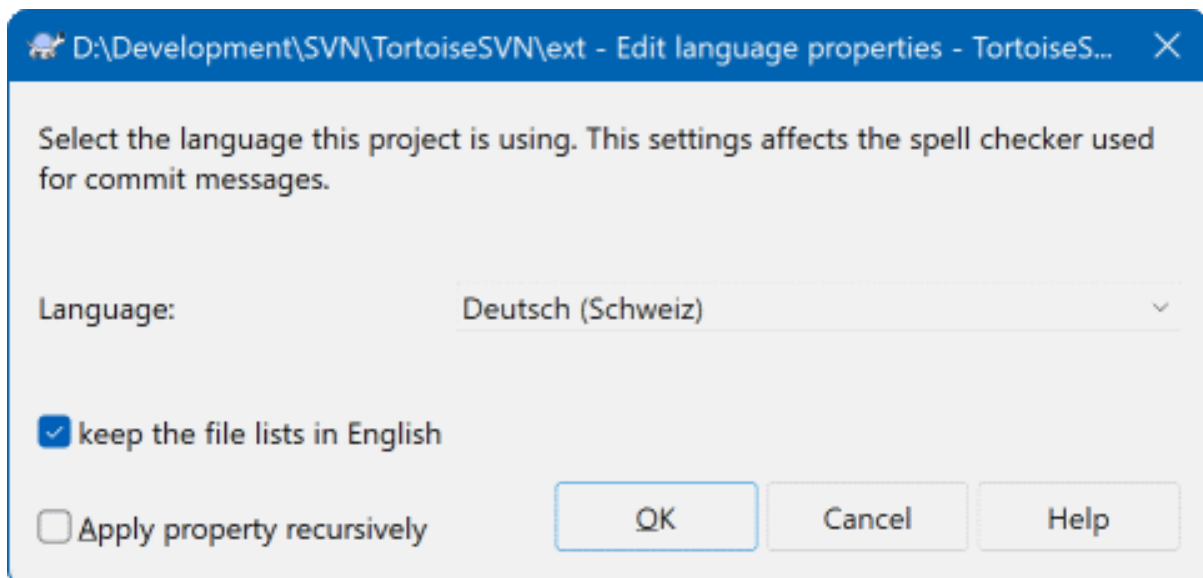
#### 4.18.3.5. Logberichtgrootte



**Afbeelding 4.47. Logberichtgrootte eigenschappen pagina**

These 3 properties control the formatting of log messages. The first 2 disable the OK in the commit or lock dialogs until the message meets the minimum length. The border position shows a marker at the given column width as a guide for projects which have width limits on their log messages. Setting a value to zero will delete the property.

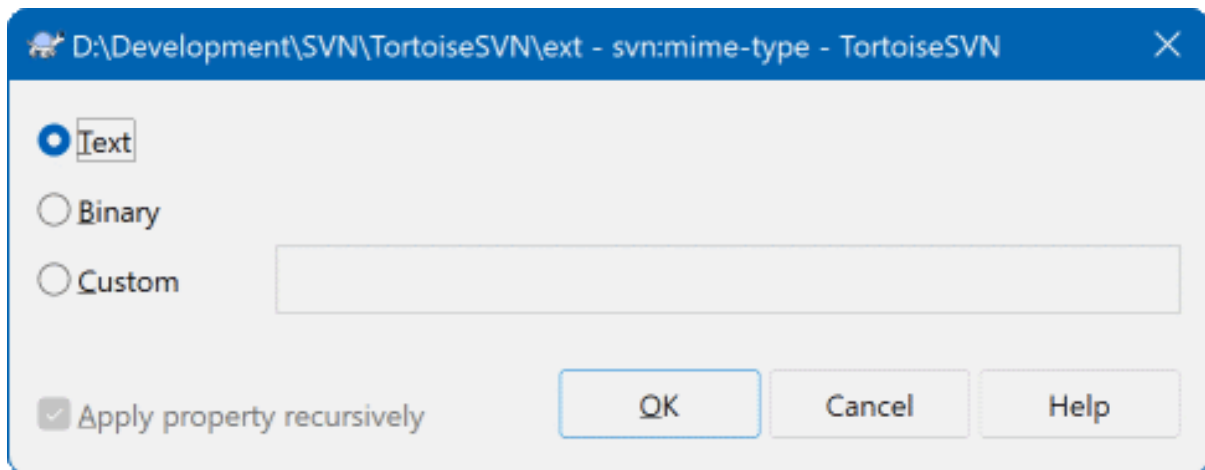
#### 4.18.3.6. Projecttaal



**Afbeelding 4.48. Taal eigenschappen pagina**

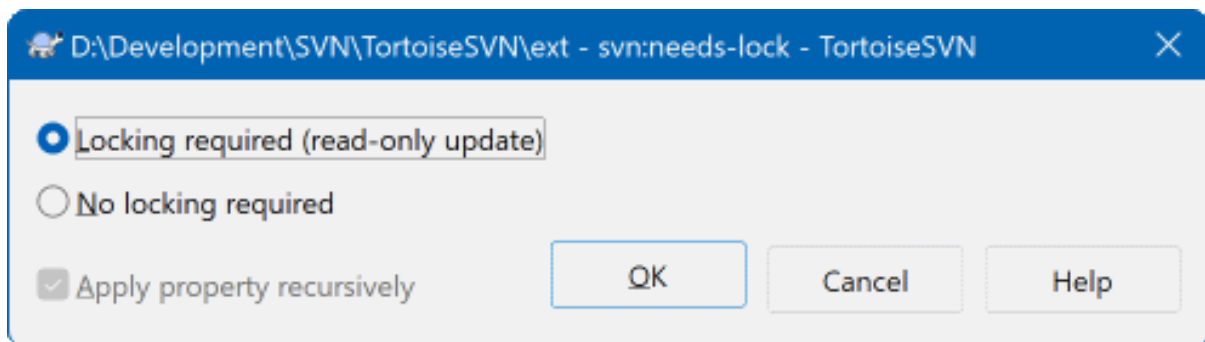
Choose the language to use for spell-checking log messages in the commit dialog. The file lists checkbox comes into effect when you right click in the log message pane and select **Paste file list**. By default the Subversion status will be shown in your local language. When this box is checked the status is always given in English, for projects which require English-only log messages.

#### 4.18.3.7. MIME-type



Afbeelding 4.49. svn:mime-type eigenschappen pagina

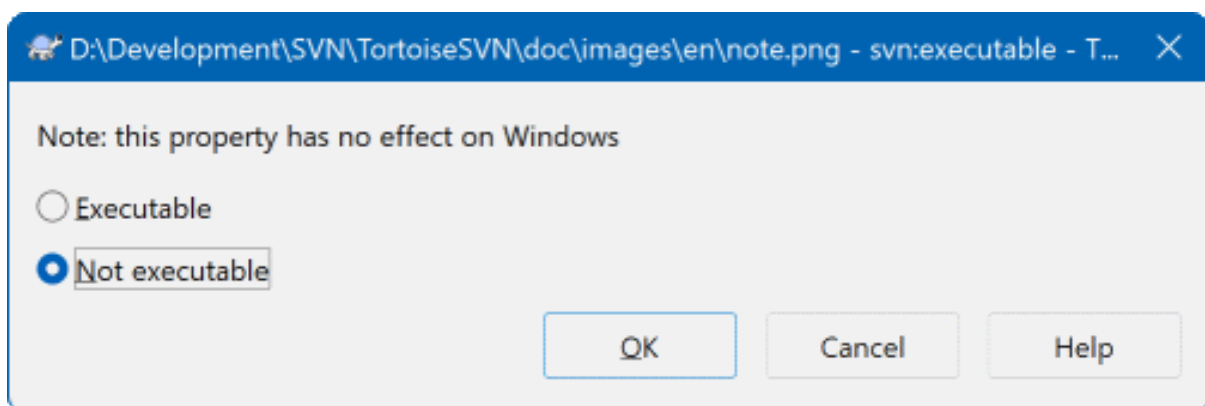
#### 4.18.3.8. svn:needs-lock



Afbeelding 4.50. svn:needs-lock eigenschappen pagina

This property simply controls whether a file will be checked out as read-only if there is no lock held for it in the working copy.

#### 4.18.3.9. svn:executable



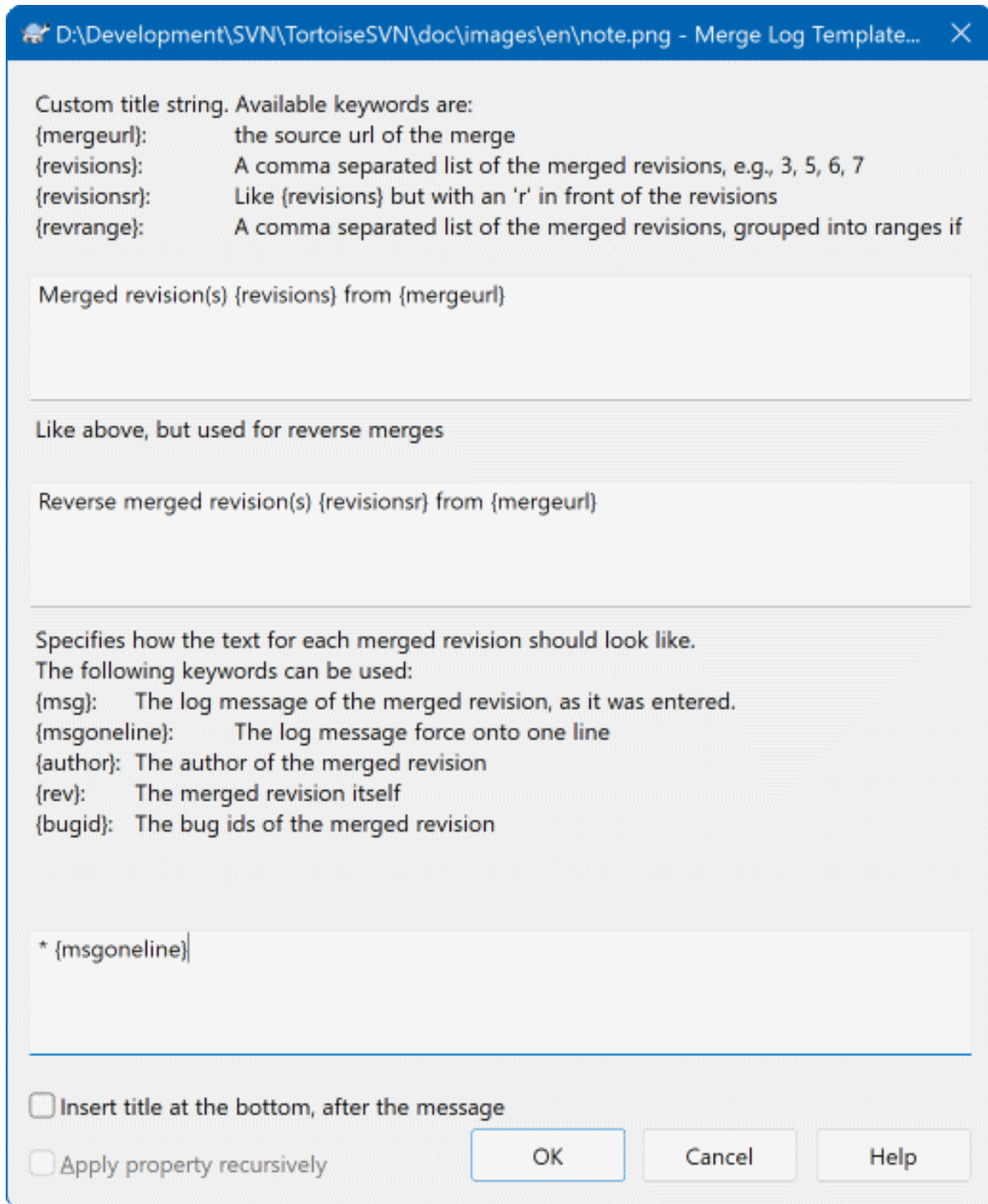
Afbeelding 4.51. svn:executable eigenschappen pagina

This property controls whether a file will be given executable status when checked out on a Unix/Linux system. It has no effect on a Windows checkout.

#### 4.18.3.10. Merge log message templates

Whenever revisions are merged into a working copy, TortoiseSVN generates a log message from all the merged revisions. Those are then available from the **Recent Messages** button in the commit dialog.

You can customize that generated message with the following properties:



**Afbeelding 4.52. Property dialog merge log message templates**

tsvn:mergelogtemplatetitle, tsvn:mergelogtemplatereversetitle

This property specifies the first part of the generated log message. The following keywords can be used:

{revisions}

A comma separated list of the merged revisions, e.g., 3, 5, 6, 7

{revisionsr}

Like {revisions}, but with each revision preceded with an r, e.g., r3, r5, r6, r7

{revrange}

A comma separated list of the merged revisions, grouped into ranges if possible, e.g., 3, 5-7

{mergeurl}

The source URL of the merge, i.e., where the revisions are merged from.

The default value for this string is `Merged revision(s) {revrange} from {mergeurl}`: with a newline at the end.

tsvn:mergelogtemplatemsg

This property specifies how the text for each merged revision should look like. The following keywords can be used:

{msg}

The log message of the merged revision, as it was entered.

{msgoneline}

Like {msg}, but all newlines are replaced with a space, so that the whole log message appears on one single line.

{author}

The author of the merged revision.

{rev}

The merged revision itself.

{bugids}

The bug IDs of the merged revision, if there are any.

tsvn:mergelogtemplatemsgtitlebottom

This property specifies the position of the title string specified with the `tsvn:mergelogtemplatetitle` or `tsvn:mergelogtemplateresertitle`. If the property is set to `yes` or `true`, then the title string is appended at the bottom instead of the top.



### Belangrijk

This only works if the merged revisions are already in the log cache. If you have disabled the log cache or not shown the log first before the merge, the generated message won't contain any information about the merged revisions.

## 4.19. Externe Objecten

Soms is het handig om een werkkopie te hebben, die bestaat uit bestanden die in verschillende aparte archieven zijn opgeslagen. Bijvoorbeeld, je wilt dat bestanden en submappen van verschillende locaties uit het archief worden gehaald, of misschien zelfs vanuit verschillende archieven. Als je wilt dat elke gebruiker dezelfde indeling gebruikt, dan kun je gebruik maken van `svn:externals` eigenschappen om de benodigde bronnen naar de locaties te halen waar ze nodig zijn.

### 4.19.1. Externe Mappen

Let's say you check out a working copy of `/project1` to `D:\dev\project1`. Select the folder `D:\dev\project1`, right click and choose `Windows Menu → Properties` from the context menu. The Properties Dialog comes up. Then go to the Subversion tab. There, you can set properties. Click `Properties...` In the



properties dialog, either double click on the `svn:externals` if it already exists, or click on the **New...** button and select `externals` from the menu. To add a new external, click the **New...** and then fill in the required information in the shown dialog.



## Let op

URLs must be properly escaped or they will not work, e.g. you must replace each space with `%20`.

Als je in lokale paden spaties en andere speciale tekens wilt gebruiken, dan moet je deze tussen aanhalingstekens plaatsen, of je kunt de `\` (schuine streep naar links, ook wel backslash genoemd) gebruiken, zoals deze gebruikt worden Unix voor speciale karakters. Dit houdt ook in dat je de `/` (schuine streep naar rechts, ook wel forward slash genoemd) moet gebruiken als scheidingsteken voor paden. Merk op dat dit gedrag nieuw is voor Subversion 1.6 en dat dit niet werkt bij oudere versies.



## Gebruik Explicite Revisie Nummers

You should strongly consider using explicit revision numbers in all of your externals definitions, as described above. Doing so means that you get to decide when to pull down a different snapshot of external information, and exactly which snapshot to pull. Besides the common sense aspect of not being surprised by changes to third-party repositories that you might not have any control over, using explicit revision numbers also means that as you backdate your working copy to a previous revision, your externals definitions will also revert to the way they looked in that previous revision, which in turn means that the external working copies will be updated to match the way *they* looked back when your repository was at that previous revision. For software projects, this could be the difference between a successful and a failed build of an older snapshot of your complex code base.

The edit dialog for `svn:externals` properties allows you to select the externals and automatically set them explicitly to the `HEAD` revision.

Als het externe project zich in hetzelfde archief bevindt, dan worden de wijzigingen die je daarin aanbrengt meegenomen in de vastleglijst als je het hoofdproject vastlegt.

If the external project is in a different repository, any changes you make to the external project will be shown or indicated when you commit the main project, but you have to commit those external changes separately.

Als je absolute URL's gebruikt in de `svn:externals` definities en je moet een keer je werkkopie verplaatsen (oftewel als de URL van je archief verandert), dan zullen de externe verwijzingen niet mee veranderen en dus zullen deze niet meer werken.

Om zulke problemen te voorkomen, ondersteunen de Subversion clients vanaf versie 1.5 relatieve externe URL's. Er zijn vier verschillende methodes voor het definiëren van relatieve URL's. Bij de volgende voorbeelden gaan we er van uit dat we tweearchieven hebben: één op `http://example.com/svn/repos-1` en de ander op `http://example.com/svn/repos-2`. We hebben een werkkopie opgehaald van `http://example.com/svn/repos-1/project/trunk` naar de locatie `C:\Working` en de `svn:externals` eigenschap is ingesteld op de basislijn.

Relatief ten opzichte van bovenliggende map

These URLs always begin with the string `../` for example:

```
../../widgets/foo common/foo-widget
```

This will extract `http://example.com/svn/repos-1/widgets/foo` into `C:\Working\common\foo-widget`.

Merk op dat de URL relatief is ten opzicht van de URL van de map met de `svn:externals` eigenschap, niet ten opzichte van de map waar de externe opgeslagen is op de schijf.

#### Relatief ten opzichte van de archief hoofdmap

These URLs always begin with the string `^/` for example:

```
^/widgets/foo common/foo-widget
```

This will extract `http://example.com/svn/repos-1/widgets/foo` into `C:\Working\common\foo-widget`.

You can easily refer to other repositories with the same `SVNParentPath` (a common directory holding several repositories). For example:

```
^/../repos-2/hammers/claw common/claw-hammer
```

This will extract `http://example.com/svn/repos-2/hammers/claw` into `C:\Working\common\claw-hammer`.

#### Relatief ten opzichte van de indeling

URLs beginning with the string `//` copy only the scheme part of the URL. This is useful when the same hostname must be accessed with different schemes depending upon network location; e.g. clients in the intranet use `http://` while external clients use `svn+ssh://`. For example:

```
//example.com/svn/repos-1/widgets/foo common/foo-widget
```

This will extract `http://example.com/svn/repos-1/widgets/foo` or `svn+ssh://example.com/svn/repos-1/widgets/foo` depending on which method was used to checkout `C:\Working`.

#### Relatief ten opzichte van de server hostnaam

URLs beginning with the string `/` copy the scheme and the hostname part of the URL, for example:

```
/svn/repos-1/widgets/foo common/foo-widget
```

This will extract `http://example.com/svn/repos-1/widgets/foo` into `C:\Working\common\foo-widget`. But if you checkout your working copy from another server at `svn+ssh://another.mirror.net/svn/repos-1/project1/trunk` then the external reference will extract `svn+ssh://another.mirror.net/svn/repos-1/widgets/foo`.

You can also specify a `peg` and operative revision for the URL if required. To learn more about `peg` and operative revisions, please read the [corresponding chapter](http://svnbook.red-bean.com/en/1.8/svn.advanced.pegrevs.html) [http://svnbook.red-bean.com/en/1.8/svn.advanced.pegrevs.html] in the Subversion book.



### Belangrijk

If you specify the target folder for the external as a subfolder like in the examples above, make sure that *all* folders in between are versioned as well. So for the examples above, the folder `common` should be versioned!

While the external will work in most situations properly if folders in between are not versioned, there are some operations that won't work as you expect. And the status overlay icons in explorer will also not show the correct status.

Als je meer informatie wilt hebben over hoe TortoiseSVN omgaat met Eigenschappen, lees dan [Paragraaf 4.18, "Project Instellingen"](#).

Lees voor meer informatie over de verschillende methodes voor het benaderen van gemeenschappelijke sub-projecten [Paragraaf B.6, "Een veelgebruikt subproject gebruiken"](#).

### 4.19.2. Externe Bestanden

Vanaf Subversion 1.6 kun je enkele externe bestanden aan je werkkopie toevoegen, door gebruik te maken van dezelfde syntax als voor mappen. Er zijn wel enkele beperkingen.

- The path to the file external must be a direct child of the folder where you set the `svn:externals` property.
- De URL van een extern bestand moet in hetzelfde archief zijn als de URL van het archief waarin het externe bestand wordt opgehaald, externe bestanden tussen verschillende archieven worden niet ondersteund.

Een extern bestand gedraagt zich net zoals elk ander bestand met versiebeheer, maar deze kunnen niet verplaatst of verwijderd worden met normale commando's; de `svn:externals` eigenschap moet in de plaats daarvan gewijzigd worden.

### 4.19.3. Creating externals via drag and drop

If you already have a working copy of the files or folders you want to include as externals in another working copy, you can simply add those via drag and drop from the windows explorer.

Simply right drag the file or folder from one working copy to where you want those to be included as externals. A context menu appears when you release the mouse button: **SVN Add as externals here** if you click on that context menu entry, the `svn:externals` property is automatically added. All you have to do after that is commit the property changes and update to get those externals properly included in your working copy.

## 4.20. Vertakken / labelen

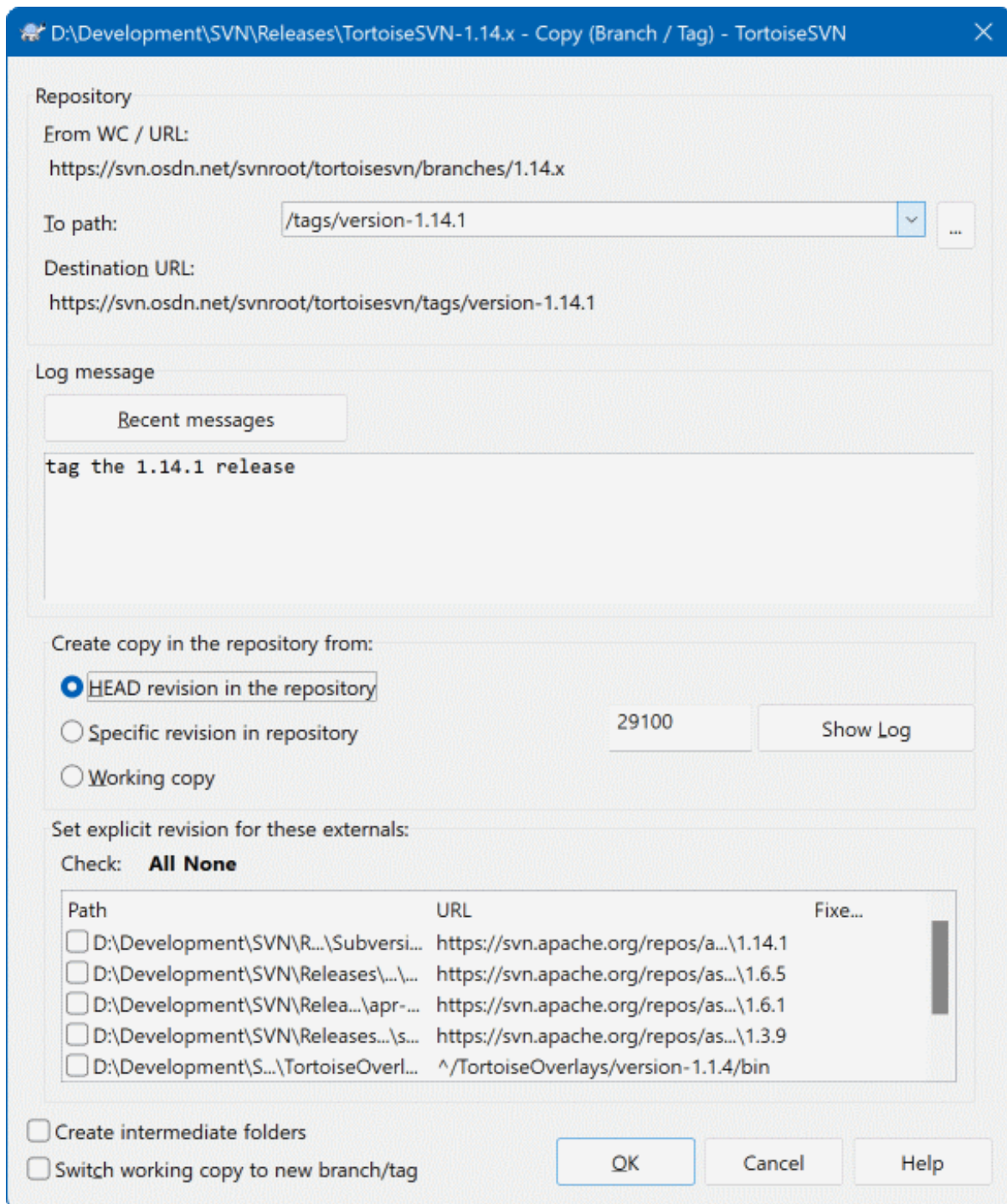
Eén van de mogelijkheden van versiebeheersystemen is het isoleren van wijzigingen in een aparte ontwikkellijn. Zo'n lijn wordt *tak* (in het Engels *branch*) genoemd. Takken worden vaak gebruikt om nieuwe functies uit te proberen, zonder dat daarbij de hoofdontwikkeling verstoord wordt met compiler fouten en bugs. Zodra de nieuwe functie stabiel genoeg is, wordt deze ontwikkel tak *samengevoegd* met de basislijn (in het Engels de *trunk* genoemd).

Een andere mogelijkheid van versiebeheersystemen is het markeren van bepaalde revisies (bijv. een vrijgegeven versie), zodat op elk gewenst moment een bepaald pakket of een omgeving opnieuw (op) kunt bouwen. Dit proces wordt *labelen* genoemd (in het Engels *tagging*).

Subversion heeft geen speciale commando's voor vertakken of labelen, maar gebruikt hiervoor zogenaamde "goedkope kopieën". Goedkope kopieën zijn vergelijkbaar met de harde links in Unix, wat inhoudt dat er in plaats van een gehele kopie te maken van het archief, er een interne link wordt aangemaakt. Deze link verwijst naar een specifieke boom of revisie. Hierdoor kunnen takken en labels heel snel gemaakt worden en hebben ze bijna geen extra ruimte nodig in het archief.

### 4.20.1. Aanmaken van Tak of Label

Als je je project met de aanbevolen mapstructuur hebt geïmporteerd, dan is het aanmaken van een tak of label erg eenvoudig:



### Afbeelding 4.53. De Tak/Label Dialog

Selecteer de map in je werkkopie, waarvan je een tak of label wilt maken. Selecteer vervolgens het commando TortoiseSVN → Maak Tak/label....

The default destination URL for the new branch will be the source URL on which your working copy is based. You will need to edit that URL to the new path for your branch/tag. So instead of

```
http://svn.collab.net/repos/ProjectName/trunk
```

you might now use something like

```
http://svn.collab.net/repos/ProjectName/tags/Release_1.10
```

If you can't remember the naming convention you used last time, click the button on the right to open the repository browser so you can view the existing repository structure.



## intermediate folders

When you specify the target URL, all the folders up to the last one must already exist or you will get an error message. In the above example, the URL `http://svn.collab.net/repos/ProjectName/tags/` must exist to create the `Release_1.10` tag.

However if you want to create a branch/tag to an URL that has intermediate folders that don't exist yet you can check the option `Create intermediate folders` at the bottom of the dialog. If that option is activated, all intermediate folders are automatically created.

Note that this option is disabled by default to avoid typos. For example, if you typed the target URL as `http://svn.collab.net/repos/ProjectName/Tags/Release_1.10` instead of `http://svn.collab.net/repos/ProjectName/tags/Release_1.10`, you would get an error with the option disabled, but with the option enabled a folder `Tags` would be automatically created, and you would end up with a folder `Tags` and a folder `tags`.

Nu moet je de bron van de kopie selecteren. Er zijn daarvoor drie mogelijkheden:

### HEAD revisie in het archief

De nieuwe tak wordt direct binnen het archief gekopieerd van de HEAD revisie in het archief. Er hoeven geen gegevens verstuurd te worden vanuit je werkkopie. De tak is dan erg snel gemaakt.

### Specifieke revisie in het archief

De nieuwe tak wordt direct binnen het archief gekopieerd, maar je kunt een oudere revisie selecteren. Dit is handig als je vergeten bent een label te maken toen je een nieuwe versie vrijgaf vorige week. Als je het revisienummer niet meer weet, klik dan op de knop aan de rechterkant om het revisie logboek te bekijken en te selecteren. Er worden ook hierbij geen gegevens vanuit je werkkopie gehaald en dus is de tak of label erg snel gemaakt.

### Werkkopie

De nieuwe tak moet een identieke kopie zijn van je lokale werkkopie. Als je bijvoorbeeld van enkele bestanden een oudere versie naar je werkkopie hebt gehaald, of als je lokale wijzigingen hebt aangebracht, dan worden exact die bestanden die je in je werkkopie hebt in de tak of label meegenomen. Uiteraard heeft een label van deze complexiteit mogelijk het gevolg dat er informatie vanuit je werkkopie naar het archief gestuurd moet worden, mocht de informatie daar nog niet bekend zijn.

Als je wilt dat je werkkopie automatisch overschakelt naar de nieuw aangemaakte tak, gebruik schakel dan de optie `Schakel werkkopie naar de nieuwe tak/label` in. Voordat je dat doet, moet je er zeker van zijn dat er geen wijzigingen meer in je werkkopie zitten. Als er wijzigingen in zitten, dan worden deze ook meteen samengevoegd in de werkkopie van de tak als je omschakelt.

If your working copy has other projects included with `svn:externals` properties, those externals will be listed at the bottom of the branch/tag dialog. For each external, the target path and the source URL is shown.

If you want to make sure that the new tag always is in a consistent state, check all the externals to have their revisions pinned. If you don't check the externals and those externals point to a HEAD revision which might change in the future, checking out the new tag will check out that HEAD revision of the external and your tag might not compile anymore. So it's always a good idea to set the externals to an explicit revision when creating a tag.

The externals are automatically pinned to either the current HEAD revision or the working copy BASE revision, depending on the source of the branch/tag:



Copy Source	Pinned Revision
HEAD revisie in het archief	external's repos HEAD revision
Specifieke revisie in het archief	external's repos HEAD revision
Werkkopie	external's WC BASE revision

**Tabel 4.1. Pinned Revision**

### externals within externals

If a project that is included as an external has itself included externals, then those will not be tagged!  
Only externals that are direct children can be tagged.

Klik op OK om de nieuwe kopie vast te leggen in het archief. Vergeet niet een logboek boodschap achter te laten. Merk op dat de kopie gemaakt wordt *in het archief*.

Merk op dat, tenzij je ervoor gekozen hebt je werkkopie om te laten schakelen naar de nieuw aangemaakte tak, het aanmaken van een Tak of Label je werkkopie *niet* verandert. Ook als je een tak aanmaakt van je werkkopie, dan worden de wijzigingen in de tak vastgelegd, niet in de basislijn. Het is dus mogelijk dat je werkkopie nog steeds als gewijzigd wordt aangemerkt ten opzicht van de basislijn.

#### 4.20.2. Andere maniere om een tak of label aan te maken

Je kunt ook een tak of label maken zonder dat je een werkkopie hebt. Open hiervoor de archief-verkenner. Daar kun je mappen slepen naar een nieuwe locatie. Je moet de **Ctrl** ingedrukt houden tijdens het slepen om een kopie te maken, anders wordt de map verplaatst in plaats van gekopieerd.

Je kunt ook een map verslepen met de rechter muisknop. Zodra je de muisknop loslaat, kun je kiezen of je de map kopiëren of verplaatsen wilt. Uiteraard moet je een map kopiëren als je een tak of label wilt aanmaken, niet verplaatsen.

Yet another way is from the log dialog. You can show the log dialog for e.g. trunk, select a revision (either the HEAD revision at the very top or an earlier revision), right click and choose **create branch/tag from revision...**

#### 4.20.3. Ophalen of Wisselen...

.. dat is (niet echt) de vraag. Terwijl bij het ophalen alles van de gewenste tak uit het archief naar je werkkopie wordt gehaald, haalt TortoiseSVN → Wissel... alleen de gewijzigde gegevens op naar je werkkopie. Dat is gunstig voor de belasting van het netwerk en goed voor je geduld. :-)

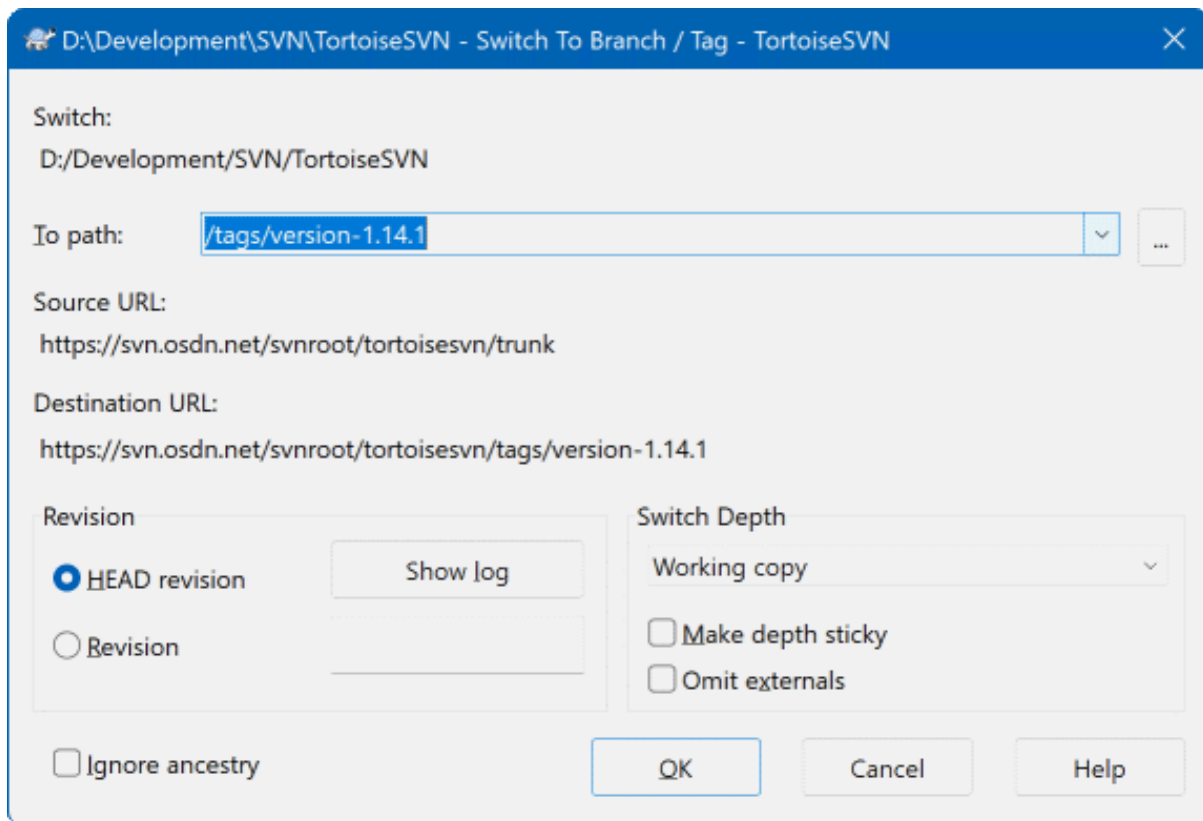
Er zijn verschillende mogelijkheden om met je nieuw aangemaakte tak of label te gaan werken. Je kunt:

- TortoiseSVN → Ophalen gebruiken om een versie kopie te maken in een lege map. Je kunt naar elke gewenste locatie op je harde schijf je gegevens ophalen en je kunt zoveel werkkopieën maken als je wilt.
- Je huidige werkkopie wisselen/schakelen naar de nieuw in het archief aangemaakte kopie. Selecteer weer de hoogste map van je project en kies TortoiseSVN → Wissel... vanuit het context menu.

Voer in het volgende venster de URL in van de tak die je net aangemaakt hebt. Selecteer het HEAD Revisie keuzerondje en klik op OK. Je werkkopie wordt dan gewisseld naar de nieuwe tak/label.

Wisselen werkt net zoals Verversen in zoverre dat het nooit de lokale wijzigingen negeert. Elke wijziging die je in je werkkopie hebt aangebracht en nog niet het vastgelegd in het archief, zal worden samengevoegd zodra je wisselt. Als je wilt voorkomen dat dit gebeurt, moet je of je wijzigingen vastleggen voordat je wisselt, of je werkkopie terugzetten naar een eerder vastgelegde revisie (meestal de HEAD revisie).

- Werken aan een basislijn en een tak, zonder dat je een verse werkkopie op moet halen. Je kunt dan met de Windows Verkenner een kopie maken van je basislijn naar een andere map en vervolgens met TortoiseSVN → Wissel... de werkkopie omschakelen naar de nieuwe tak.



**Afbeelding 4.54. De Wissel Dialoog**

Ondanks dat Subversion geen verschil maakt tussen labels en takken, verschilt de manier waarop ze gebruikt worden iets.

- Labels worden in principe gemaakt om een momentopname te maken van een project op een bepaald moment. Als gevolg hiervan, worden deze normaal gesproken niet gebruikt voor ontwikkeling. Hiervoor worden takken gebruikt. Daarom adviseren we gebruik te maken van de mapstructuur met de mappen `/trunk` `/branches` `/tags`. Het werken aan een label *is geen goed idee*, maar omdat er op je lokale bestanden geen schrijf beveiliging zit, zal niets je tegenhouden. Als je echter probeert wijzigingen vast te leggen naar een pad waarin `/tags/` voor komt, dan zal TortoiseSVN je wel waarschuwen.
- Het is soms nodig om extra wijzigingen aan te brengen in vrijgaves die je al gelabeld hebt. De juiste manier is om eerst een nieuwe tak te maken van de label en vervolgens de tak vast te leggen. Voor je wijzigingen door op deze tak en maak dan opnieuw een label aan van deze tak, bijv. `Versie_1.0.1`.
- Als je een werkkopie van een tak wijzigt en vastlegt, dan zullen de wijzigingen in de tak worden vastgelegd en *niet* in de basislijn. Alleen de wijzigingen worden opgeslagen. De rest blijft een goedkope kopie.

## 4.21. Samenvoegen

Nadat takken zijn gebruikt om meerdere ontwikkelrichtingen uit te gaan kan het op gegeven moment nodig zijn om de parallel ontwikkelde wijzigingen uit een tak weer terug samen te voegen in de basislijn of omgekeerd.

It is important to understand how branching and merging works in Subversion before you start using it, as it can become quite complex. It is highly recommended that you read the chapter [Branching and Merging](http://) [http://

[svnbook.red-bean.com/en/1.8/svn.branchmerge.html](http://svnbook.red-bean.com/en/1.8/svn.branchmerge.html)] in the Subversion book, which gives a full description and many examples of how it is used.

Een volgend belangrijk punt is dat samenvoegen *altijd* vanuit een werkkopie begint. Indien wijzigingen uit een tak *naar* een andere tak samengevoegd moeten worden, moet deze laatste op uw computer zijn afgehaald en moet van hier uit het samenvoegen worden geïnitieerd met het menu TortoiseSVN → Samenvoegen....

Er wordt over het algemeen aangeraden om samen te voegen naar een werkkopie waar alle wijzigingen zijn vastgelegd. Indien het samenvoegen niet naar verwachting zou gebeuren zullen *alle* niet vastgelegde wijzigingen anders mee ongedaan worden gemaakt bij gebruik van het **Ongedaan Maken** commando, en dus ook de wijzigingen van voor het samenvoegen.

Er zijn drie situaties waar samenvoegen telkens op een verschillende manier werkt, zoals hieronder beschreven. De eerste vraag die wordt gesteld bij het Samenvoegen is welke methode gewenst is.

#### Een revisiebereik samenvoegen

Deze methode wordt gebruikt als u één of meer wijzigingen in een tak (of de basislijn) heeft aangebracht en deze ook wilt toepassen op een andere tak.

What you are asking Subversion to do is this: “ Calculate the changes necessary to get [FROM] revision 1 of branch A [TO] revision 7 of branch A, and apply those changes to my working copy (of trunk or branch B). ”

If you leave the revision range empty, Subversion uses the merge-tracking features to calculate the correct revision range to use. This is known as a reintegrate or automatic merge.

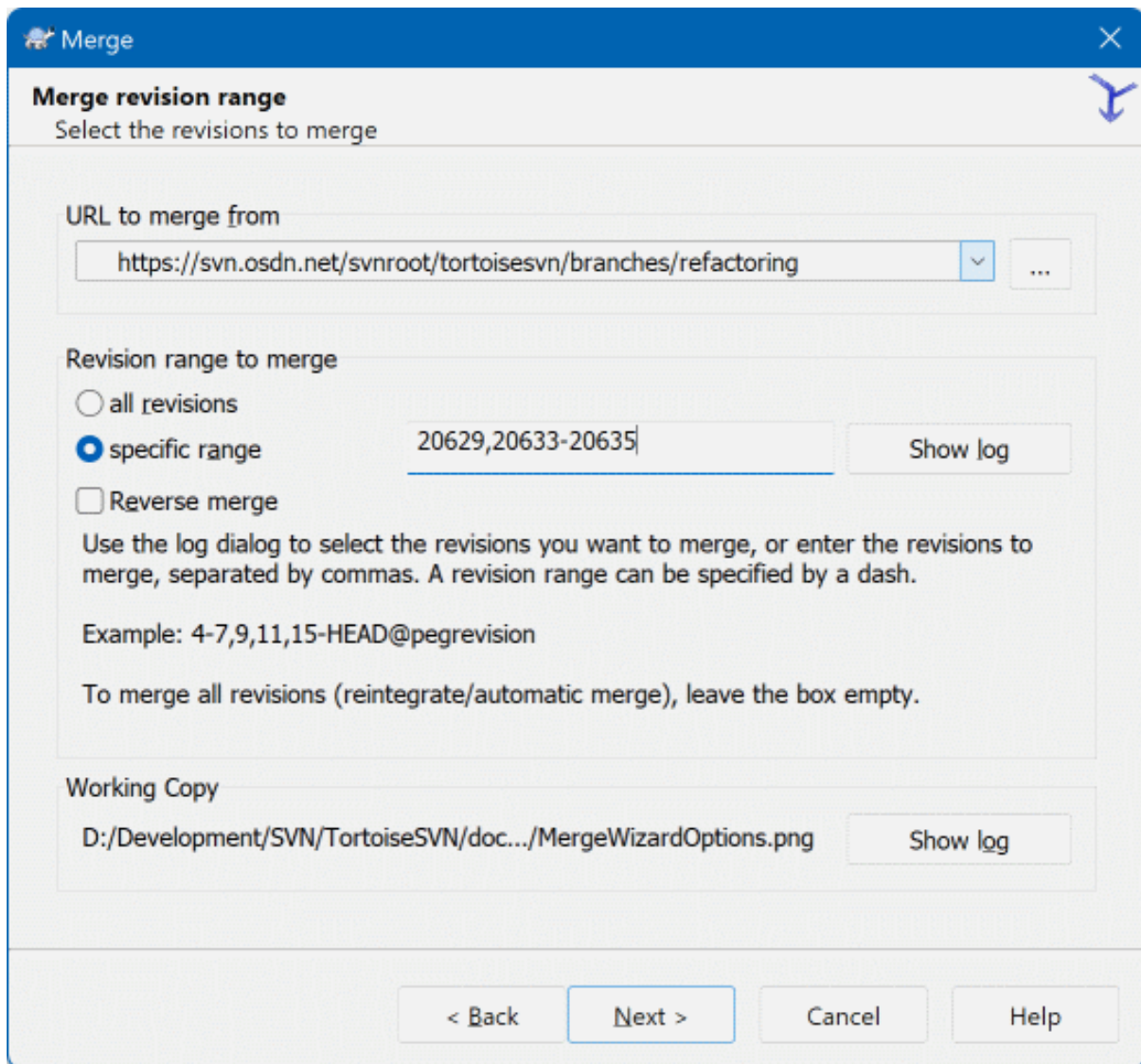
#### Twee verschillende takken samenvoegen

This is a more general case of the reintegrate method. What you are asking Subversion to do is: “ Calculate the changes necessary to get [FROM] the head revision of the trunk [TO] the head revision of the branch, and apply those changes to my working copy (of the trunk). ” The net result is that trunk now looks exactly like the branch.

If your server/repository does not support merge-tracking then this is the only way to merge a branch back to trunk. Another use case occurs when you are using vendor branches and you need to merge the changes following a new vendor drop into your trunk code. For more information read the chapter on [vendor branches](#) [<http://svnbook.red-bean.com/en/1.8/svn.advanced.vendorbr.html>] in the Subversion Book.

### 4.21.1. Een Revisie-Bereik Samenvoegen





**Afbeelding 4.55. Het Samenvoeg Hulpprogramma - Selecteer Revisie Bereik**

Geef in het Van: veld de complete folder URL van de tak of label, welke de wijzigingen bevat die u naar uw werkkopie wilt doorvoeren. U kunt ook klikken op ... om door het archief te bladeren naar de gewenste tak. Als u eerder van deze tak heeft samengevoegd, kunt u de uitklaplijst gebruiken welke een geschiedenis van eerder gebruikte URL's toont.

If you are merging from a renamed or deleted branch then you will have to go back to a revision where that branch still existed. In this case you will also need to specify that revision as a peg revision in the range of revisions being merged (see below), otherwise the merge will fail when it can't find that path at HEAD.

Geef in het Revisiebereik om samen te voegen veld de lijst aan van revisies welke u wilt samenvoegen. Dit kan een enkele revisie zijn, een lijst van specifieke revisies gescheiden door komma's, een bereik van revisies gescheiden door een min-teken, of een combinatie van deze manieren.

If you need to specify a peg revision for the merge, add the peg revision at the end of the revisions, e.g. 5-7,10@3. In the above example, the revisions 5,6,7 and 10 would be merged, with 3 being the peg revision.



## Belangrijk

Er is een belangrijk verschil in de manier waarop een revisiebereik wordt gespecificeerd in TortoiseSVN of in de commandoregel client. De gemakkelijkste manier om dit te visualiseren is om een omheining voor ogen te nemen compleet met toegangspoorten en hekwerk.

Met de commandoregel client geeft u de vast te leggen wijzigingen aan door gebruik te maken van twee “toegangspoort” revisies welke de *voor* en *na* punten aangeven.

Met TortoiseSVN geeft u de vast te leggen wijzigingsset aan door gebruik te maken van “hekwerken”. De reden hiervoor wordt duidelijk wanneer u het log-dialoog gebruikt om revisies aan te geven om samen te voegen, waarbij elke revisie zichtbaar is als een wijzigingsset.

If you are merging revisions in chunks, the method shown in the Subversion book will have you merge 100-200 this time and 200-300 next time. With TortoiseSVN you would merge 100-200 this time and 201-300 next time.

Dit verschil heeft een hoop tam-tam in de mailing lijsten opgeleverd. We geven toe dat er een verschil is met de commandoregel client, maar we geloven dat het voor de meerderheid van de GUI-gebruikers de methode welke wij hebben toegepast gemakkelijker is om te begrijpen.

De makkelijkste methode om een reeks van revisies te selecteren die je nodig hebt, is door op de **Toon logboek** te klikken. Hiermee wordt een lijst met recente wijzigingen en de bijbehorende logboek boodschappen getoond. Als je de wijzigingen van een enkele revisie wilt samenvoegen, selecteer dan die ene revisie. Als je de wijzigingen van meerdere revisies wilt samenvoegen, selecteer dan de reeks terwijl je de **Shift** knop gebruikt. Klik op de **OK** knop en lijst met revisienummers die samengevoegd worden, wordt voor je gevuld.

Als je wijzigingen van een samenvoeging weer *uit* je werkkopie wilt halen, om een wijziging die al vastgelegd is weer ongedaan te maken, select dan de revisies die ongedaan gemaakt moeten worden en zorg er dan voor dat de optie **Omgekeerd samenvoegen** aangevinkt is.

Als je al enkele wijzigingen van deze tak hebt samengevoegd, hopelijk heb toen je in het logboek bericht opgenomen wat de laatste samengevoegde revisie was toen je de wijzigingen vastlegde. Je kunt dan **Toon logboek** gebruiken voor de werkkopie, om zo de logboek berichten te kunnen bekijken. Hou in gedachten dat we revisies als een set van wijzigingen zien, waardoor je als startpunt voor de nieuwe samenvoeging de eerste revisie na de vorige samenvoeging moet selecteren. Als je bijvoorbeeld de vorige keer revisie 37 naar revisie 39 hebt samengevoegd, dan moet je de nieuwe samenvoeging beginnen op revisie 40.

Als de **samenvoeg volg** functies van Subversion gebruikt, dan hoeft je niet te onthouden welke revisies je al samengevoegd hebt - Subversion onthoudt dat voor je. Als je de revisie reeks leeg laat, worden alle revisies die nog niet samengevoegd zijn meegenomen. Lees [Paragraaf 4.21.5, “Samenvoegen traceren”](#) voor meer informatie.

When merge tracking is used, the log dialog will show previously merged revisions, and revisions pre-dating the common ancestor point, i.e. before the branch was copied, as greyed out. The **Hide non-mergeable revisions** checkbox allows you to filter out these revisions completely so you see only the revisions which *can* be merged.

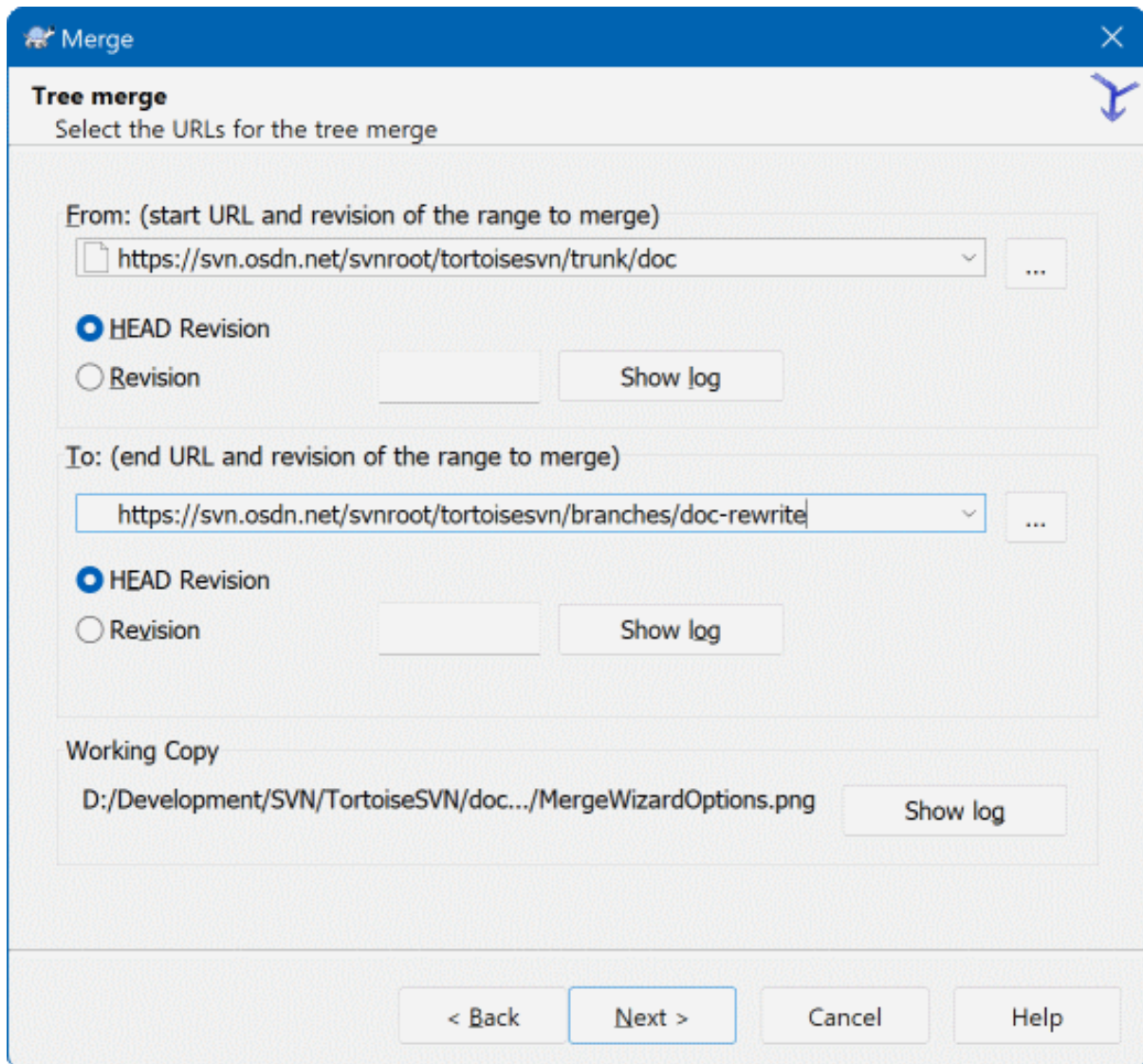
Als er andere personen mogelijk wijzigingen aan het vastleggen zijn, wees dan voorzichtig met het gebruik van de **HEAD** revisie. Het kan zijn dat deze niet wijst naar de revisie die je verwacht, bijvoorbeeld als er iemand wijzigingen heeft vastgelegd nadat jij de laatste keer ververst hebt.

If you leave the range of revisions empty or have the radio button **all revisions** checked, then Subversion merges all not-yet merged revisions. This is known as a reintegrate or automatic merge.

Er zijn enkele voorwaarden voor een reïntegratie samenvoeging. Ten eerste moet de server samenvoeg tracking ondersteunen. De werkkopie moet een onbegrensde diepte hebben (dus geen gedeeltelijk opgehaalde versie), er mogen geen lokale wijzigingen zijn gemaakt, objecten gewisseld zijn of objecten van een andere revisie dan de **HEAD** revisie. Alle wijzigingen die aan de basislijn zijn doorgevoerde gedurende de ontwikkeling van de tak, moeten in de tak samengevoegd zijn (of gemarkeerd als samengevoegd zijn). De reeks van de samen te voegen revisies wordt automatisch berekend.

Click Next and go to [Paragraaf 4.21.3, “Samenvoegopties”](#).

## 4.21.2. Twee Verschillende Takken Samenvoegen



**Afbeelding 4.56. Het Samenvoeg Hulpprogramma - Samenvoegen van Boomstructuren**

Als je deze methode gebruikt om een functie tak samen te voegen naar de basislijn, dan moet je het samenvoeg hulpprogramma starten vanuit de werkkopie van de basislijn.

Voer in het Van: veld de volledige URL van de map van de *basislijn*. Dit lijkt verkeerd te zijn, maar onthoudt dat de basislijn het startpunt is waar je de wijzigingen van de tak aan toe wilt voegen. Je kunt ook op de ... klikken om het archief te verkennen.

Voer in het Naar: veld de volledige URL van de map van de nieuwe functie tak.

Vul in zowel het Van Revisie veld als in het Naar Revisie veld het laatste revisie nummer in, waarop de beide boomstructuren gesynchroniseerd zijn. Als je er zeker van bent dat niemand anders wijzigingen vastlegt, dan kun je voor beide de HEAD revisie gebruiken. Als er een kans is dat iemand anders wijzigingen na de laatste synchronisatie heeft vastgelegd, gebruik dan de exacte revisie nummers om verlies van recente wijzigingen te voorkomen.

Je kunt ook met Toon logboek de revisie selecteren.

### 4.21.3. Samenvoegopties

Dit scherm van het hulpprogramma geeft je de mogelijkheid om geavanceerde opties in te stellen, voordat het samenvoegproces start. In de meeste gevallen kun je de standaard instellingen gebruiken.

Je kunt de diepte instellen die gebruikt moet worden voor het samenvoegen, dat houdt in hoe diep in je werkkopie de samenvoegactie doorgevoerd moet worden. De definitie van de diepte wordt beschreven in [Paragraaf 4.3.1, "Ophaaldiepte"](#). De standaard diepte staat ingesteld op *Werkkopie*, waardoor de bestaande diepte instelling gebruikt wordt. Dit is eigenlijk altijd de diepte die je wilt gebruiken.

Most of the time you want merge to take account of the file's history, so that changes relative to a common ancestor are merged. Sometimes you may need to merge files which are perhaps related, but not in your repository. For example you may have imported versions 1 and 2 of a third party library into two separate directories. Although they are logically related, Subversion has no knowledge of this because it only sees the tarballs you imported. If you attempt to merge the difference between these two trees you would see a complete removal followed by a complete add. To make Subversion use only path-based differences rather than history-based differences, check the *Ignore ancestry* box. Read more about this topic in the Subversion book, *Noticing or Ignoring Ancestry* [<http://svnbook.red-bean.com/en/1.8/svn.branchmerge.advanced.html#svn.branchmerge.advanced.ancestry>].

Je kunt ook opgeven hoe regeleinden, spaties en witregels behandeld moeten worden. Deze opties worden beschreven in [Paragraaf 4.11.2, "Opties voor regeleinden, spaties en witregels"](#). Standaard worden alle regeleinden, spaties en witregels als echte wijzigingen behandeld en samengevoegd.

Het selectievak *Forceer het samenvoegen* wordt gebruikt om mapstructuur conflicten te voorkomen, daar waar een binnenkomende verwijdering een bestand raakt die of lokaal gewijzigd is of niet onder versiebeheer valt. Als het bestand verwijderd is, dan is er geen enkele manier om deze te herstellen. Daarom is deze optie standaard niet ingeschakeld.

Als je gebruik maakt van het traceren van samenvoelingen en je wilt een revisie markeren als zijnde samengevoegd, zonder dat de feitelijke samenvoeging hier heeft plaatsgevonden, selecteer dan de *Samenvoegen alleen vastleggen* optie. Er zijn twee mogelijke redenen waarom je deze optie zou willen gebruiken. Het kan zijn dat het samenvoegen te gecompliceerd is voor de gebruikte algoritmes. Je codeert de wijzigingen dan handmatig, vervolgens markeer je de wijzigingen als samengevoegd, zodat het samenvoeg traceer algoritme dit ook ziet. De andere mogelijke reden is dat je wilt voorkomen dat een bepaalde revisie samengevoegd wordt. Door deze te markeren als reeds samengevoegd voorkom je dat de samenvoeging plaatsvindt, zolang de samenvoeg programma's rekening kunnen houden met de samenvoeg traceer informatie.

Alles is nu ingesteld en je hoeft alleen nog maar op de *Samenvoegen* knop te drukken. Als je vooraf het resultaat wilt beoordelen, kun je met *Test Samenvoegen* de samenvoeging simuleren. Hiermee wijzig je niets in je werkkopie. Het toont een lijst met bestanden die gewijzigd zullen worden bij de echte samenvoeg actie en geeft de bestanden aan waarmee conflicten op zouden *kunnen* treden. Omdat het traceren van de samenvoelingen het proces veel gecompliceerder maakt, is er geen manier om van te voren zeker te weten of er een conflict zal optreden of niet. Oftewel bestanden die als conflicterend worden aangemerkt bij een samenvoeg test hoeven in de praktijk helemaal geen probleem op te leveren.

The merge progress dialog shows each stage of the merge, with the revision ranges involved. This may indicate one more revision than you were expecting. For example if you asked to merge revision 123 the progress dialog will report "Merging revisions 122 through 123". To understand this you need to remember that Merge is closely related to Diff. The merge process works by generating a list of differences between two points in the repository, and applying those differences to your working copy. The progress dialog is simply showing the start and end points for the diff.

### 4.21.4. De Resultaten van het Samenvoegen Bekijken

Het samenvoegen is nu klaar. Het is verstandig om het resultaat te bekijken om na te gaan of het is zoals verwacht. Samenvoegen is vaak erg gecompliceerd. Conflicten komen vaak voor als een tak ver van de basislijn is afgeweken.





## Tip

Whenever revisions are merged into a working copy, TortoiseSVN generates a log message from all the merged revisions. Those are then available from the **Recent Messages** button in the commit dialog.

To customize that generated message, set the corresponding project properties on your working copy. See [Paragraaf 4.18.3.10, "Merge log message templates"](#)

Bij Subversion clients en servers van voor version 1.5, is er geen samenvoeg informatie opgeslagen en samengevoegde revisies moeten handmatig achterhaald worden. Als je de wijzigingen getest hebt en klaar bent om deze revisie vast te leggen, zou je logboek boodschap *altijd* de revisie nummers van de geporteerde revisies moeten bevatten. Als je op een later moment nog een samenvoeging wilt uitvoeren, dan moet je weten wat er al samengevoegd is, omdat je wijzigingen niet vaker dan één maal wilt doorvoeren. Lees voor meer informatie [Best Practices for Merging](http://svnbook.red-bean.com/en/1.4/svn.branchmerge.copychanges.html#svn.branchmerge.copychanges.bestprac) [http://svnbook.red-bean.com/en/1.4/svn.branchmerge.copychanges.html#svn.branchmerge.copychanges.bestprac] in het Subversion boek.

Als je server en alle clients Subversion versie 1.5 of hoger gebruiken, zullen de samenvoeg acties getraceerd worden, zodat voorkomen wordt dat wijzigingen meer dan eens worden toegepast. Dit maakt het leven een stuk makkelijker, omdat je dan elke keer de hele revisie reeks kunt samenvoegen en dat dan alleen de nieuwe revisies samengevoegd zullen worden.

Het managen van takken is belangrijk. Als je deze tak dicht bij de basislijn wilt houden, zul je vaak moeten samenvoegen, zodat de tak en de basislijn niet te ver uit elkaar drijven. Natuurlijk moet je het herhaaldelijk samenvoegen van wijzigingen voorkomen, zoals eerder beschreven.



## Tip

Als je net een functie tak hebt samengevoegd naar de basislijn, dan heeft de basislijn nu alle nieuwe functionaliteit en is de tak verouderd. Je kunt de tak nu verwijderen uit het archief indien nodig.



## Belangrijk

Subversion kan niet bestanden met mappen samenvoegen en vice versa - alleen mappen naar mappen en bestanden naar bestanden. Als je op een bestand klikt en het samenvoeg scherm opent, dan moet je het pad naar een bestand opgeven. Als je een map selecteert en samenvoeg scherm opent, dan moet je een URL van een map opgeven om samen te voegen.

### 4.21.5. Samenvoegen traceren

In Subversion 1.5 is functionaliteit geïntroduceerd voor het traceren van samenvoegingen. Als je wijzigingen vanuit de ene boomstructuur naar de andere samenvoegt, worden de samengevoegde revisie nummers opgeslagen. Deze informatie kan dan voor verschillende doeleinden gebruikt worden.

- Het is mogelijk om het gevaar van het twee keer samenvoegen van een revisie te voorkomen. Als eenmaal een revisie gemarkeerd is als samengevoegd, zal er bij toekomstige samenvoegen waarbij deze revisie in de reeks is opgenomen, deze revisie overgeslagen worden.
- Als je een tak terug naar de basislijn samenvoegt, dan kan het logboek scherm de vastleggingen aan de tak deel laten uitmaken van het logboek van de basislijn. Hierdoor is het makkelijker om wijzigingen te traceren.
- Als je het logboek vanuit het samenvoeg scherm opvraagt, worden de revisies die al samengevoegd zijn grijs weergegeven.
- Als je de verklarende informatie bekijkt voor een bestand, dan kun je er voor kiezen de originele auteur te tonen van samengevoegde revisie, in plaats van de persoon die de samenvoeging uitvoerde.

- Je kunt revisies markeren met *niet samenvoegen* door deze al in de lijst van samengevoegde revisies op te nemen, zonder de echte samenvoeging uit te voeren.

De samenvoeg informatie is opgeslagen in de `svn:mergeinfo` eigenschap door de client als deze een samenvoeging uitvoert. Als de samenvoeging vastgelegd wordt, dan slaat de server de gegevens op in de database. Als je samenvoeg, logboek of verklaren informatie opvraagt, dan kan de server deze informatie doorgeven. Om dit goed te laten werken moet je er voor zorgen dat de server, het archief en alle clients naar de nieuwe versie bijgewerkt worden. Oudere clients zullen de `svn:mergeinfo` eigenschappen niet opslaan en oudere servers zullen de door de clients opgevraagde informatie niet opgeven.

Find out more about merge tracking from Subversion's [Merge tracking documentation](http://svn.apache.org/repos/asf/Subversion/trunk/notes/merge-tracking/index.html) [http://svn.apache.org/repos/asf/Subversion/trunk/notes/merge-tracking/index.html].

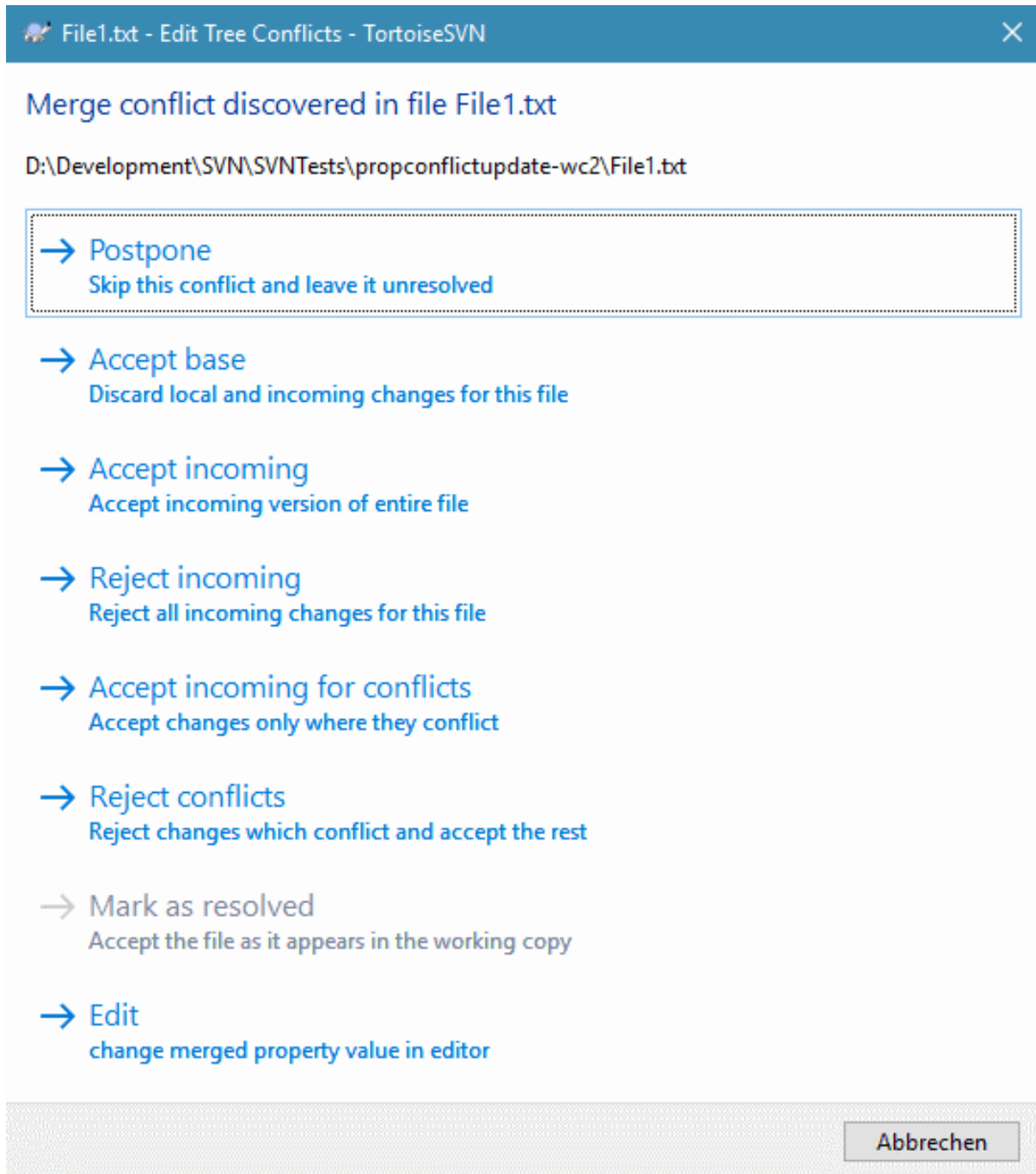
#### 4.21.6. Handling Conflicts after Merge



##### **Belangrijk**

The text in the conflict resolver dialogs are provided by the SVN library and might therefore not (yet) be translated as the TortoiseSVN dialogs are. Sorry for that.

Merging does not always go smoothly. Sometimes there is a conflict. TortoiseSVN helps you through this process by showing the *merge conflict* dialog.



### Afbeelding 4.57. The Merge Conflict Dialog

It is likely that some of the changes will have merged smoothly, while other local changes conflict with changes already committed to the repository. All changes which can be merged are merged. The Merge Conflict dialog gives you different ways of handling the lines which are in conflict.

For normal conflicts that happen due to changes in the file content or its properties, the dialog shows buttons which allow you to choose which of the conflicting parts to keep or reject.

#### Postpone

Don't deal with the conflict now. Let the merge continue and resolve the conflicts after the merge is done.

#### Accept base

This leaves the file as it was, without neither the changes coming from the merge nor the changes you've made in your working copy.

**Accept incoming**

This discards all your local changes and uses the file as it arrives from the merge source.

**Reject incoming**

This discards all the changes from the merge source and leaves the file with your local edits.

**Accept incoming for conflicts**

This discards your local changes where they conflict with the changes from the merge source. But it leaves all your local changes which don't conflict.

**Reject conflicts**

This discards changes from the merge source which conflict with your local changes. But it keeps all changes that don't conflict with your local changes.

**Mark as resolved**

Marks the conflicts as resolved. This button is disabled until you use the button **Edit** to edit the conflict manually and save those changes back to the file. Once the changes are saved, the button becomes enabled.

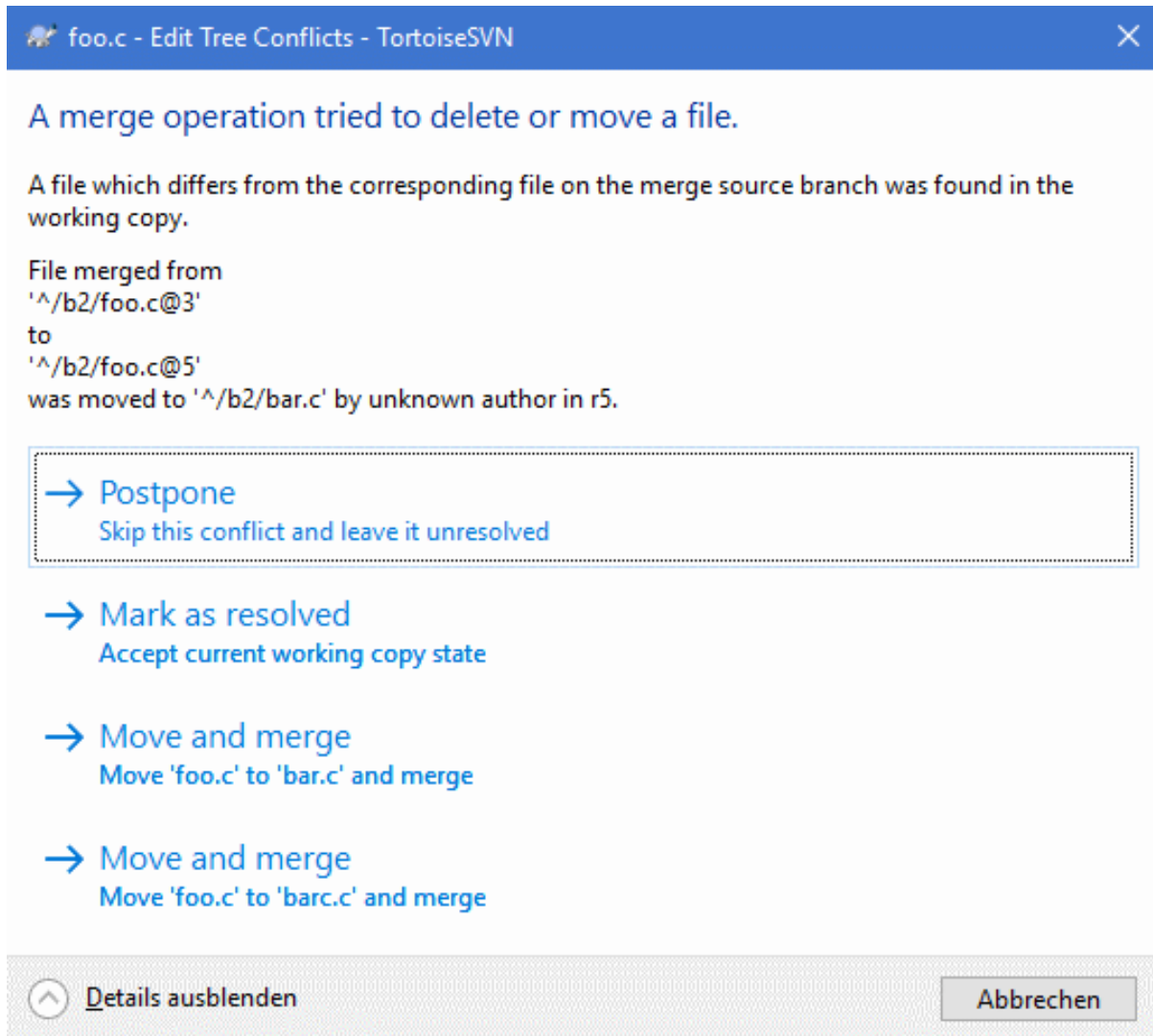
**Wijzigen**

Starts the merge editor so you can resolve the conflicts manually. Don't forget to save the file so the button **Mark as resolved** becomes enabled.

If there's a tree conflict, please first see [Paragraaf 4.6.3, "Mapstructuur conflicten"](#) about the various types of tree conflicts and how and why they can happen.

To resolve tree conflicts after a merge, a dialog is shown with various options on how to resolve the conflict:





#### Afbeelding 4.58. The Merge Tree Conflict Dialog

Since there are various possible tree conflict situations, the dialog will show buttons to resolve those depending on the specific conflict. The button texts and labels explain what the option to resolve the conflict does. If you're not sure, either cancel the dialog or use the Postpone button to resolve the conflict later.

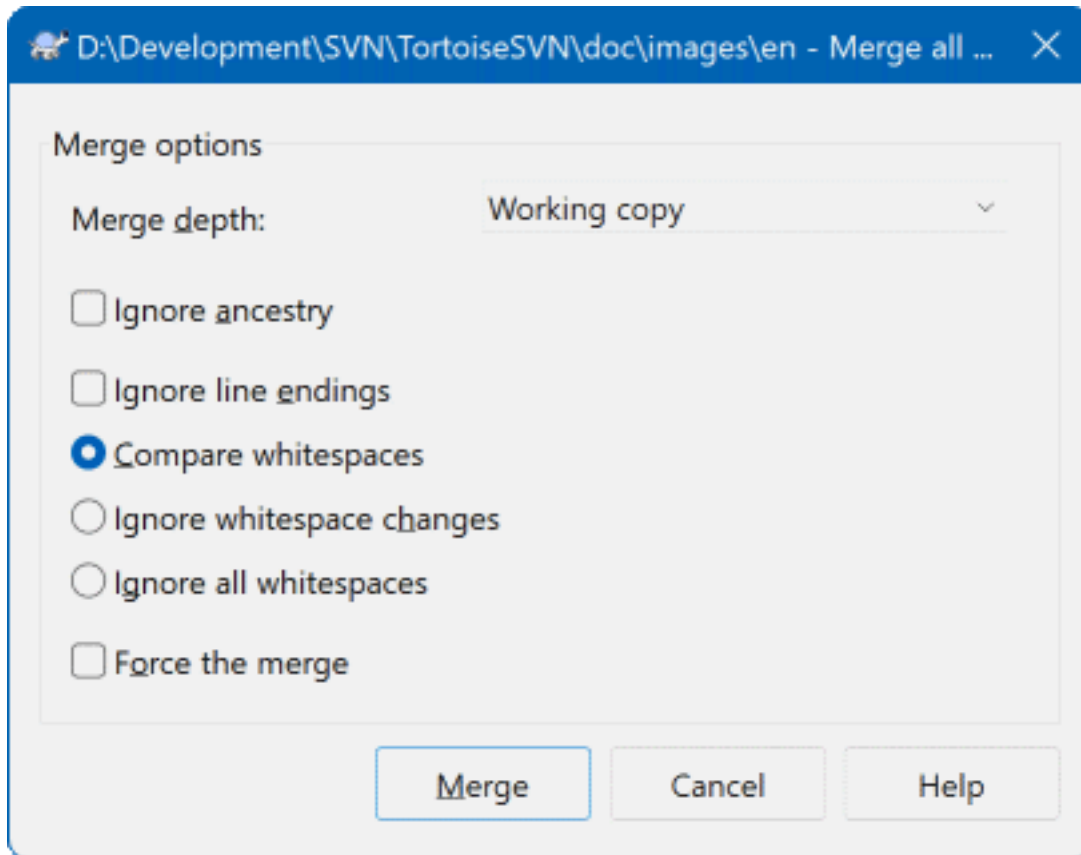
#### 4.21.7. Onderhoud aan een Tak met Nieuwe Functies

Als je een nieuwe functie ontwikkelt in een aparte tak, dan is het verstandig om een beleid af te spreken voor het reïntegreren als de functie klaar is. Als er op het zelfde moment gewoon doorgewerkt wordt aan de basislijn, dan kun je er tegenaan lopen dat wijzigingen steeds meer gevolgen hebben, waardoor het samenvoegen echt een nachtmerrie kan worden.

Als de nieuwe functie relatief eenvoudig is en de ontwikkeling niet lang duurt, dan kun je een vrij eenvoudige aanpak gebruiken. Hou de tak helemaal gescheiden totdat de functie klaar is en voeg de wijzigingen in de tak dan samen naar de basislijn. In het samenvoeg hulpprogramma zou die een eenvoudige Voeg een reeks revisies samen actie zijn met de reeks van revisies van de tak.

Als de ontwikkeling langer duurt en je rekening moet gaan houden met wijzigingen in de basislijn, dan moet je je tak synchronoos houden met de basislijn. Dit houdt gewoon in dat je regelmatig de wijzigingen in de basislijn moet samenvoegen naar je tak, zodat de tak de wijzigingen van de basislijn *plus* de nieuwe functie heeft. Het synchronisatie proces gebruikt de Voeg een reeks revisies samen functie. Als de functie klaar is, kun je deze samenvoegen naar de basislijn met Reïntegreer een tak of met Twee verschillende bomen samenvoegen.

Another (fast) way to merge all changes from trunk to the feature branch is to use the TortoiseSVN → Merge all... from the extended context menu (hold down the **Shift** key while you right click on the file).



**Afbeelding 4.59. The Merge-All Dialog**

This dialog is very easy. All you have to do is set the options for the merge, as described in [Paragraaf 4.21.3, “Samenvoegopties”](#). The rest is done by TortoiseSVN automatically using merge tracking.

## 4.22. Blokkeren

Subversion werkt het beste als er geen gebruik wordt gemaakt van blokkeren, als je gebruik maakt van de “Kopieer-Wijzig-Samenvoeg” methode die eerder beschreven is in [Paragraaf 2.2.3, “De Kopieer-Wijzig-Samenvoeg Oplossing”](#). Er zijn echter enkele gevallen waarbij je een zekere vorm van blokkeerbeleid moet gebruiken.

- Je maakt gebruik van “niet samen te voegen” bestanden, bijvoorbeeld grafische bestanden. Als twee mensen hetzelfde bestand wijzigen, dan is samenvoegen niet mogelijk. Oftewel, één van de twee zal de wijzigingen kwijtraken.
- Jouw bedrijf heeft in het verleden altijd gebruik gemaakt van een versiebeheersysteem met blokkeringsmethode en er is een management besluit genomen dat “blokkeren het beste is”.

Allereerst moet je er zeker van zijn dat je Subversion server is bijgewerkt tot minimaal versie 1.2. Eerdere versies ondersteunen blokkeren helemaal niet. Als je gebruik maakt van `file://`, hoeft natuurlijk alleen maar je eigen programma bij te werken.



### The Three Meanings of “Lock”

In this section, and almost everywhere in this book, the words “lock” and “locking” describe a mechanism for mutual exclusion between users to avoid clashing commits. Unfortunately, there are two other sorts of “lock” with which Subversion, and therefore this book, sometimes needs to be concerned.

The second is *working copy locks*, used internally by Subversion to prevent clashes between multiple Subversion clients operating on the same working copy. Usually you get these locks whenever a command like `update/commit/...` is interrupted due to an error. These locks can be removed by running the cleanup command on the working copy, as described in [Paragraaf 4.17, “Schonen”](#).

And third, files and folders can get locked if they're in use by another process, for example if you have a word document opened in Word, that file is locked and can not be accessed by TortoiseSVN.

You can generally forget about these other kinds of locks until something goes wrong that requires you to care about them. In this book, “lock” means the first sort unless the contrary is either clear from context or explicitly stated.

### 4.22.1. Hoe Het Blokkeren Werkt In Subversion

Standaard wordt er niets geblokkeerd en kan iedereen die de rechten heeft wijzigingen op elk moment vastleggen in het archief. Anderen zullen hun werkkopieën regelmatig moeten verversen en de wijzigingen in het archief worden dan samengevoegd met de lokale wijzigingen.

Als je een bestand *blokkeert*, dan heb jij alleenrecht op het vastleggen van dat bestand. Vastleggingen van andere gebruikers zullen geblokkeerd worden totdat jij de blokkade opheft. Een geblokkeerd bestand kan op geen enkele manier in het archief gewijzigd worden, dat houdt ook in dat niemand anders dan de eigenaar van de blokkade het bestand kan verwijderen of hernoemen.



#### Belangrijk

A lock is not assigned to a specific user, but to a specific user and a working copy. Having a lock in one working copy also prevents the same user from committing the locked file from another working copy.

As an example, imagine that user Jon has a working copy on his office PC. There he starts working on an image, and therefore acquires a lock on that file. When he leaves his office he's not finished yet with that file, so he doesn't release that lock. Back at home Jon also has a working copy and decides to work a little more on the project. But he can't modify or commit that same image file, because the lock for that file resides in his working copy in the office.

Andere gebruikers zullen echter niet direct weten dat jij een bestand geblokkeerd hebt, tenzij ze regelmatig de blokkade status opvragen. Het eerste moment waarop ze er meestal pas achter komen is als ze wijzigingen proberen vast te leggen, wat niet altijd handig is. Om het werken met blokkades te vereenvoudigen is er een nieuwe Subversion eigenschap geïntroduceerd, `svn:needs-lock`. Als deze eigenschap (met een willekeurige waarde) aan een bestand is gekoppeld, dan wordt het lokale bestand altijd als alleen-lezen gemarkeerd als deze opgehaald wordt uit het archief, *tenzij/totdat* die werkkopie het bestand blokkeert. Op deze manier wordt je gewaarschuwd dat je dit bestand niet moet wijzigen voordat je het geblokkeerd hebt. Bestanden, die voorzien zijn van versiebeheer en als alleen-lezen zijn gemarkeerd, hebben in TortoiseSVN een apart overlappend pictogram waarmee aangegeven wordt dat je eerst het bestand moet blokkeren.

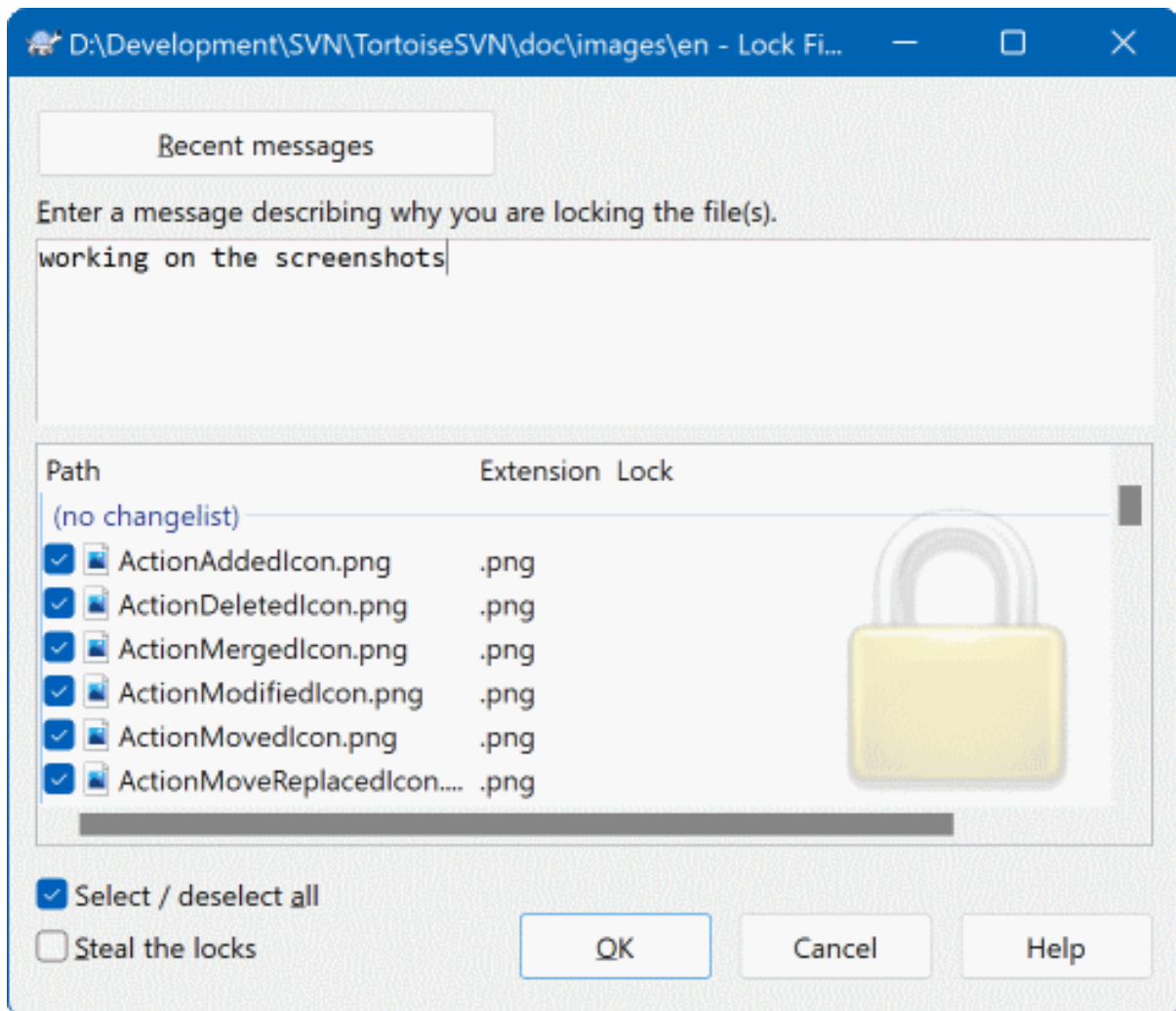
Blokkades worden geadmineistreerd met de werkkopie locatie en de eigenaar. Als je meerdere werkkopieën hebt (thuis, op het werk), dan kun je in maar *één* werkkopie zo'n blokkade initiëren.

Stel dat één van je medewerkers zo'n blokkade instelt en vervolgens op vakantie gaat, wat doe je dan? Subversion heeft de mogelijkheid om blokkades te forceren. Het vrijgeven van een blokkade die door iemand anders is ingesteld wordt aangeduid als het *Verbreken* van een blokkade. Het geforceerd toe-eigenen van een blokkade die door iemand anders is ingesteld, wordt het *Stelen* van een blokkade genoemd. Hier moet je natuurlijk voorzichtig mee omgaan als je vrienden wilt blijven met je collega's.

Blokkades worden geadmineistreerd in het archief en er wordt ook een merkteken (een zogenaamde token) in je lokale werkkopie geplaatst. Als er een discrepantie is opgetreden, bijvoorbeeld als iemand anders de blokkade geforceerd heeft, dan zal het lokale blokkade merkteken ongeldig worden. De status in het archief is altijd leidend.

### 4.22.2. Blokkade verkrijgen

Selecteer de bestanden in je werkkopie die je wilt blokkeren. Vervolgens selecteer je het commando TortoiseSVN → Plaats blokkade....



#### Afbeelding 4.60. Het blokkeringsscherm

Een apart venster verschijnt, waarin je aan anderen kunt aangeven waarom je de bestanden geblokkeerd hebt. Het invullen is optioneel en wordt op dit moment alleen gebruikt in archieven die op basis van Svnserve opgezet zijn. Alleen als je *echt* een blokkade moet stelen van iemand anders, moet je de **Steal de blokkades** selectie aanvinken om de blokkade te stelen. Klik vervolgens op OK om de actie in gang te zetten.

You can set the project property `tsvn:logtemplatelock` to provide a message template for users to fill in as the lock message. Refer to [Paragraaf 4.18, "Project Instellingen"](#) for instructions on how to set properties.

Als je een map selecteert en vervolgens het commando TortoiseSVN → Plaats blokkade... kiest, zal het blokkeringsscherm geopend worden voor *elk* bestand in de map *en alle* bestanden in de sub-mappen. Als je echt de hele hiërarchie wilt blokkeren, dan is dat de methode. Je collega's zullen er mogelijk niet blij van worden als je hun helemaal buiten het project sluit. Gebruik dit dus met verstand...

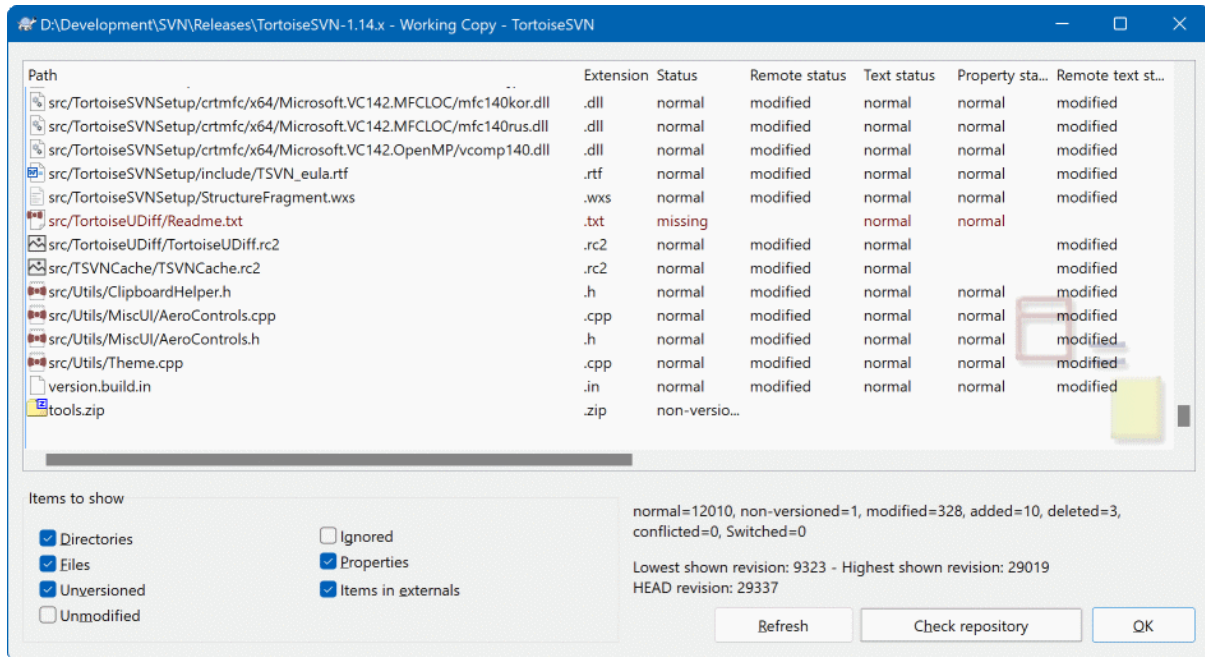
### 4.22.3. Een Blokkade Vrijgeven

Om te voorkomen dat je vergeet een blokkade vrij te geven, worden geblokkeerde bestanden weergegeven in het vastleggen venster en standaard geselecteerd. Als je dan doorgaat met het vastleggen, zullen blokkades van de geselecteerde bestanden verwijderd worden, zelfs als er niets gewijzigd is. Als je de blokkade op bepaalde

bestanden niet wilt vrijgeven, dan kun je deze de-selecteren (als ze niet gewijzigd zijn). Als je de blokkade op gewijzigde bestanden wilt behouden, dan moet je het **Blokkades handhaven** selectievak aanvinken voordat je je wijzigingen vastlegt.

Als je een blokkade handmatig wilt opheffen, moet je de bestanden in je werkkopie selecteren waarvoor je de blokkade wilt opheffen. Vervolgens kies je het commando TortoiseSVN → **Blokkade opheffen**. Er hoeft verder geen informatie ingevoerd te worden, dus TortoiseSVN zal het archief benaderen en de blokkades opheffen. Je kunt dit commando ook op mappen toepassen om alle blokkades recursief op te heffen.

#### 4.22.4. Blokkadestatus Controleren



#### Afbeelding 4.61. Het Kijk of er updates zijn scherm

Om na te gaan welke blokkades er door jou of anderen geplaatst zijn, kun je gebruik maken van TortoiseSVN → **Kijk of er updates zijn...** Lokale blokkade merktekens worden meteen getoond. Om de blokkades van anderen te zien (en om na te gaan of jouw blokkades verbroken of gestolen zijn) moet je op **Controleer archief** klikken.

Via het context menu, kun je hier ook blokkades plaatsen en opheffen, maar ook blokkades van anderen verbreken en stelen.



#### Voorkom het verbreken en stelen van blokkades

Het verbreken of stelen van iemand anders blokkade zonder dat die persoon daarvan op de hoogte is, kan verlies van gegevens tot gevolg hebben. Als je iemand zijn blokkade steelt en werkt met niet samenvoegbare bestandsformaten, dan kan iemand anders jouw wijzigingen overschrijven op het moment dat jij jouw blokkade opheft en zij hun wijzigingen vastleggen in het archief. Subversion raakt geen informatie kwijt, maar je verliest wel de beveiliging de blokkade functionaliteit je biedt bij het werken in teamverband.

#### 4.22.5. Niet-geblokkeerde bestanden als Alleen-lezen instellen

Zoals eerder beschreven, kun je blokkades het effectiefst gebruiken door gebruik te maken van de `svn:needs-lock` eigenschap. Lees [Paragraaf 4.18, "Project Instellingen"](#) voor meer informatie over het instellen van eigenschappen. Bestanden die deze eigenschap hebben, krijgen meteen de alleen-lezen eigenschap mee als ze worden opgehaald uit het archief, tenzij jouw werkkopie het bestand geblokkeerd heeft.





Ter herinnering, TortoiseSVN gebruikt een apart overlappend pictogram om deze situatie aan te geven.

Als je een beleid hanteert waardoor elk bestand geblokkeerd moet worden, dan is het waarschijnlijk makkelijker om automatische-eigenschappen functie van Subversion te gebruiken. Hiermee wordt automatisch dezelfde eigenschap ingesteld als je een nieuwe bestand toevoegt. Lees ook [Paragraaf 4.18.1.5, “Automatisch eigenschappen zetten”](#) voor meer informatie.

#### 4.22.6. Hook Scripts Voor Blokkeringen

Als je een nieuw archief aanmaakt met Subversion 1.2 of hoger, worden er vier hook sjablonen aangemaakt in de map `hooks` van het archief. Deze worden aangeroepen voor en na het plaatsen van een blokkade, en voor en na het opheffen van een blokkade.

Het is verstandig om op de server een `na-het-blokkeren` en `na-het-deblokkeren` hook script te installeren, waardoor een email verstuurd wordt waarin aangegeven wordt welk bestand ge(de)blokkeerd is. Met zo'n script kunnen alle gebruikers geïnformeerd worden over het feit dat er een blokkade geplaatst of opgeheven wordt. Je kunt een voorbeeld `hooks/post-lock.tmpl` vinden in je archief.

Je kunt ook hook scripts gebruiken om het verbreken of stelen van blokkades te voorkomen, of dat alleen een beheerder dit mag doen. Je kunt ook een email laten sturen naar de eigenaar van een blokkade als zijn blokkade verbroken of gestolen wordt.

Lees [Paragraaf 3.3, “Server gesitueerde hook scripts”](#) voor meer informatie hierover.

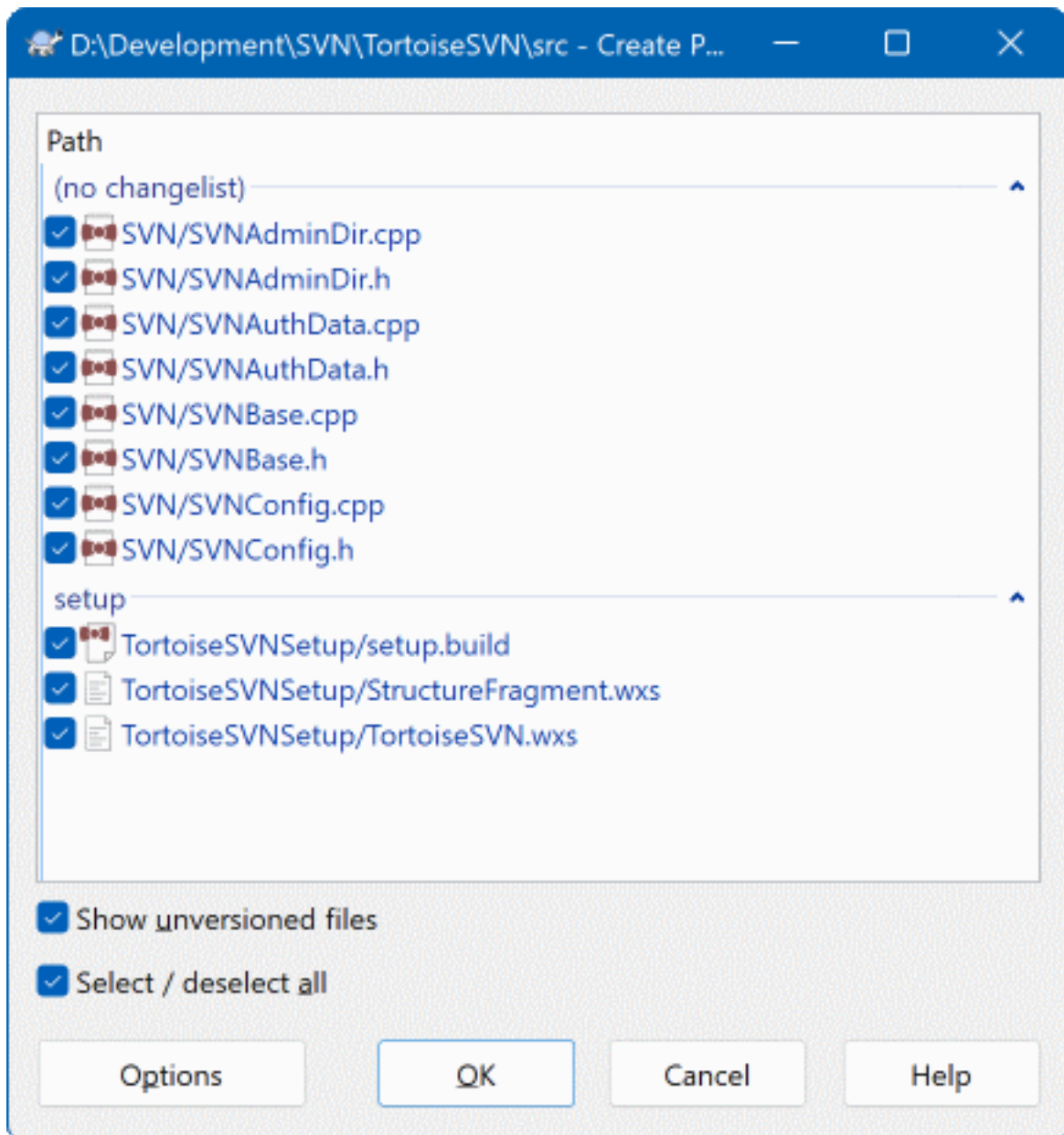
### 4.23. Een Patch Maken en Toepassen

Bij Open bron projecten (zoals deze) heeft iedereen leesrechten voor het archief en kan iedereen een bijdrage leveren aan het project. Hoe worden die bijdrages behandeld? Als iedereen wijzigingen kan vastleggen, dan zal het project bijna altijd onstabiel zijn en zelfs in de meeste gevallen niet werken. Om dit te voorkomen wordt er gebruik gemaakt van *patch* bestanden die naar het ontwikkelteam gestuurd kunnen worden, die natuurlijk wel schrijfrechten heeft. Zij kunnen de patch dan controleren en in het archief inbrengen of de auteur laten weten dat deze afgewezen is.

Patch bestanden zijn Unified-Diff bestanden waarin de verschillen tussen jouw werkkopie en de basis revisie zijn opgenomen.

#### 4.23.1. Maken van een Patchbestand

Eerst moet je natuurlijk je wijzigingen maken *en testen*. Vervolgens moet je op de bovenliggende map TortoiseSVN → Maak Patch... selecteren in plaats van TortoiseSVN → Vastleggen...



**Afbeelding 4.62. Het Maak Patch Venster**

Nu kun je de bestanden selecteren die je in je patch wilt opnemen, net zoals je zou doen bij een normale vastlegging. Hiermee wordt dan een enkel bestand gemaakt met daarin een samenvatting van alle wijzigingen die je aan de geselecteerde bestanden hebt aangebracht sinds de laatste verversing van het archief.

Je kunt de kolommen in dit dialoogscherf configureren op dezelfde manier als die van het dialoogscherf Kijk of er updates zijn. Ga naar [Paragraaf 4.7.3, “Lokale en Server Status”](#) voor meer informatie.

By clicking on the Options button you can specify how the patch is created. For example you can specify that changes in line endings or whitespaces are not included in the final patch file.

Je kunt verschillende patches aanmaken met wijzigingen aan verschillende bestandsgroepen. Als je een patch bestand aanmaakt, dan nog enkele wijzigingen aan *dezelfde* bestanden doorvoert en vervolgens weer een patch bestand aanmaakt, dan zal in het tweede patch bestand *beide* sets met wijzigingen te zien zijn.

Sla het bestand op met een naam naar keuze. Patch bestanden kunnen elke extensie hebben die je leuk vindt, maar standaard wordt de extensie `.patch` of `.diff` gebruikt. Nu ben je klaar om je patch bestand op te sturen.



### Tip

Do not save the patch file with a `.txt` extension if you intend to send it via email to someone else. Plain text files are often mangled with by the email software and it often happens that whitespaces and newline chars are automatically converted and compressed. If that happens, the patch won't apply smoothly. So use `.patch` or `.diff` as the extension when you save the patch file.

Je kunt je patch ook op het klembord opslaan in plaats van in een bestand. Dit is handig als je de patch bijvoorbeeld in een mail wilt plakken zodat iemand anders hem kan controleren. Je kunt deze methode ook gebruiken als je twee PC's gebruikt en je de wijzigingen van de ene naar de andere PC wilt verplaatsen.

If you prefer, you can create a patch file from within the Commit or Check for Modifications dialogs. Just select the files and use the context menu item to create a patch from those files. If you want to see the Options dialog you have to hold **shift** when you right click.

## 4.23.2. Een Patchbestand Toepassen

Patch bestanden worden toegepast op je werkkopie. Dit moet gedaan worden vanaf hetzelfde map-niveau als waarop de patch gemaakt is. Als je niet zeker weet op welk niveau dat was, kijk dan naar de eerste regel van het patch bestand. Als bijvoorbeeld het eerste bestand waaraan gewerkt is `doc/source/english/chapter1.xml` is en de eerste regel in het patch bestand is `Index: english/chapter1.xml`, dan moet je de patch toepassen op de map `doc/source/`. Als je per ongeluk het verkeerde map-niveau kiest, dan zal TortoiseSVN je hiervan op de hoogte stellen en het juiste niveau voorstellen. Dit werkt alleen als je in de juiste werkkopie aan het werken bent.

Als je een patch bestand wilt toepassen op je werkkopie, dan heb je op zijn minst lees-rechten nodig bij het archief. De reden hiervoor is dat het samenvoegprogramma de revisie moet kunnen traceren waarop de wijzigingen door de ontwikkelaar zijn gemaakt.

Klik in het context menu van die folder op TortoiseSVN → Pas Patch toe.... Er wordt dan een nieuw scherm geopend waarin je het patch bestand, die je wilt toepassen, kunt selecteren. Standaard zullen alleen `.patch` of `.diff` bestanden weergegeven worden, maar je kunt ook “Alle bestanden” laten zien. Als je een patch op het klembord hebt staan, dan kun je de knop Openen vanaf klembord... klikken. Merk op dat deze optie alleen weergegeven wordt als je de patch op het klembord hebt gezet met TortoiseSVN → Maak Patch.... De knop zal niet getoond worden als je een patch vanuit een ander programma naar het klembord hebt gekopieerd.

Als jouw patch bestand de extensie `.patch` of `.diff` heeft, kun je deze ook direct toepassen als je met rechts op het bestand klikt en TortoiseSVN → Pas Patch toe... selecteert.

Dit zijn twee verschillende manieren waarmee je hetzelfde bereikt. Met de eerste methode selecteer je de werkkopie en blader je naar het patch bestand. Met de tweede selecteer je het patch bestand en blader je naar de werkkopie.

Als je eenmaal het patch bestand en de locatie van de werkkopie hebt geselecteerd, zal TortoiseMerge de wijzigingen van het patch bestand met je werkkopie samenvoegen. In een klein scherm zullen de gewijzigde bestanden weergegeven worden. Dubbelklik op elk bestand om de wijzigingen van de samenvoeging te bekijken en deze op te slaan.

De patch van de andere ontwikkelaar is nu toegepast op jouw werkkopie. Je moet nu de wijzigingen vastleggen in het archief, zodat iedereen de wijzigingen kan ophalen.

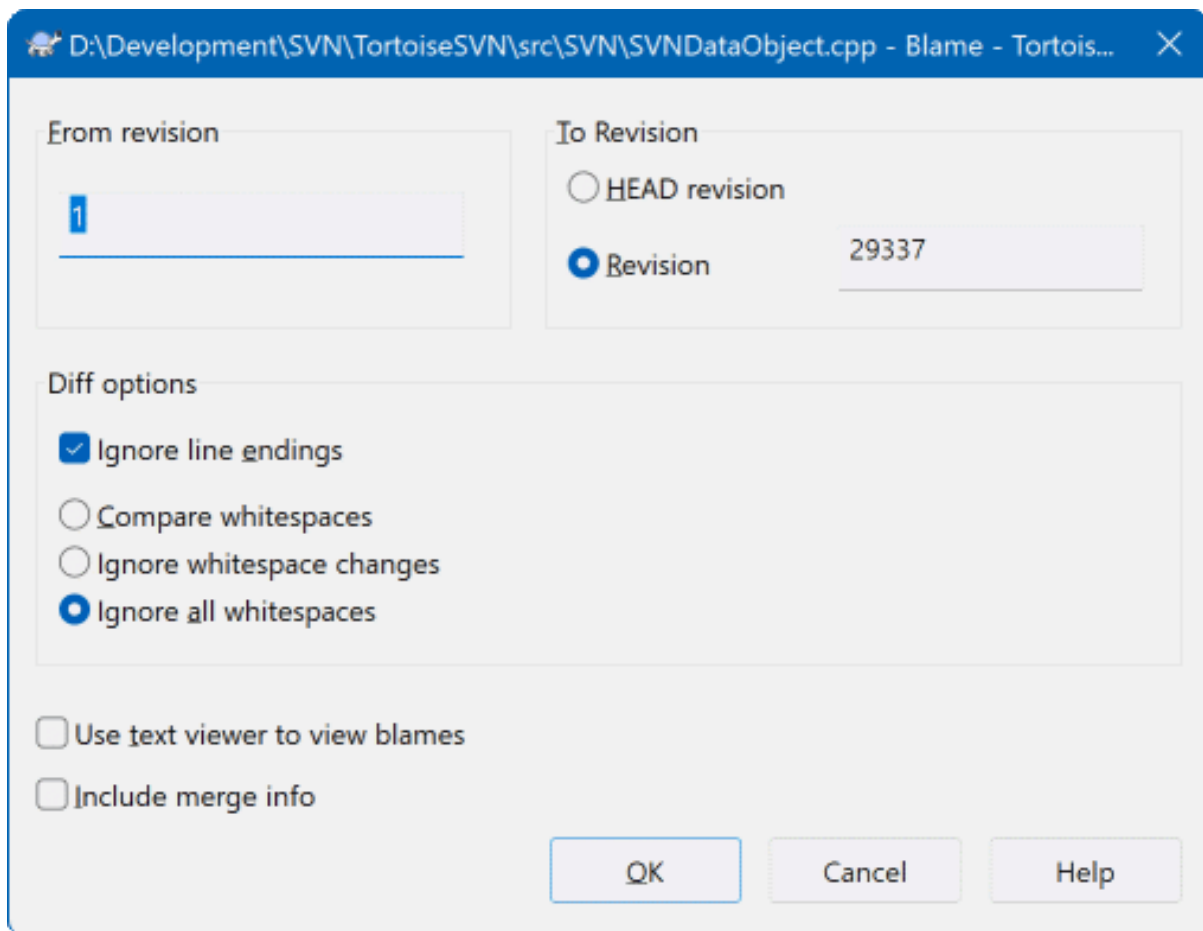
## 4.24. Wie Heeft Welke Regel Gewijzigd?

Soms wil je niet alleen welke regels er gewijzigd zijn, maar ook wie er precies welke regel gewijzigd heeft. Hierbij komt het commando TortoiseSVN → Verklaren..., soms ook aangeduid als *annoteren*, goed van pas.

Dit commando laat voor elke regel in een bestand zien wie, en in welke revisie, een regel heeft gewijzigd.



#### 4.24.1. Verklar Bestanden



**Afbeelding 4.63. Het Verklar/Annoteren Scherm**

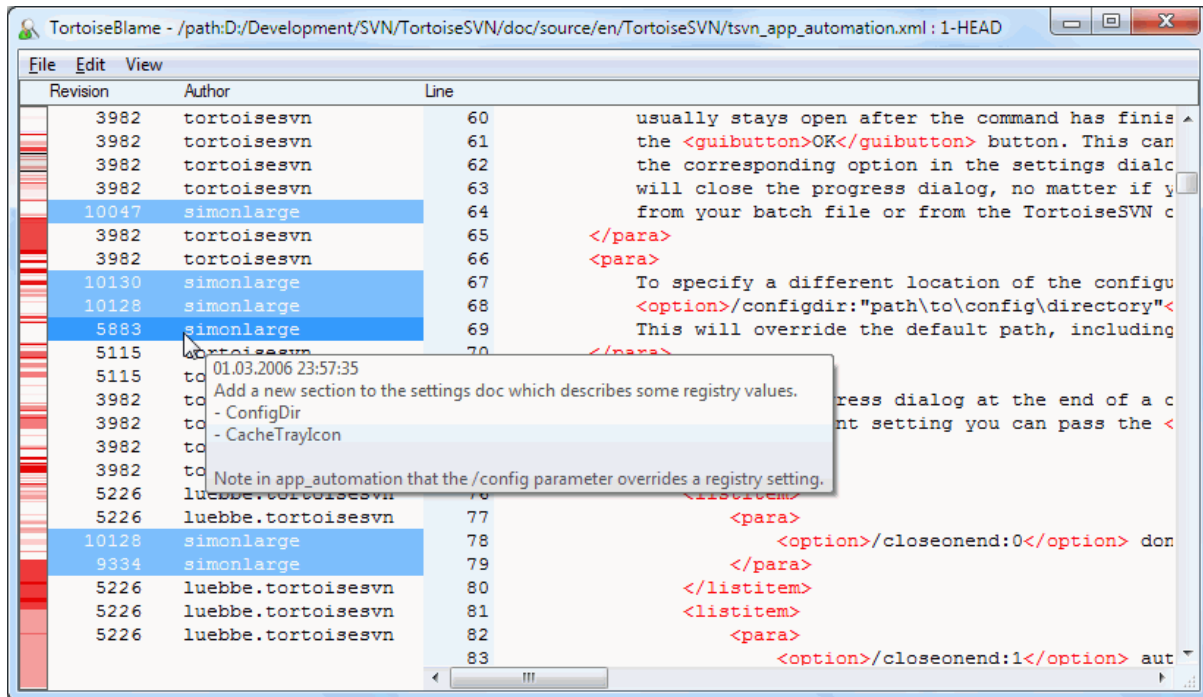
Als je niet geïnteresseerd bent in de wijzigingen van eerdere revisies, kun je aangegeven vanaf welke revisie het verklaren moet beginnen. Stel deze waarde in op 1 als je de wijzigingen voor *elke* revisie wilt zien.

By default the blame file is viewed using *TortoiseBlame*, which highlights the different revisions to make it easier to read. If you wish to print or edit the blame file, select **Use Text viewer to view blames**.

Je kunt instellen hoe regeleinden en spaties behandeld moeten worden. Deze opties staan beschreven in [Paragraaf 4.11.2, “Opties voor regeleinden, spaties en witregels”](#). Standaard worden alle verschillen in spaties en regeleinden als echte wijzigingen behandeld. Als je een inspring wijziging wilt negeren en de originele auteur wilt vinden, dan kun je de benodigde instellingen hier kiezen.

You can include merge information as well if you wish, although this option can take considerably longer to retrieve from the server. When lines are merged from another source, the blame information shows the revision the change was made in the original source as well as the revision when it was merged into this file.

Once you press OK TortoiseSVN starts retrieving the data to create the blame file. Once the blame process has finished the result is written into a temporary file and you can view the results.



**Afbeelding 4.64. TortoiseBlame**

TortoiseBlame, die met TortoiseSVN wordt meegeleverd, maakt het lezen van verklaren-bestanden makkelijker. Als je je muis boven een regel in de verklaren kolom houdt, dan worden alle regels van dezelfde revisie weergegeven met een donkere achtergrond. Regels van andere revisies die door dezelfde auteur gewijzigd zijn, worden weergegeven met een lichte achtergrond. De kleuren worden niet optimaal weergegeven als je je scherm hebt ingesteld op 256 kleuren mode.

Als je met links op een regel klikt, worden alle regels van dezelfde revisie gemarkeerd en regels van andere revisies van dezelfde auteur worden met een lichtere kleur gemarkeerd. De markeringen blijven actief, waardoor je je muis kunt bewegen zonder dat de markeringen verloren gaan. Als je weer op de revisie klikt, worden de markeringen weer uitgezet.

De revisie opmerkingen (logboek boodschappen) worden in een klein hint scherm weergegeven als je je muis boven de verklaring informatie kolom houdt. Als je de logboek boodschap van die revisie wilt kopiëren, gebruik je het context menu door met rechts te klikken op de verklaring informatie kolom.

Je kunt in het verklaren rapport zoeken met **Wijzigen → Zoeken...** Hiermee kun je zoeken op revisienummers, auteurs en de inhoud van het bestand zelf. Logboek boodschappen worden niet meegenomen in de zoekopdracht - hiervoor moet je het Logboek venster gebruiken.

Je kunt ook naar een specifiek regelnummer springen met **Wijzigen → Ga naar regel...**

In de verklaring informatie kolom is een context menu beschikbaar, waarmee je makkelijk revisies kunt vergelijken en de geschiedenis kunt onderzoeken met het revisienummer van de regel waar de muis boven gehouden wordt als referentie. **Context menu → Verklaar vorige revisie** genereert een verklaar rapport voor hetzelfde bestand, maar dan met de vorige revisie als bovenste limiet. Hiermee krijg je een verklaar rapport voor de staat van het bestand vlak voordat de regel die je bekijkt gewijzigd werd. **Context menu → Toon wijzigingen** start jou vergelijk programma, waarmee dan de wijzigingen van de gerefereerde revisie worden weergegeven. **Context menu → Toon logboek** toont het revisie log venster vanaf de gerefereerde revisie.

Als je een betere visuele indicatie wilt hebben van waar de oudste en nieuwe wijzigingen staan, selecteer dan **Bekijk → Kleur regels naar leeftijd**. Hiermee wordt een kleur verloop gebruikt om nieuwe regels met een rode

kleur weer te geven en oudere met een blauwe kleur. De standaard kleurinstelling is vrij licht, maar dat kun je wijzigen bij de TortoiseBlame instellingen.

If you are using Merge Tracking and you requested merge info when starting the blame, merged lines are shown slightly differently. Where a line has changed as a result of merging from another path, TortoiseBlame will show the revision and author of the last change in the original file rather than the revision where the merge took place. These lines are indicated by showing the revision and author in italics. The revision where the merge took place is shown separately in the tooltip when you hover the mouse over the blame info columns. If you do not want merged lines shown in this way, uncheck the **Include merge info** checkbox when starting the blame.

If you want to see the paths involved in the merge, select **View → Merge paths**. This shows the path where the line was last changed, excluding changes resulting from a merge.

The revision shown in the blame information represents the last revision where the content of that line changed. If the file was created by copying another file, then until you change a line, its blame revision will show the last change in the original source file, not the revision where the copy was made. This also applies to the paths shown with merge info. The path shows the repository location where the last change was made to that line.

De instellingen van TortoiseBlame kunnen bekeken worden bij **TortoiseSVN → Instellingen...** onder de TortoiseBlame tab. Lees ook [Paragraaf 4.31.9, “TortoiseBlame Instellingen”](#).

#### 4.24.2. Verklaar Verschillen

One of the limitations of the Blame report is that it only shows the file as it was in a particular revision, and the last person to change each line. Sometimes you want to know what change was made, as well as who made it. If you right click on a line in TortoiseBlame you have a context menu item to show the changes made in that revision. But if you want to see the changes *and* the blame information simultaneously then you need a combination of the diff and blame reports.

Het revisie log venster heeft enkele opties waarmee je dit kunt doen.

##### Verklaar Revisies

Selecteer in het bovenste deelvenster 2 revisies en kies dan **Context menu → Verklaar revisies**. Hiermee worden de verklaar-gegevens voor die 2 revisies opgehaald. Gebruik daarna dan de diff viewer om de twee verklaarde revisies te vergelijken.

##### Verklaar Wijzigingen

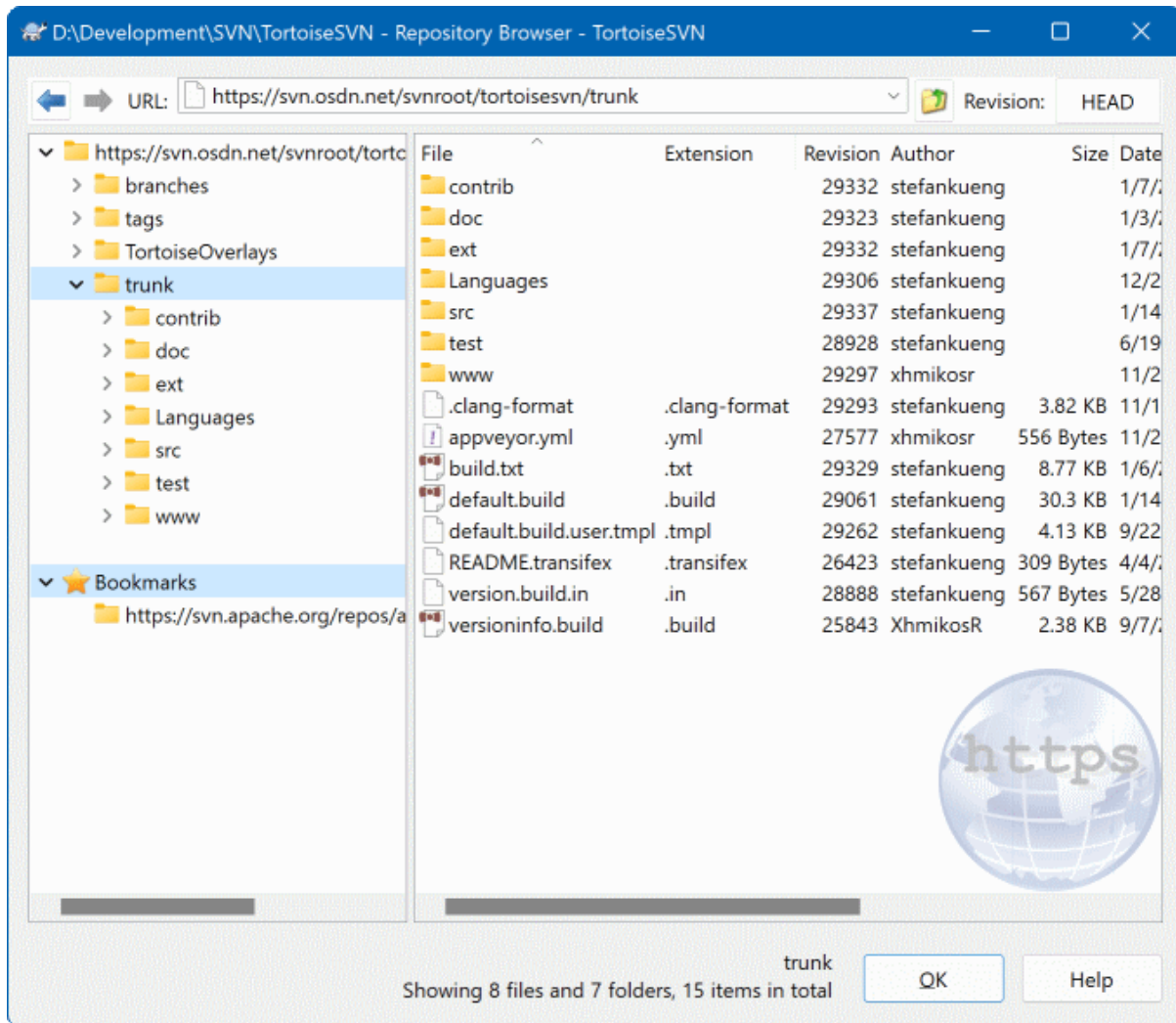
Selecteer één revisie in het bovenste deelvenster, kies vervolgens één bestand uit het onderste deelvenster en selecteer **Context menu → Verklaar wijzigingen**. Hiermee worden de verklaar-gegevens voor de geselecteerde revisie en de revisie daarvoor opgehaald. Gebruik daarna je diff programma om de twee verklaar-bestanden te vergelijken.

##### Vergelijk en Verklaar met Werkbasis

Toon het logboek voor een enkel bestand en selecteer in het bovenste deelvenster een enkele revisie. Kies vervolgens **Context menu → Vergelijk en verklaar met werkkopie (BASE)**. De verklaar-gegevens worden dan voor de geselecteerde revisie en het bestand in de werkbasis opgehaald. Gebruik dan het diff programma om de twee verklaar-bestanden te vergelijken.

### 4.25. De Archief-verkenner

Soms wil je direct dingen in het archief wijzigen, zonder dat je een werkkopie hebt. Daarvoor kun je de *Archief-verkenner* gebruiken. Met de archief-verkenner kun je de structuur en status van het archief bekijken, net zoals je met je verkenner en overlappende picotogrammen de status van je werkkopie kunt bekijken.



**Afbeelding 4.65. De Archief-verkenner**

Met de archiefverkenner kun je acties zoals kopiëren, verplaatsen, hernoemen, ... direct op het archief uitvoeren.

De archief-verkenner ziet er ongeveer hetzelfde uit als de windows verkenner, behalve dan dat het de inhoud van het archief van een bepaalde revisie laat zien, in plaats van bestanden op je computer. In het linker deelvenster wordt de mapstructuur weergegeven en in het rechter de inhoud van de geselecteerd map. Aan de bovenzijde van de archief-verkenner kun je de URL en de revisie van een archief invoeren, welke je wilt bekijken.

Folders included with the `svn:externals` property are also shown in the repository browser. Those folders are shown with a small arrow on them to indicate that they are not part of the repository structure, just links.

Net zoals bij de windows verkenner, kun je op de koptekst van een kolom klikken om de sorteervolgorde te wijzigen. Net zoals in de verkenner zijn er context menu's beschikbaar in beide deelvensters.

Vanuit het context menu kun je verschillende dingen doen met bestanden:

- Open het geselecteerde bestand met het standaard programma voor dit bestandstype of met een zelf gekozen programma.
- Edit the selected file. This will checkout a temporary working copy and start the default editor for that file type. When you close the editor program, if changes were saved then a commit dialog appears, allowing you to enter a comment and commit the change.

- Het revisie logboek laten tonen van een bestand of een figuur laten tonen met alle revisies, zodat je kunt zien waar het bestand vandaan is gekomen.
- Een bestand laten verklaren om te zien wie welke regel wanneer in een bestand gewijzigd heeft.
- Checkout a single file. This creates a “sparse” working copy which contains just this one file.
- Verwijderen of hernoemen van het bestand.
- Een kopie zonder versiebeheer van een bestand opslaan op je harde schijf.
- Copy the URL shown in the address bar to the clipboard.
- Een kopie laten maken van een bestand naar een ander deel van het archief of naar een werkkopie gesitueerd in hetzelfde archief.
- Bekijken/Wijzigen van de kenmerken van het bestand.
- Create a shortcut so that you can quickly start repo browser again, opened directly at this location.

Met het context menu van een map kun je dingen doen zoals:

- Het revisie logboek laten tonen van een map of een figuur laten tonen met alle revisies, zodat je kunt zien waar de map vandaan is gekomen.
- De map laten exporteren naar een locatie op je harde schijf zonder versiebeheer.
- Een map laten ophalen, zodat er een lokale werkkopie wordt gemaakt op je harde schijf.
- Een nieuwe map aanmaken in het archief.
- Add unversioned files or folders directly to the repository. This is effectively the Subversion Import operation.
- Een map verwijderen of hernoemen.
- Een kopie laten maken van een map naar een ander deel van het archief of naar een werkkopie gesitueerd in hetzelfde archief. Hiermee kan ook een vertakking/label gemaakt worden, zonder dat daarvoor een werkkopie nodig is.
- Eigenschappen van een map bekijken of wijzigen.
- Een map markeren voor vergelijken. Een gemarkeerd map wordt vetgedrukt weergegeven.
- De map vergelijken met een eerder gemarkeerde map, als unified diff of als een lijst met gewijzigde bestanden, die dan met het standaard vergelijkingsprogramma vergeleken kan worden. Dit kan in het bijzonder handig zijn als je twee labels wilt vergelijken, of een basislijn (trunk) met een vertakking.

Als je twee mappen in het rechter deelvenster selecteert, dan kun je de verschillen als unified-diff bekijken of als lijst van bestanden, welke dan weer met een vergelijkingsprogramma bekeken kan worden.

Als je meerdere mappen in het rechter deelvenster selecteert, dan kun je deze allemaal tegelijk ophalen naar een zelfde hoofdmap.

Als je 2 labels selecteert die van dezelfde hoofdmap gekopieerd zijn (normaal gesproken de basislijn / `trunk/`), dan kun je met Context Menu → Toon Logboek... de lijst met revisies zien tussen die twee labels.

External items (referenced using `svn:externals` are also shown in the repository browser, and you can even drill down into the folder contents. External items are marked with a red arrow over the item.

Je kunt zoals gebruikelijk met **F5** het beeld verversen. Hiermee wordt alles verversd wat weergegeven worden. Als je de informatie van andere delen, die je nog niet geopend hebt, wilt verversen of alvast wilt ophalen, gebruik dan **Ctrl-F5**. Daarna kun je elk ander deel direct openen, zonder dat daarvoor vertraging ontstaat voor het ophalen van informatie via het netwerk.

Je kunt de archief-verkenner ook gebruiken voor drag-and-drop acties. Als je een map vanuit de verkenner naar de archief-verkenner sleept, dan zal deze map in het archief geïmporteerd worden. Merk op dat als je meerdere objecten sleept, deze mappen als aparte vastleggingen worden behandeld.

Als je een object binnen het archief wilt verplaatsen, dan kun je deze gewoon met links verslepen naar de nieuwe locatie. Als je een kopie van het object wilt hebben in plaats van het alleen te verplaatsen, houdt dan ook **Ctrl** ingedrukt tijdens het verslepen. Als je een object gaat kopiëren, dan heeft de cursor een “plus” teken net zoals in de windows verkenner.

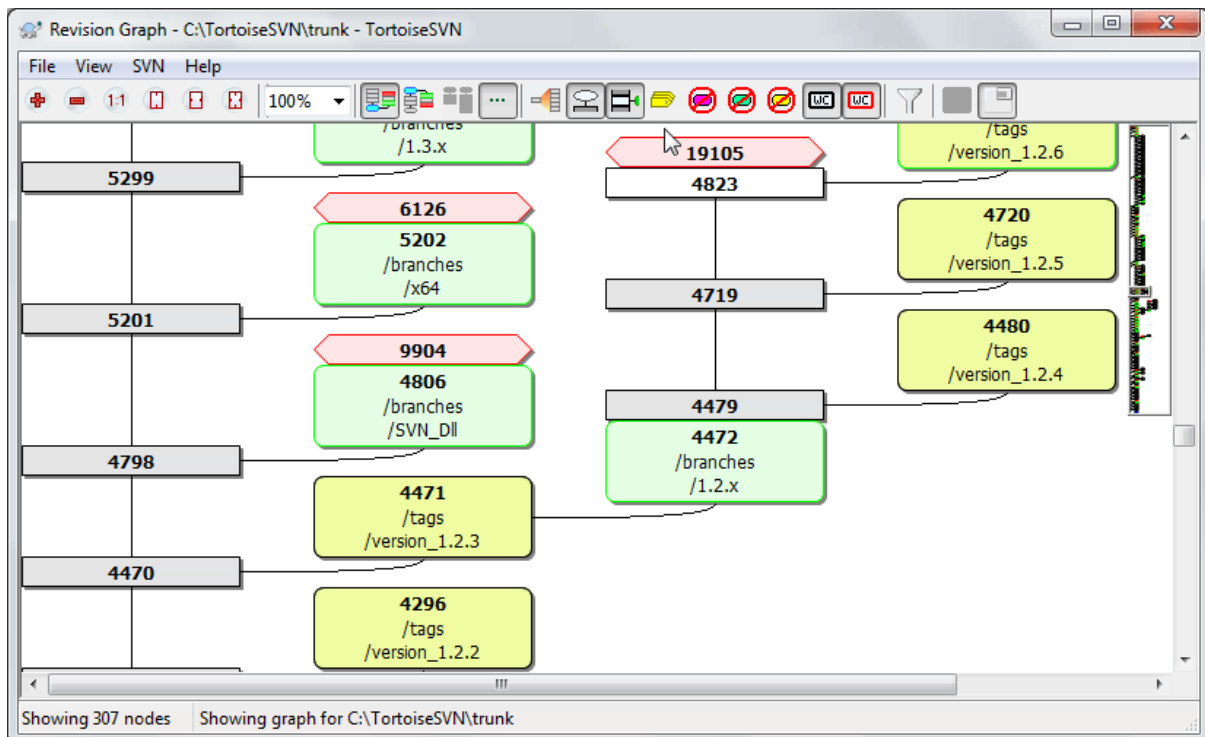
Als je een bestand of map wilt verplaatsen/kopiëren naar een andere locatie en ook tegelijkertijd de naam wilt wijzigen, dan moet je het object met rechts slepen of **Ctrl**-rechts slepen in plaats van links slepen. Er wordt dan een scherm getoond waarin je de nieuwe naam van de map of het bestand moet invoeren.

Telkens als je op deze manier wijzigingen in het archief doorvoert, zal er een logboek boodschappen venster getoond worden. Als je per ongeluk objecten hebt verslept, dan is dit je kans om die actie te stoppen.

Soms als je een pad wilt openen, krijg je een foutmelding in plaats van de objectdetails. Dit kan komen doordat je een verkeerde URL hebt ingevoerd of doordat je geen toegangsrechten hebt of omdat er een ander probleem met de server is. Als je de melding moet kopiëren naar bijvoorbeeld een email, klik dan met rechts en selecteer Context Menu → Kopieer foutmelding naar klembord of gebruikt **Ctrl+C**.

Bookmarked urls/repositories are shown below the current repository folders in the left tree view. You can add entries there by right clicking on any file or folder and select Context Menu → Add to Bookmarks. Clicking on a bookmark will browse to that repository and file/folder.

## 4.26. Revisie schema's



Afbeelding 4.66. Een Revisie Schema

Soms moet je weten op welk punt vertakkingen en labels gemaakt zijn van de basislijn (trunk). De ideale manier om dit te bekijken is door de informatie te sorteren in een grafiek of boomstructuur. Hiervoor kun je gebruik maken van TortoiseSVN → Revisie schema....

Dit commando analyseert de revisiegeschiedenis en probeert daarvan een boomstructuur te maken, waarin de punten aangegeven worden waarop kopieën gemaakt en vertakkingen/labels verwijderd zijn.



### Belangrijk

Om de figuur te maken, moet TortoiseSVN alle logboek boodschappen ophalen vanuit de hoofdmap van het archief. Overbodig is het om te zeggen dat dit enkele minuten kan duren, zelfs bij een archief met maar enkele duizenden revisie. Dit hangt natuurlijk sterk af van de snelheid van de server en de bandbreedte van het netwerk. Als je dit probeert te doen op bijvoorbeeld het *Apache* project met op dit moment meer dan 500.000 revisies, dan moet je wel een tijd wachten.

Het goede nieuws is dat als je log caching gebruikt, je maar een keer op deze vertraging hoeft te wachten. Daarna worden de logboek boodschappen lokaal opgeslagen. Log caching kun je inschakelen bij de instellingen van TortoiseSVN.

#### 4.26.1. Revisie Schema Knooppunten

Elke knoop in de revisie grafiek geeft een revisie in het archief weer, waarbij er iets gewijzigd is in de mapstructuur waar je naar kijkt. De verschillende soorten knopen zijn te onderscheiden door hun vorm en kleur. De vormen staan vast, maar de kleuren kunnen aangepast worden met TortoiseSVN → Instellingen

**Toegevoegde of gekopieerde objecten**

Items which have been added, or created by copying another file/folder are shown using a rounded rectangle. The default colour is green. Tags and trunks are treated as a special case and use a different shade, depending on the TortoiseSVN → Settings.

**Verwijderde objecten**

Deleted items e.g. a branch which is no longer required, are shown using an octagon (rectangle with corners cut off). The default colour is red.

**Hernoemde objecten**

Hernoemde objecten worden ook als een achthoek weergegeven, maar de standaard kleur daarvoor is blauw.

**Revisie van het uiteinde van een tak**

De grafiek laat normaal gesproken alleen de vertak-punten zien, maar soms is het ook handig om de HEAD revisie van een tak ook te zien. Als je optie **Toon HEAD revisies** selecteert, dan wordt elke HEAD revisie knoop getoond als ellips. Merk op dat hier met de HEAD revisie de laatste vastgelegde revisie van dat pad bedoeld wordt, niet de actuele HEAD revisie van het hele archief.

**Werkkopie revisie**

Als je de revisie grafiek van een werkkopie hebt opgevraagd, dan kun je de BASE (basis) revisie met **Toon revisie van werkkopie** laten weergeven. Hierdoor wordt de BASE knoop dikkere omtrek getoond.

**Gewijzigde werkkopie**

Als je de revisie grafiek van een werkkopie hebt opgevraagd, dan kun je additionele knopen, die jouw gewijzigde werkkopie representeren, laten zien met **Toon wijzigingen in werkkopie**. Dit wordt weergegeven als een ellipsvormige knoop met een dikkere omtrek en standaard een rode kleur.

**Normaal object**

Alle andere objecten worden als een normale rechthoek weergegeven.

Merk op dat standaard de grafiek alleen de punten laat zien waarop er objecten zijn toegevoegd, gekopieerd of verwijderd. Een grafiek laten maken van elke revisie van het project, levert een enorm grote grafiek op met niet



triviale gegevens. Als je echt *alle* wilt zien waarin wijzigingen zijn doorgevoerd, dan kun je dat doen instellen in het **Beeld** menu of via de knoppenbalk.

De standaard weergave (groeperen uitgeschakeld) positioneert de knopen zo dat deze verticaal op volgorde gerangschikt zijn, waardoor je direct een overzicht hebt van de volgorde waarop de dingen gedaan zijn. Als er twee knopen in dezelfde kolom staan, dan spreekt de volgorde voor zich. Als er twee knopen zijn in aangrenzende kolommen, dan is de afstand veel kleiner. Er is dan geen noodzaak om te voorkomen dat de knopen elkaar overlappen. Het gevolg hiervan is dat de volgorde iets minder overzichtelijk kan zijn. Deze optimalisaties zijn helaas nodig om complexe grafieken een acceptabel formaat te laten houden. Merk op dat bij het rangschikken de *rand* van de knoop aan de *oude* kant van de knoop gebruikt wordt als referentiepunt. Dat is de onderkant van de knoop als de oudste knoop onderaan de grafiek wordt weergegeven. Dit referentiepunt is belangrijk, omdat niet alle figuren van knopen dezelfde hoogte hebben.

#### 4.26.2. Wijzigen van de Weergave

Een revisie grafiek is vaak erg complex. Er zijn daarom verschillende functies beschikbaar, waarmee je de weergave kunt aanpassen naar jouw wensen. Deze zijn beschikbaar in het **Beeld** menu en via de werkbalk.

#### 4.26.3. Gebruik van de grafiek

Om het navigeren door een grote grafiek makkelijker te maken, kun je het overzicht scherm gebruiken. Hiermee wordt een klein scherm getoond, waarin de huidige weergave gemarkeerd is. Je kunt het gemarkeerde gebied verslepen om de actuele weergave te veranderen.

De revisie datum, auteur en het commentaar worden weergegeven in een hint box als je de muis boven een revisie houdt.

Als je twee revisies selecteert (gebruikt **Ctrl**-klik met links), dan kun je via het context menu de verschillen tussen deze revisies opvragen. Je kunt verschillen laten zien van het moment dat er vertakkingen zijn aangemaakt, maar meestal zul je de verschillen willen zien bij het eindpunt van een vertakking, bij de **HEAD** revisie.

Je kunt de verschillen in een Unified-Diff bestand bekijken, welke de exacte verschillen laat zien met minimale context. Als je voor **Context Menu** → **Vergelijk Revisies** kiest, zul je een lijst met wijzigingen krijgen. Dubbelklik op een bestandsnaam om beide revisies van dat bestand op te halen en te vergelijken met een visueel vergelijkingsprogramma.

If you right click on a revision you can use **Context Menu** → **Show Log** to view the history.

Je kunt ook de wijzigingen van de geselecteerde revisie(s) samenvoegen in een aparte werkkopie. Een scherm waarin je een map kunt selecteren, geeft de mogelijkheid om de werkkopie te selecteren waarin de selectie samengevoegd moet worden. Daarna is er geen bevestigingsscherm meer en geen kans meer om de samenvoeging te testen. Het is verstandig om de wijzigingen naar een niet gemodificeerde werkkopie samen te laten voegen, zodat je de wijzigingen ongedaan kunt maken als de samenvoeging niet oplevert wat je ervan had verwacht. Dit is handig als je functies van de geselecteerde revisie van de ene naar de andere tak wilt samenvoegen.



### Hoe moet je de Revisie Grafiek lezen

Mensen die de grafiek voor het eerst gebruiken, kunnen verrast worden doordat grafisch weergave heel anders kan zijn dan het model wat de mensen zelf voor ogen hebben. Als een revisie bijvoorbeeld meerdere kopieën, takken, een bestand of map veranderd, dan zullen er verschillen knopen zijn voor die revisie. Het is verstandig om dan te beginnen met de meest linkse opties in de werkbalk en de grafiek zo stap voor stap aan te passen, totdat het het dichtst bij het model komt dat je voor ogen had.

Alle filteropties proberen zo min mogelijk informatie verloren te laten gaan. Hierdoor kunnen de kleuren van sommige knopen bijvoorbeeld veranderen. Als een resultaat onverwacht is, maak dan de laatste filteractie ongedaan en probeer te achterhalen wat er speciaal is aan die revisie of tak. In de meeste gevallen is het initiële resultaat van de filteractie onnauwkeurig of misleidend.



#### 4.26.4. De weergave verversen

Als je wilt nagaan of er nieuwe informatie beschikbaar is op de server, dan kun je het beeld eenvoudig verversen met **F5**. Als je de log cache gebruikt (standaard ingeschakeld), dan zal het archief op nieuwe vastleggingen gecontroleerd worden en worden ook alleen de nieuwe vastleggingen opgehaald. Als de log cache in offline mode werkte, dan zal deze ook proberen weer naar online mode om te schakelen.

Als je de log cache gebruikt en je vermoedt dat de inhoud van een bericht of auteur gewijzigd is, gebruik dan het logboek venster om de berichten die je nodig hebt te verversen. Omdat de revisie grafiek werkt vanaf de basis van het archief, moeten we de hele log cache ongeldig verklaren. Het opnieuw vullen hiervan kan *erg* lang duren.

#### 4.26.5. Boomstructuren inkorten

Het navigeren door een grote boomstructuur kan lastig zijn. Soms wil je onderdelen verbergen of de structuur opdelen in een bos met kleinere bomen. Als je je muis boven het punt houdt waar een verbinding een knoop binnenkomt of verlaat, zul je een popup te zien krijgen met knoppen waarmee je dit kunt doen.



Klik op de min-knop om de bijbehorende onderverdeling in te klappen.



Klik op de plus-knop om de ingeklapte boom uit te klappen. Als een boom is ingeklapt blijft deze knop zichtbaar om de verborgen onderverdeling aan te geven .



Klik op de kruis-knop om de gekoppelde onderverdeling te splitsen en deze als een afzonderlijke boom op de grafiek te tonen.

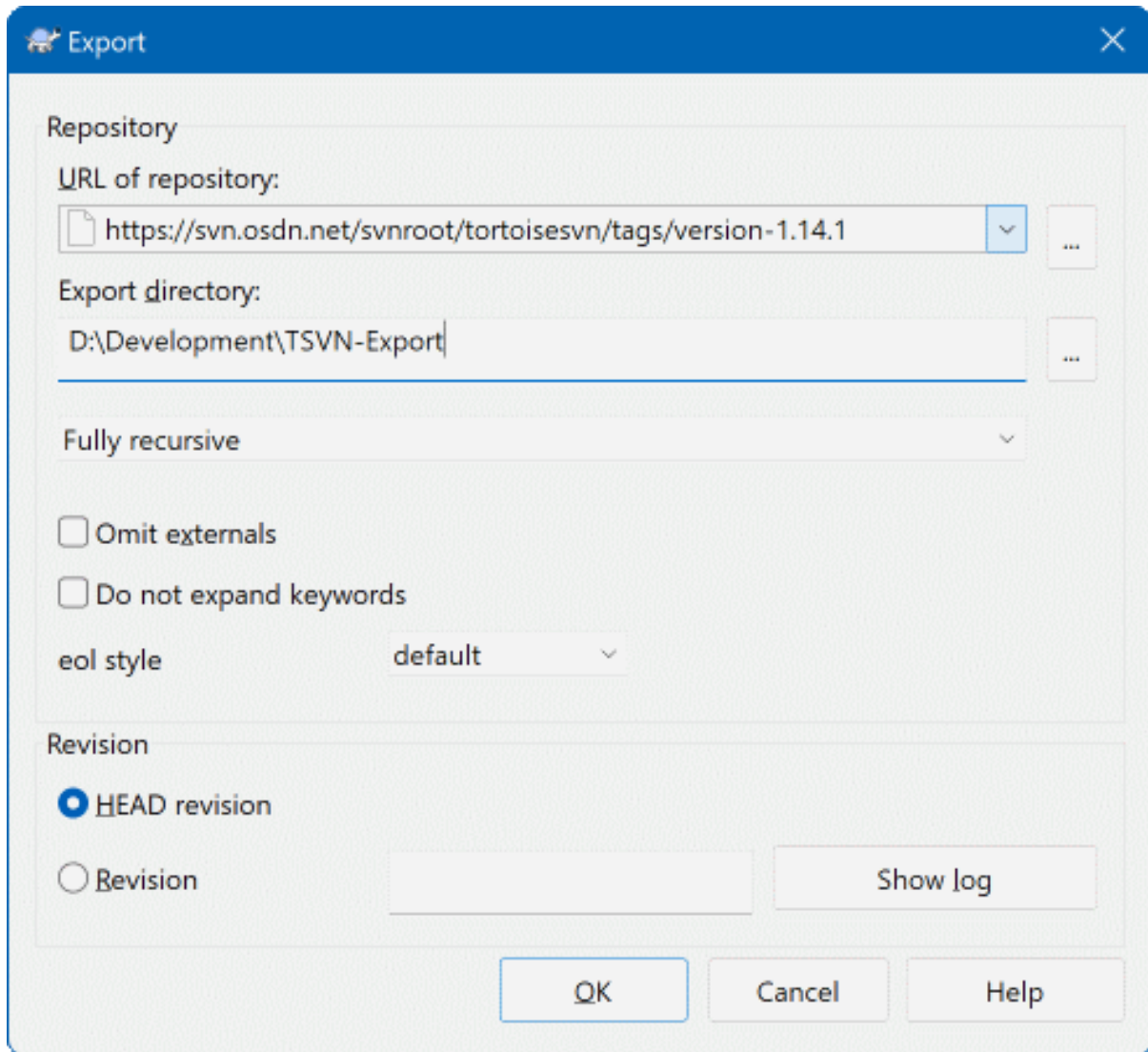


Klik op de cirkel-knop om een afgesplitste boom weer aan te koppelen. Wanneer een boom is afgesplitst blijft deze knop zichtbaar om aan te geven dat er een aparte onderverdeling aanwezig is.

Klik op de achtergrond van de grafiek voor het hoofd contextmenu, wat opties biedt naar Alles uitklappen en Alle samenvoegen. Als er geen tak is ingeklapt of afgesplitst zal het contextmenu niet worden getoond.

### 4.27. Exporteren van een Subversion Werkkopie

Sometimes you may want a clean copy of your working tree without the `.svn` directory, e.g. to create a zipped tarball of your source, or to export to a web server. Instead of making a copy and then deleting the `.svn` directory manually, TortoiseSVN offers the command TortoiseSVN → Export.... Exporting from a URL and exporting from a working copy are treated slightly differently.



**Afbeelding 4.67. Het Exporteren-vanaf-een-URL scherm**

Als je dit commando op een map zonder versiebeheer toepast, dan zal TortoiseSVN deze map als de doel map zien en een scherm openen waarin je de URL en revisie in kunt voeren van waar er geëxporteerd moet worden. Er zijn opties beschikbaar waarmee kunt selecteren dat je alleen de hoofdmap wilt hebben, externen wilt uitsluiten en de stijl voor geregeleinden voor bestanden, die de eigenschap `svn:eol-style` hebben, kunt negeren.

Natuurlijk kun je ook direct vanuit het archief exporteren. Gebruik de archief-verkenner om naar de relevante submappen te gaan in je archief en gebruik dan **Context Menu** → **Exporteren**. Je krijgt dan het **Exporteren van URL** venster zoals eerder beschreven.

If you execute this command on your working copy you'll be asked for a place to save the *clean* working copy without the `.svn` folder. By default, only the versioned files are exported, but you can use the **Export unversioned files too** checkbox to include any other unversioned files which exist in your WC and not in the repository. External references using `svn:externals` can be omitted if required.

Another way to export from a working copy is to right drag the working copy folder to another location and choose **Context Menu** → **SVN Export versioned items here** or **Context Menu** → **SVN Export all items here** or **Context Menu** → **SVN Export changed items here**. The second option includes the unversioned files as well. The third option exports only modified items, but maintains the folder structure.

When exporting from a working copy, if the target folder already contains a folder of the same name as the one you are exporting, you will be given the option to overwrite the existing content, or to create a new folder with an automatically generated name, e.g. Target (1).



## Exporteren van enkele bestanden

Het export scherm staat het exporteren van een enkel bestand niet toe, ondanks dat Subversion dat wel ondersteunt.

Voor het exporteren van enkele bestanden met TortoiseSVN, moet je de archief-verkenner gebruiken ([Paragraaf 4.25, “De Archief-verkenner”](#)). Sleep het bestand (of de bestanden) die je wilt exporteren, van de archief-verkenner naar de plaats in de verkenner waar je deze wilt hebben. Je kunt ook bestanden exporteren vanuit het context menu in de archief-verkenner.



## Exporteren van een gewijzigde boom

Als je een kopie van je project boomstructuur wilt exporteren met alleen de in een specifieke revisie (of tussen 2 revisies) gewijzigde bestanden, gebruik dan de revisies vergelijken functionaliteit best [Paragraaf 4.11.3, “Vergelijken van mappen”](#).

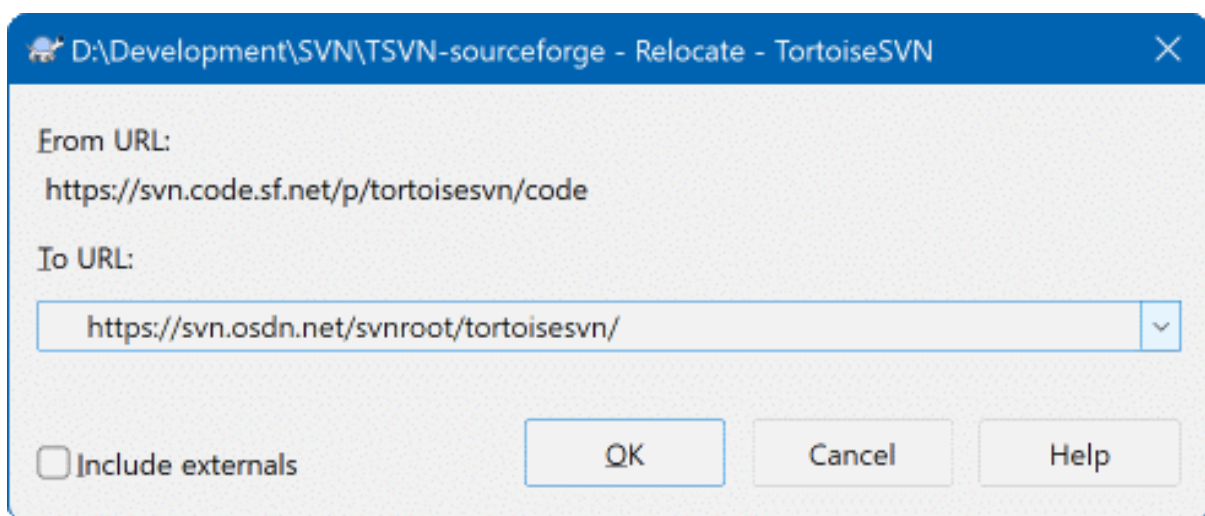
If you want to export your working copy tree structure but containing only the files which are locally modified, refer to SVN Export changed items here above.

### 4.27.1. Versiebeheer van een werkkopie verwijderen

Sometimes you have a working copy which you want to convert back to a normal folder without the `.svn` directory. All you need to do is delete the `.svn` directory from the working copy root.

Alternatively you can export the folder to itself. In Windows Explorer right drag the working copy root folder from the file pane onto itself in the folder pane. TortoiseSVN detects this special case and asks if you want to make the working copy unversioned. If you answer *yes* the control directory will be removed and you will have a plain, unversioned directory tree.

## 4.28. Verhuizen van een werkkopie



### Afbeelding 4.68. De Verhuis Dialog

If your repository has for some reason changed it's location (IP/URL). Maybe you're even stuck and can't commit and you don't want to checkout your working copy again from the new location and to move all your changed

data back into the new working copy, TortoiseSVN → Relocate is the command you are looking for. It basically does very little: it rewrites all URLs that are associated with each file and folder with the new URL.

### Opmerking

This operation only works on working copy *roots*. So the context menu entry is only shown for working copy roots.

Misschien wordt je verrast als TortoiseSVN het archief benaderd gedurende deze actie. Het enige wat het doet, is controleren of de nieuwe URL daadwerkelijk naar het archief wijst van de bestaande werkkopie.



### Waarschuwing

*Dit is een zelden gebruikte methode.* Het verhuizen commando wordt *alleen* gebruikt als de URL van de basis van het archief is gewijzigd. Mogelijke redenen hiervan zijn:

- Het IP-adres van de server is gewijzigd.
- Het protocol is gewijzigd (vb: http:// naar https://).
- Het pad van de hoofdmap van het archief is veranderd in de server instellingen.

Anders gezegd, je moet verhuizen als je werkkopie refereert naar dezelfde locatie in hetzelfde archief, maar het archief zelf is verhuisd.

Het is niet van toepassing als:

- Je wilt verhuizen naar een ander Subversion archief. In zo'n geval moet je een verse werkkopie ophalen vanaf de nieuwe archief locatie.
- Je wilt schakelen naar een andere tak of map binnen hetzelfde archief. Om dat te doen moet je gebruik maken van TortoiseSVN → Wisselen.... Lees [Paragraaf 4.20.3, "Ophalen of Wisselen..."](#) voor meer informatie.

Als je verhuizen gebruikt bij één van de twee hierboven genoemde situaties, dan *zal je werkkopie beschadigd worden* en krijg je mogelijk onverklaarbare foutmeldingen bij het verversen, vastleggen, etc. Als dat gebeurt is, kun je dat alleen oplossen met een vers opgehaalde werkkopie.

## 4.29. Integratie met Bug Tracking Systemen / Issue Trackers

Het is gemeengoed in Software Ontwikkeling om wijzigingen te relateren aan een specifieke bug of issue ID. Gebruikers van bug tracking systemen (issue trackers) zullen de wijzigingen welke zij maken in Subversion willen associëren aan een specifiek ID in hun issue tracker. Het merendeel van de issue trackers levert hiervoor een pre-commit hook script welke het logbericht analyseert om het bug ID te vinden waarmee de vastlegging wordt geassocieerd. Dit is enigzins foutgevoelig omdat dit afhankelijk is van de correctheid van het door de gebruiker geschreven logbericht zodat het pre-commit hook script het correct kan analyseren.

TortoiseSVN kan de gebruiker op twee manieren helpen:

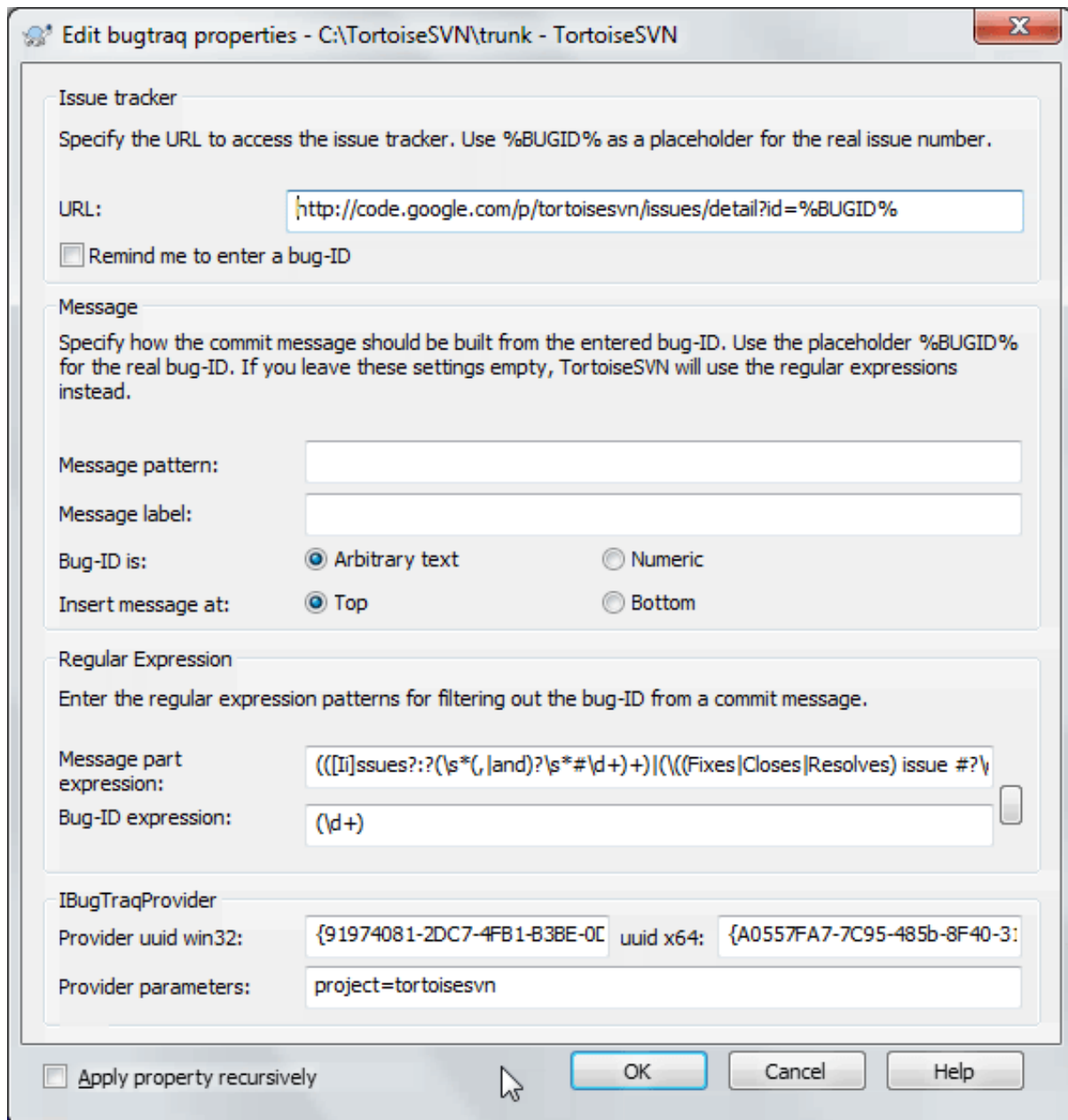
1. Wanneer de gebruiker een logbericht ingeeft kan automatisch een vooraf gedefiniëerde regel, met daarin het met de vastlegging geassocieerde issue-nummer, worden toegevoegd. Dit verkleint het risico dat de gebruiker het issue-nummer op een dusdanige wijze ingeeft dat dit niet correct door de bug tracking tools kan worden geanalyseerd.

Of TortoiseSVN kan het gedeelte van het ingegeven logbericht, wat wordt herkend door de issue tracker, accentueren. Op die manier is het voor de gebruiker duidelijk of het logbericht correct kan worden geanalyseerd.

2. Wanneer de gebruiker door de logberichten bladert, maakt TortoiseSVN van elke bug ID in het logbericht een verwijzing waarmee de genoemde issue in een browser kan worden geopend.

#### 4.29.1. Issue-nummers aan Logberichten toevoegen

U kunt een bug tracking tool naar keuze in TortoiseSVN integreren. Hiervoor dient een aantal eigenschappen te worden gedefinieerd, welke starten met `bugtraq:`. Ze dienen te worden ingesteld op Mappen: (Paragraaf 4.18, “Project Instellingen”)



**Afbeelding 4.69. The Bugtraq Properties Dialog**

When you edit any of the bugtraq properties a special property editor is used to make it easier to set appropriate values.

Er zijn twee manieren om TortoiseSVN te integreren met issue trackers. De een is gebaseerd op eenvoudige tekenreeksen, de ander is gebaseerd op *reguliere expressies*. De eigenschappen welke door beide benaderingen worden gebruikt zijn:

**bugtraq:url**

Set this property to the URL of your bug tracking tool. It must be properly URI encoded and it has to contain %BUGID%. %BUGID% is replaced with the Issue number you entered. This allows TortoiseSVN to display a link in the log dialog, so when you are looking at the revision log you can jump directly to your bug tracking tool. You do not have to provide this property, but then TortoiseSVN shows only the issue number and not the link to it. e.g the TortoiseSVN project is using `http://issues.tortoisesvn.net/?do=details&id=%BUGID%`.

U kunt ook gebruik maken van relatieve URL's in plaats van absolute. Dit kan handig zijn in het geval dat uw issue tracker zich op hetzelfde domein/server bevindt als uw bron archief. In het geval dat de domeinnaam ooit wijzigt hoeft u de `bugtraq:url` niet aan te passen. Er zijn twee manieren om een relatieve URL aan te geven:

If it begins with the string `^/` it is assumed to be relative to the repository root. For example, `^/../?do=details&id=%BUGID%` will resolve to `https://tortoisesvn.net/?do=details&id=%BUGID%` if your repository is located on `https://tortoisesvn.net/svn/trunk/`.

A URL beginning with the string `/` is assumed to be relative to the server's hostname. For example `/?do=details&id=%BUGID%` will resolve to `https://tortoisesvn.net/?do=details&id=%BUGID%` if your repository is located anywhere on `https://tortoisesvn.net`.

**bugtraq:warnifnoissue**

Stel dit in op `true` indien u wilt dat TortoiseSVN u waarschuwt in het geval van een leeg issue-nummer tekst veld. Geldige waarden zijn `true/false`. *Indien niet gedefiniëerd, wordt aangenomen dat de waarde `false` is.*

#### 4.29.1.1. Issue-nummer in Tekst Veld

In de eenvoudige benadering toont TortoiseSVN een apart invoerveld aan de gebruiker, waar een bug ID kan worden ingegeven. Daarna wordt een aparte regel toegevoegd/vooringevoegd aan het door de gebruiker ingegeven logbericht.

**bugtraq:message**

Deze eigenschap activeert het bug tracking systeem in *Invoerveld* modus. Als deze eigenschap is ingesteld zal TortoiseSVN u vragen om een issue-nummer in te geven wanneer u uw wijzigingen vastlegt. Dit wordt gebruikt om een regel aan het einde van het logbericht toe te voegen. Het moet %BUGID% bevatten, wat wordt vervangen door het issue-nummer bij vastleggen. Dit biedt de zekerheid dat uw vastleggingslog een referentie naar het issue-nummer bevat, welke altijd in een consistent formaat is en kan worden geanalyseerd door uw bug tracking tool, om het issue-nummer met een bepaalde vastlegging te associëren. Als voorbeeld zou u `Issue : %BUGID%` kunnen gebruiken, maar dat hangt van uw tool af.

**bugtraq:label**

Deze tekst wordt door TortoiseSVN getoond op het vastleggings venster om het invoerveld aan te geven waar u het issue-nummer ingeeft. Wanneer het niet is ingesteld zal `Bug-ID / Issue-Nr :` worden getoond. Houdt in gedachten dat het venster niet in grootte wordt aangepast om het label te laten passen, dus houdt de grootte van het label onder de 20-25 tekens.

**bugtraq:number**

Indien ingesteld op `true` zijn alleen nummers in het issue-nummer tekstveld toegestaan. Een uitzondering is de komma, zodat meerdere nummers door een komma kunnen worden gescheiden. Geldige waarden zijn `true/false`. *Indien niet gedefiniëerd wordt aangenomen dat de waarde `true` is.*

**bugtraq:append**

Deze eigenschap definiëert of het bug-ID wordt toegevoegd (`true`) aan het einde van het logbericht of ingevoegd (`false`) aan het begin van het logbericht. Geldige waarden zijn `true/false`. *Indien niet gedefiniëerd wordt aangenomen dat de waarde `true` is, zodat bestaande projecten niet mislukken.*

#### 4.29.1.2. Issue-nummers bij gebruik van Reguliere Expressies

In de benadering met *reguliere expressies* toont TortoiseSVN geen apart invoerveld, maar wordt het gedeelte van de door de gebruiker ingegeven logbericht aangemerkt wat door de issue tracker wordt herkend. Dit wordt

gedaan terwijl de gebruiker het logbericht ingeeft. Dit betekent ook dat het bug ID zich overal in het logbericht mag bevinden! Deze methode is veel flexibeler en is de methode welke door het TortoiseSVN project zelf wordt gebruikt.

#### bugtraq:logregex

Deze eigenschap activeert het bug tracking systeem in *Regex* modus. Het bevat of een enkele reguliere expressie, of twee door een nieuwe regel gescheiden reguliere expressies.

If two expressions are set, then the first expression is used as a pre-filter to find expressions which contain bug IDs. The second expression then extracts the bare bug IDs from the result of the first regex. This allows you to use a list of bug IDs and natural language expressions if you wish. e.g. you might fix several bugs and include a string something like this: "This change resolves issues #23, #24 and #25".

If you want to catch bug IDs as used in the expression above inside a log message, you could use the following regex strings, which are the ones used by the TortoiseSVN project: `[ Ii ]ssues? : ? ( \s* ( , | and ) ? \s* # \d+ ) + and ( \d+ )`.

De eerste expressie haalt "issues #23, #24 and #25" uit het omringende logbericht. De tweede regex extraheert zuivere decimale nummers uit de uitvoer van de eerste regex, zodat het "23", "24" en "25" teruggeeft om te gebruiken als bug ID's.

Wanneer we de eerste regex in stukken opbreken, dan moet deze beginnen met het woord "issue", mogelijk in hoofdletters. Dit wordt eventueel gevolgd door een "s" (meerdere issues) en eventueel een dubbele punt. Dit wordt gevolgd door een of meerdere groepen welke elk worden voorafgegaan door 0 of meer witruimte tekens, een komma of "en" en meer optionele spaties. Als laatste is er een verplichte "#" en een verplicht decimaal nummer.

Indien slechts één expressie is ingesteld, moeten de kale bug ID's worden vergeleken met de groepen van de regex tekenreeks. Voorbeeld: `[ Ii ]ssue ( ? : s ) ? # ? ( \d+ )` Deze methode is vereist door een aantal issue trackers, bijv. trac, maar het is moeilijker om de regex samen te stellen. We adviseren deze methode alleen te gebruiken wanneer de documentatie van de issue tracker dit voorschrijft.

If you are unfamiliar with regular expressions, take a look at the introduction at [https://en.wikipedia.org/wiki/Regular\\_expression](https://en.wikipedia.org/wiki/Regular_expression), and the online documentation and tutorial at <http://www.regular-expressions.info/>.

It's not always easy to get the regex right so to help out there is a test dialog built into the bugtraq properties dialog. Click on the button to the right of the edit boxes to bring it up. Here you can enter some test text, and change each regex to see the results. If the regex is invalid the edit box background changes to red.

Als beide eigenschappen `bugtraq:message` en `bugtraq:logregex` zijn ingesteld, krijgt `logregex` voorrang.



### Tip

Zelfs als u geen issue tracker heeft welke uw logberichten analyseert met een pre-commit hook, kunt u dit gebruiken om de in uw logbericht genoemde issues om te zetten naar verwijzingen!

En zelfs als u de verwijzingen niet nodig heeft, worden de issue-nummers getoond als een aparte kolom in het log venster wat het gemakkelijker maakt om de wijzigingen te vinden welke aan een bepaalde issue zijn gerelateerd.

Sommige `tsvn:` eigenschappen vereisen een `true/false` waarde. TortoiseSVN begrijpt ook `yes` als een synoniem voor `true` en `no` als een synoniem voor `false`.



### Stel de Eigenschappen in op Mappen

These properties must be set on folders for the system to work. When you commit a file or folder the properties are read from that folder. If the properties are not found there, TortoiseSVN will search



upwards through the folder tree to find them until it comes to an unversioned folder, or the tree root (e.g. `C:\`) is found. If you can be sure that each user checks out only from e.g. `trunk/` and not some sub-folder, then it's enough if you set the properties on `trunk/`. If you can't be sure, you should set the properties recursively on each sub-folder. A property setting deeper in the project hierarchy overrides settings on higher levels (closer to `trunk/`).

As of version 1.8, TortoiseSVN and Subversion use so called *inherited properties*, which means a property that is set on a folder is automatically also implicitly set on all subfolders. So there's no need to set the properties on all folders anymore but only on the root folder.

Bij eigenschappen welke *alleen* op projecten van toepassing zijn, bijv. `tsvn:`, `bugtraq:` en `webviewer:`, kunt u gebruik maken van de **Recursief vinkbox** om de eigenschap op alle sub-mappen binnen de hiërarchie in te stellen, zonder het ook op de bestanden in te stellen.

Als u met TortoiseSVN nieuwe sub-mappen aan een werkkopie toevoegt, zullen alle project eigenschappen welke in de bovenliggende map aanwezig zijn, automatisch aan de nieuwe sub-map worden toegevoegd.



### Geen Issue Tracker Informatie vanuit de Archief Verkenner

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 unless you started the repo browser from your working copy. If you started the repo browser by entering the URL of the repository you won't see this feature.

Om de zelfde reden zullen de project eigenschappen niet automatisch worden gepropageerd wanneer een sub-map wordt toegevoegd vanuit de archief-verkenner.

This issue tracker integration is not restricted to TortoiseSVN; it can be used with any Subversion client. For more information, read the full *Issue Tracker Integration Specification* [<https://svn.code.sf.net/p/tortoisesvn/code/trunk/doc/notes/issuetrackers.txt>] in the TortoiseSVN source repository. (**Paragraaf 3, "Licentie"** explains how to access the repository.)

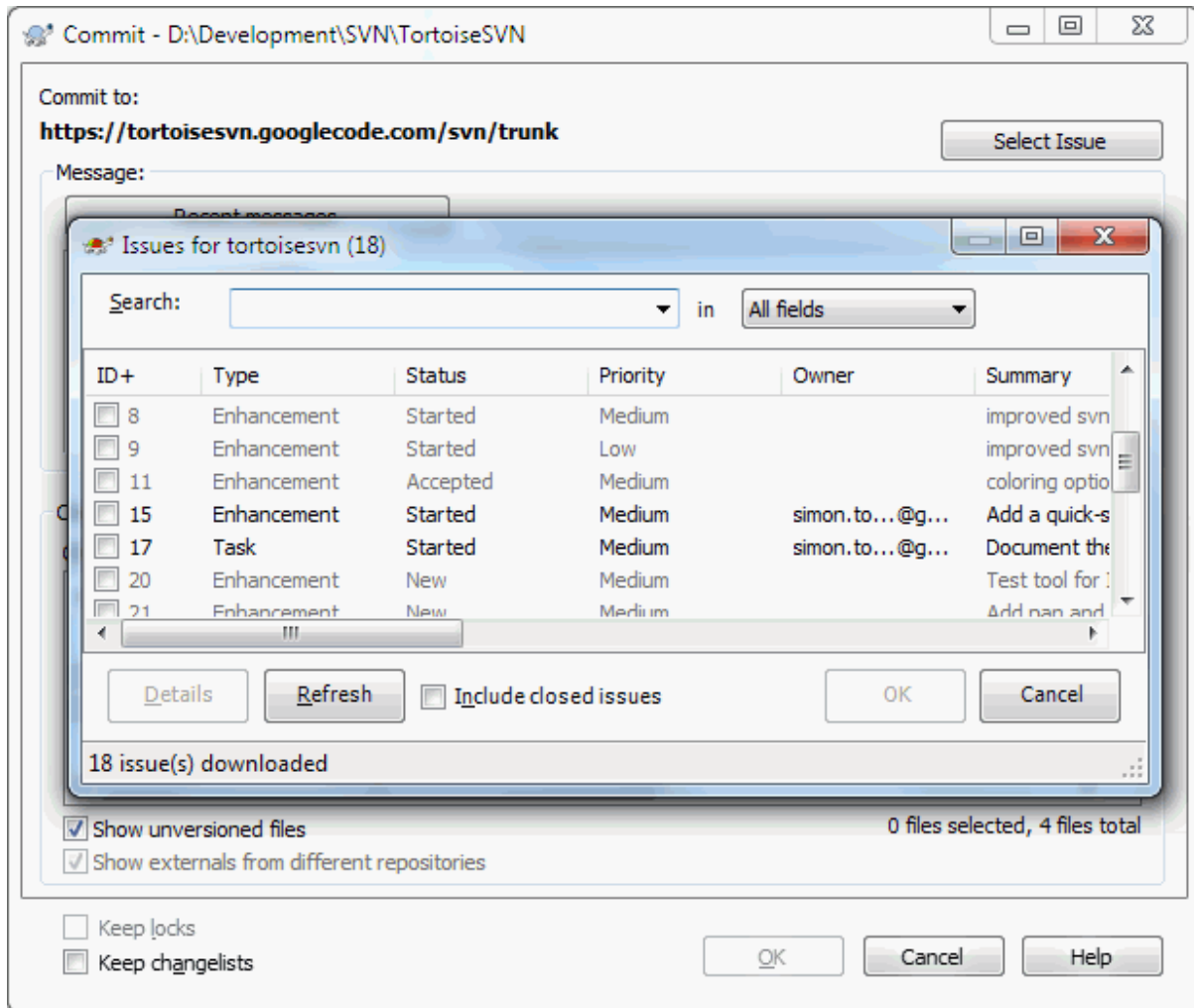
## 4.29.2. Informatie Verkrijgen vanuit de Issue Tracker

De vorige sectie betrof het toevoegen van issue informatie aan de logberichten. Maar wat als u de informatie wilt verkrijgen uit de issue tracker? Het vastleg scherm bezit een COM interface welke de mogelijkheid biedt een extern programma, wat met uw tracker kan praten, te integreren. Gewoonlijk wilt u de tracker een lijst vragen met openstaande issues op uw naam, zodat u die issues eruit kunt pikken welke betrekking hebben op deze vastlegging.

Any such interface is of course highly specific to your issue tracker system, so we cannot provide this part, and describing how to create such a program is beyond the scope of this manual. The interface definition and sample plugins in C# and C++/ATL can be obtained from the `contrib` folder in the *TortoiseSVN repository* [<https://svn.code.sf.net/p/tortoisesvn/code/trunk/contrib/issue-tracker-plugins>]. (**Paragraaf 3, "Licentie"** explains how to access the repository.) A summary of the API is also given in **Hoofdstuk 7, *IBugtraqProvider interface***. Another (working) example plugin in C# is *Gurtle* [<http://code.google.com/p/gurtle/>] which implements the required COM interface to interact with the *Google Code* [<http://code.google.com/hosting/>] issue tracker.

Als voorbeeld: Aangenomen dat uw systeemadministrator u een issue tracker plugin heeft aangeleverd welke u heeft geïnstalleerd, en dat u enkele van uw werkkopieën zo heeft ingesteld, met behulp van het TortoiseSVN instellingen scherm, dat ze gebruik maken van de plugin. Wanneer u het vastleggen scherm opent vanuit de werkkopie waaraan de plugin is toegekend, zult u een nieuwe knop zien aan de bovenkant van het venster.





Afbeelding 4.70. Voorbeeld issue tracker zoek scherm

In dit voorbeeld kunt u één of meerdere issues selecteren. De plugin kan vervolgens een speciaal opgemaakte tekst genereren welke deze toevoegt aan uw logbericht.

## 4.30. Integratie met Web-gebaseerde Archief Viewers

There are several web-based repository viewers available for use with Subversion such as [ViewVC](http://www.viewvc.org/) [http://www.viewvc.org/] and [WebSVN](https://websvnphp.github.io/) [https://websvnphp.github.io/]. TortoiseSVN provides a means to link with these viewers.

U kunt een archief viewer naar keuze integreren in TortoiseSVN. Hiervoor dient u een aantal eigenschappen te definiëren welke de koppeling definiëren. Deze dienen op Mappen te worden ingesteld: (Paragraaf 4.18, "Project Instellingen")

webviewer:revision

Set this property to the URL of your repo viewer to view all changes in a specific revision. It must be properly URI encoded and it has to contain %REVISION%. %REVISION% is replaced with the revision number in question. This allows TortoiseSVN to display a context menu entry in the log dialog Context Menu → View revision in webviewer.

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 for 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`.

U kunt ook gebruik maken van relatieve URL's in plaats van absolute. Dit komt van pas in het geval dat uw webviewer zich op het zelfde domein/server bevindt als uw bronarchief. In het geval dat de domeinnaam ooit wijzigt hoeft u de `webviewer:revision` en `webviewer:pathrevision` eigenschappen niet aan te passen. De schrijfwijze is dezelfde als van de `bugtraq:url` eigenschap. Zie ook [Paragraaf 4.29, "Integratie met Bug Tracking Systemen / Issue Trackers"](#).



## Stel de Eigenschappen in op Mappen

These properties must be set on folders for the system to work. When you commit a file or folder the properties are read from that folder. If the properties are not found there, TortoiseSVN will search upwards through the folder tree to find them until it comes to an unversioned folder, or the tree root (e.g. `C:\`) is found. If you can be sure that each user checks out only from e.g. `trunk/` and not some sub-folder, then it's enough if you set the properties on `trunk/`. If you can't be sure, you should set the properties recursively on each sub-folder. A property setting deeper in the project hierarchy overrides settings on higher levels (closer to `trunk/`).

Bij eigenschappen welke *alleen* op projecten van toepassing zijn, bijv. `tsvn:`, `bugtraq:` en `webviewer:`, kunt u gebruik maken van de **Recursief** vinkbox om de eigenschap op alle sub-mappen binnen de hiërarchie in te stellen, zonder het ook op de bestanden in te stellen.

Als u met TortoiseSVN nieuwe sub-mappen aan een werkkopie toevoegd, zullen alle project eigenschappen welke in de bovenliggende map aanwezig zijn, automatisch aan de nieuwe sub-map worden toegevoegd.



## Beperkingen Van De Archief-verkenner

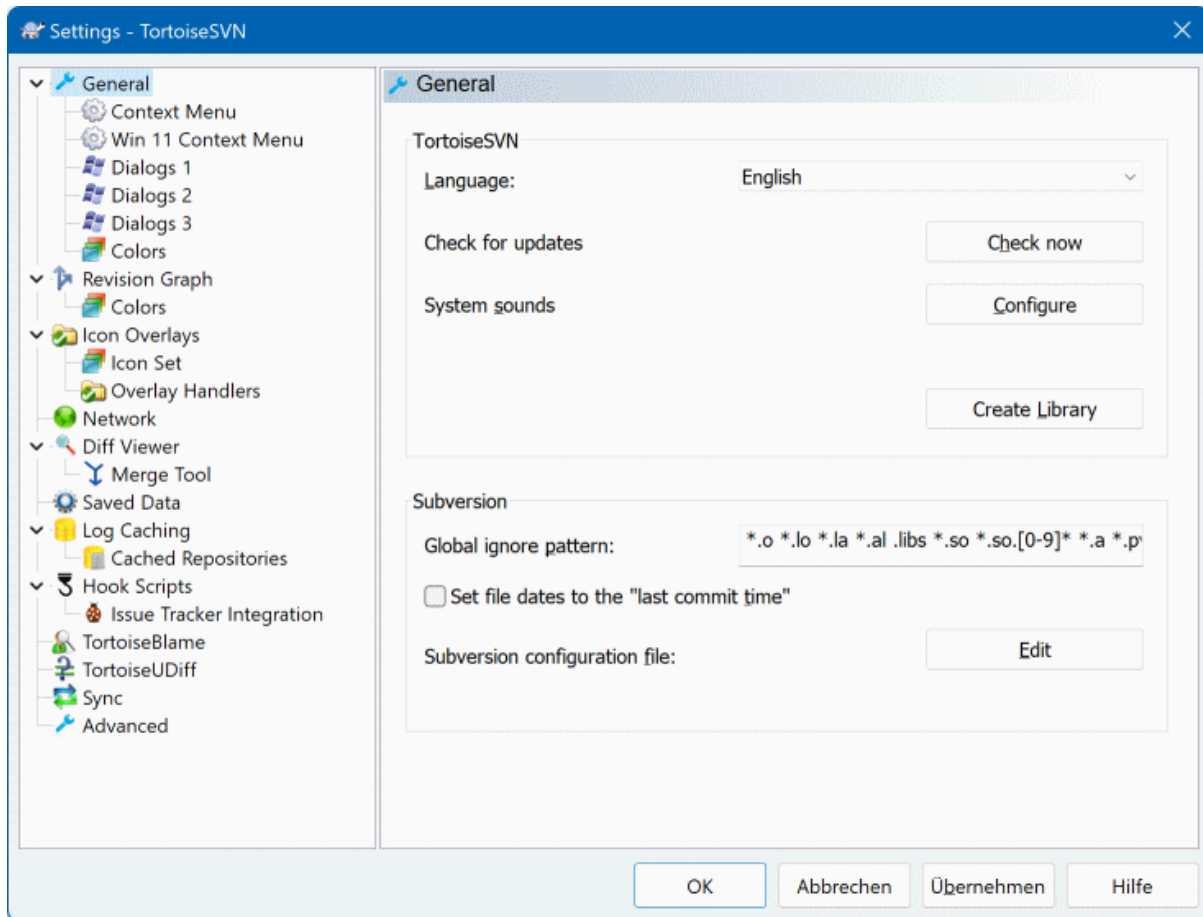
Because the repo viewer integration depends upon accessing Subversion properties, you will only see the results when using a checked out working copy. Fetching properties remotely is a slow operation, so you will not see this feature in action from the repo browser unless you started the repo browser from your working copy. If you started the repo browser by entering the URL of the repository you won't see this feature.

Om de zelfde reden zullen de project eigenschappen niet automatisch worden gepropageerd wanneer een sub-map wordt toegevoegd vanuit de archief-verkenner.

## 4.31. TortoiseSVN Instellingen

Om uit te vinden waar de diverse instellingen voor dienen kun je met de muisaanwijzer naar het invoerveld wijzen....en dan verschijnt er een handige "tooltip".

### 4.31.1. Algemene instellingen



### Afbeelding 4.71. Het Instellingen Scherm, Algemene Pagina

In dit scherm kan de voorkeurstaal worden ingesteld en specifieke Subversion instellingen.

#### Taal

Selects your user interface language. Of course, you have to install the corresponding language pack first to get another UI language than the default English one.

#### Controleer op updates

TortoiseSVN will contact its download site periodically to see if there is a newer version of the program available. If there is it will show a notification link in the commit dialog. 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.

#### Systeemgeluiden

TortoiseSVN heeft standaard drie eigen geluidseffecten geïnstalleerd.

- Fout
- Opmerking
- Waarschuwing

Om andere geluidseffecten te selecteren (of het geluid helemaal uit te zetten) kan gebruik gemaakt worden van het standaard Windows Configuratiescherm (Control Panel). Instellen is een koppeling naar dit Configuratiescherm.

#### Use Aero Dialogs

On Windows Vista and later systems this controls whether dialogs use the Aero styling.

### Bibliotheek aanmaken

On Windows 7 you can create a Library in which to group working copies which are scattered in various places on your system.

### Globaal neger patroon

Patroonherkenning kan worden gebruikt om ervoor te zorgen, dat bestanden die niet onder versiebeheer vallen ook niet zullen worden getoond bij (bijvoorbeeld) het uitvoeren van een vastlegging. Deze bestanden worden dan ook genegeerd bij het importeren. Bestanden of mappen kunnen worden genegeerd door de namen of extensies in te voeren. Deze patronen moeten worden gescheiden door spaties, bijvoorbeeld `bin obj *.bak .~?? *.jar *.[Tt]mp`. In deze patronen mogen géén scheidingstekens voorkomen die ook in bestandspaden worden gebruikt. Houd er rekening mee, dat er geen onderscheid gemaakt kan worden tussen bestanden en mappen. Zie [Paragraaf 4.14.1, “Patroonherkenning in negerlijsten”](#) voor meer informatie over de syntax van deze patronen.

Houd er rekening mee, dat de patronen die je opgeeft ook effect zullen hebben op andere Subversion clients op dezelfde PC, inclusief de command-line client.



### Let op

Wanneer je in de Subversion configuratiefile een globaal neger-patroon instelt, dan heeft die instelling voorrang op waarden die je hier invoert. Het Subversion configuratiebestand kun je openen met behulp van de Wijzigen hieronder.

This ignore pattern will affect all your projects. It is not versioned, so it will not affect other users. By contrast you can also use the versioned `svn:ignore` or `svn:global-ignores` property to exclude files or directories from version control. Read [Paragraaf 4.14, “Negeren bestanden en mappen”](#) for more information.

### Bestandsdatum overschrijven met de “datum van laatste vastlegging”

Deze optie zorgt ervoor, dat TortoiseSVN de bestandsdatum van het lokale bestand aanpast aan de laatste datum waarop dit bestand is bijgewerkt (“last commit time”) in Subversion. Standaard zal TortoiseSVN hiervoor de huidige datum gebruiken. Wanneer je software ontwikkelt is het meestal de beste keuze om de huidige datum te laten gebruiken, want ontwikkel-tools maken vaak gebruik van de bestandsdatum om te beoordelen of het bestand weer opnieuw gecompileerd moet worden. Wanneer je dan de “last commit time” heb gekozen en dan een oudere revisie terugzet, dan kan dit tot een onverwacht resultaat leiden.

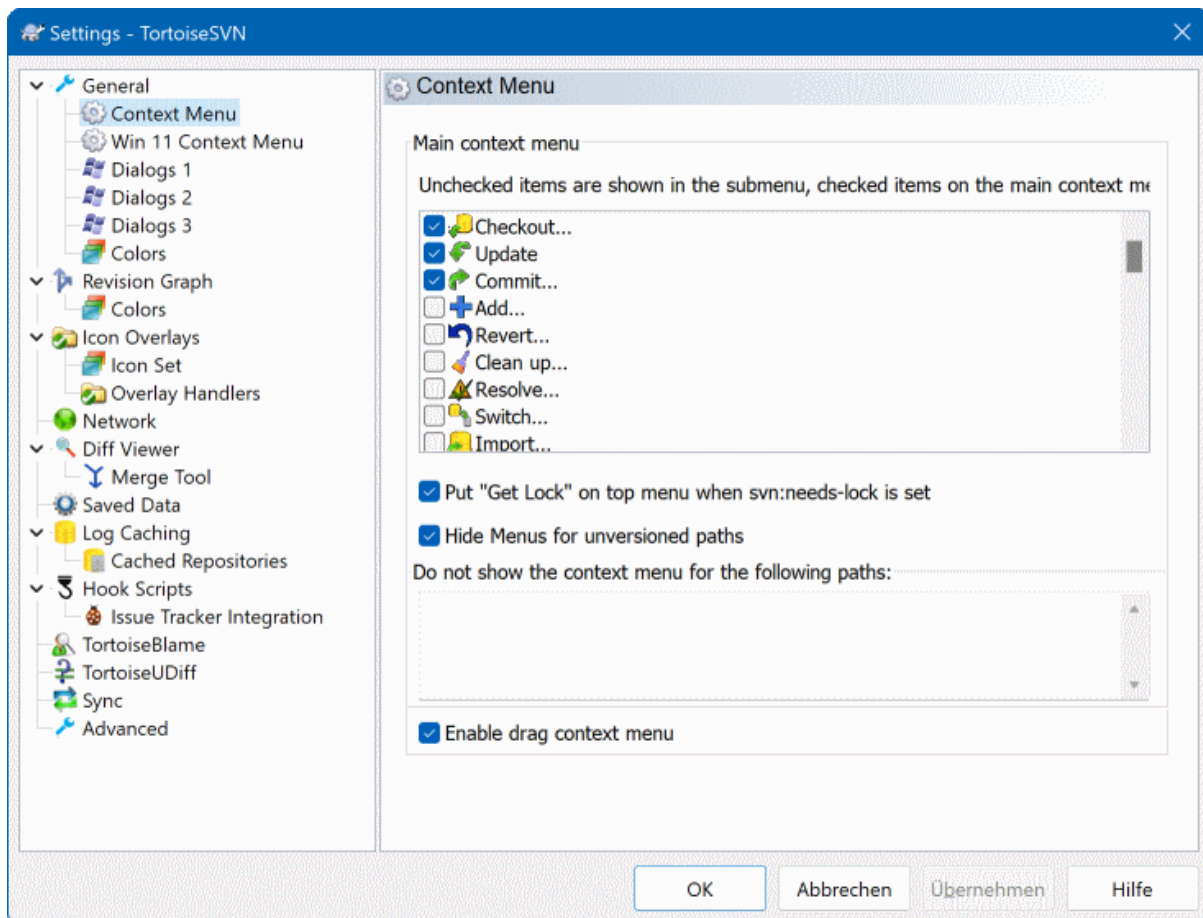
### Subversion configuratiebestand

Use **Edit** to edit the Subversion configuration file directly. Some settings cannot be modified directly by TortoiseSVN, and need to be set here instead. For more information about the Subversion `config` file see the [Runtime Configuration Area](http://svnbook.red-bean.com/en/1.8/svn.advanced.confarea.html) [http://svnbook.red-bean.com/en/1.8/svn.advanced.confarea.html]. The section on [Automatic Property Setting](http://svnbook.red-bean.com/en/1.8/svn.advanced.props.html#svn.advanced.props.auto) [http://svnbook.red-bean.com/en/1.8/svn.advanced.props.html#svn.advanced.props.auto] is of particular interest, and that is configured here. Note that Subversion can read configuration information from several places, and you need to know which one takes priority. Refer to [Configuration and the Windows Registry](http://svnbook.red-bean.com/en/1.8/svn.advanced.confarea.html#svn.advanced.confarea.windows-registry) [http://svnbook.red-bean.com/en/1.8/svn.advanced.confarea.html#svn.advanced.confarea.windows-registry] to find out more.

### Apply local modifications to svn:externals when updating

This option tells TortoiseSVN to always apply local modifications to the `svn:externals` property when updating the working copy.

### 4.31.1.1. Context Menu Instellingen



**Afbeelding 4.72. Het Instellingen Scherm, Context Menu Pagina**

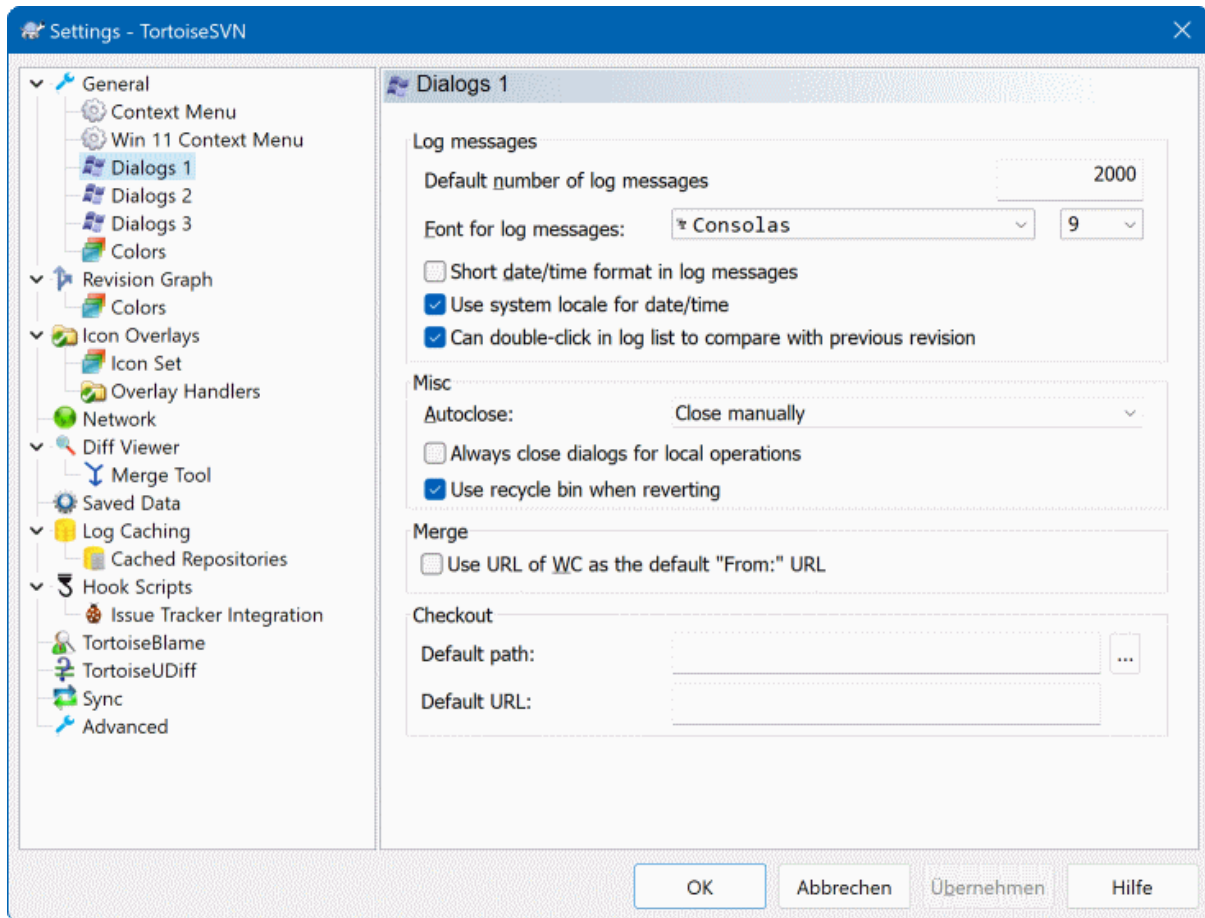
Op deze pagina kun je instellen welke TortoiseSVN context menu onderdelen aan het standaard contextmenu toegevoegd moeten worden, en welke zullen verschijnen in het TortoiseSVN submenu. Standaard staan de meeste items uitgevinkt en verschijnen dus in het submenu.

Een uitzondering wordt gemaakt voor **Plaats Blokkade**. Je kunt deze menukeuze natuurlijk ook verplaatsen naar het hoogste niveau met behulp van de bovenstaande lijst, maar aangezien de meeste bestanden helemaal niet vergrendeld hoeven te worden wordt dit menu er niet overzichtelijker door. Aan de andere kant, wanneer een bestand met de `svn:needs-lock` eigenschap deze actie elke keer nodig heeft wanneer het wordt gewijzigd, dan is het wel weer handig dat deze menukeuze op het hoogste niveau getoond wordt. Het aanvinken van deze optie zorgt er voor, dat bij het selecteren van een bestand waarvoor de `svn:needs-lock` eigenschap is ingesteld, de **Plaats Blokkade** keuze getoond zal worden op het hoogste niveau.

Most of the time, you won't need the TortoiseSVN context menu, apart for folders that are under version control by Subversion. For non- versioned folders, you only really need the context menu when you want to do a checkout. If you check the option `Hide menus for unversioned paths`, TortoiseSVN will not add its entries to the context menu for unversioned folders. But the entries are added for all items and paths in a versioned folder. And you can get the entries back for unversioned folders by holding the **Shift** key down while showing the context menu.

Wanneer er mappen op je computer zijn waar je het TortoiseSVN context-menu helemaal niet wilt zien, dan kun je de paden daarnaar intoetsen in het vak onderaan.

### 4.31.1.2. TortoiseSVN Instellingen Venster 1



#### Afbeelding 4.73. Het Instellingen Scherm, Schermen 1 Pagina

Dit scherm geeft de mogelijkheid om de TortoiseSVN dialogen aan te passen aan je eigen voorkeuren.

##### Standaard aantal logbestanden

Beperkt het aantal log-regels dat TortoiseSVN toont wanneer je kiest voor TortoiseSVN → Toon logboek. Dit kan handig zijn bij trage verbindingen. Je kunt altijd kiezen voor Toon alles of Volgende 100 om meer regels te tonen.

##### Lettersoort voor logmeldingen

Instellen van het lettertype en de lettergrootte die worden gebruikt om de logregel te tonen in het middelste venster van de Revisie Log dialoog en bij het invoeren van de beschrijving in Vastleg dialoog.

##### Verkorte datum/tijd in logboek-boodschappen

Wanneer de standaard uitgebreide meldingen te veel ruimte innemen op het scherm gebruik dan de verkorte versie.

##### Can double click in log list to compare with previous revision

If you frequently find yourself comparing revisions in the top pane of the log dialog, you can use this option to allow that action on double click. It is not enabled by default because fetching the diff is often a long process, and many people prefer to avoid the wait after an accidental double click, which is why this option is not enabled by default.

##### Automatisch sluiten

TortoiseSVN kan automatisch alle vensters met meldingen over de voortgang te sluiten wanneer de actie zonder fouten is afgerond. Deze instelling maakt het mogelijk om de condities hiervoor op te geven. De

standaard (aanbevolen) instelling is **Niet automatisch sluiten** waarmee je alle meldingen te zien krijgt en dus kunt controleren wat er allemaal is gebeurd. Maar je kunt er ook voor kiezen om bepaalde typen meldingen te negeren en de bijbehorende vensters automatisch te laten sluiten wanneer er geen bijzonderheden zijn.

**Automatisch sluiten**, behalve na samenvoegingen, verwijderen of toevoegen wil zeggen dat de voortgangsvenster zal sluiten wanneer er eenvoudige updates zijn, maar wanneer er wijzigingen uit de repository werden samengevoegd met de jouwe, of wanneer er bestanden zijn toegevoegd of verwijderd, dan blijft het dialoogvenster geopend. Het blijft ook geopend wanneer er conflicten of fouten zijn opgetreden bij het uitvoeren van de operatie.

**Automatisch sluiten**, behalve na conflicten versoepelt de criteria nog iets en zal het dialoogvenster zelfs sluiten in het geval dat samenvoegingen, toevoegingen of verwijderingen zijn. Mocht zich echter een conflict of fout voordoen, dan blijft het dialoogvenster wel geopend.

**Automatisch sluiten**, behalve na fouten sluit het dialoogvenster zelfs wanneer er conflicten zijn opgetreden. Uitsluitend wanneer er een foutsituatie is opgetreden, zoals wanneer Subversion de actie niet kan voltooien, zal het dialoogvenster geopend blijven. Bijvoorbeeld wanneer een update mislukt omdat de server onbereikbaar is of een vastlegging mislukt doordat de werkkopie te oud is.

#### Sluit dialogen altijd voor lokale bewerkingen

Local operations like adding files or reverting changes do not need to contact the repository and complete quickly, so the progress dialog is often of little interest. Select this option if you want the progress dialog to close automatically after these operations, unless there are errors.

#### Gebruik prullenbak bij herstellen

Wanneer je lokale aanpassingen ongedaan maakt gaan de zelf aangebrachte wijzigingen weer verloren. TortoiseSVN geeft je een extra vangnet door de aangepaste file naar de prullenbak te zenden voordat deze door de voorgaande versie worden overschreven. Wanneer je deze stap wilt overslaan kun je deze optie uit zetten.

#### Gebruik URL van werkkopie als standaard “Van:” URL

In de samenvoeg-dialoog wordt standaard de **Van:-URL** onthouden voor de volgende keer. Maar sommige mensen willen samenvoegen vanaf veel verschillende plaatsen en willen hier liever de URL van hun huidige werkkopie ingevuld hebben. Deze kan dan vervolgens aangepast worden naar een vergelijkbaar pad in een andere aftakking.

#### Standaard ophaalpad

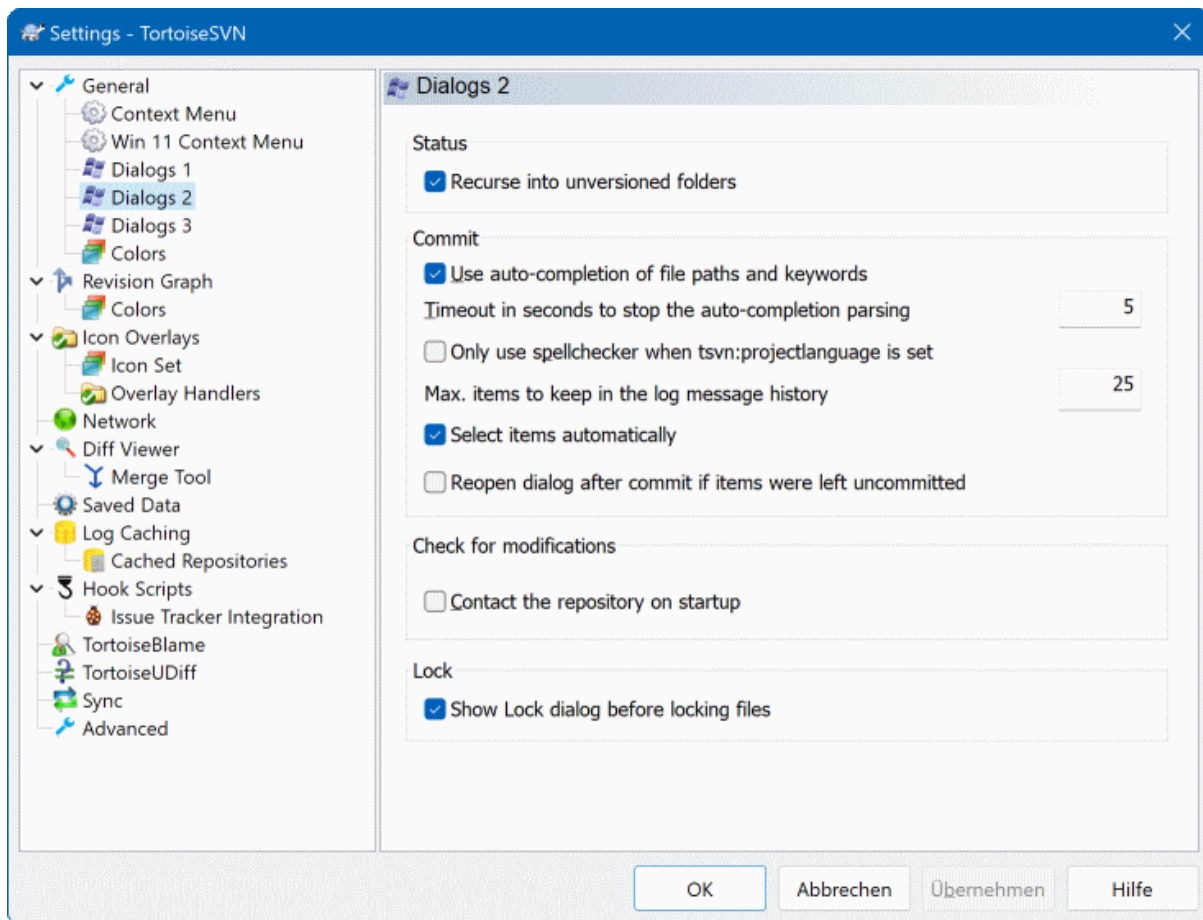
Het standaard pad voor het ophalen kan hier opgegeven worden. Wanneer dit ophalen altijd op dezelfde plaats gedaan wordt, dan is het handig dat de schijfaanuiding en de map al ingevuld zijn en alleen nog de nieuwe mapnaam aan het einde ingevuld hoeft te worden.

#### Standaard ophaal URL

Je kunt ook de standaard URL voor het ophalen opgeven. Wanneer je vaak sub-projecten van een erg groot project ophaalt, kan het handig zijn dat de URL alvast ingevuld is, zodat je daaraan alleen nog het sub-project hoeft toe te voegen.



### 4.31.1.3. TortoiseSVN Instellingen Venster 2



**Afbeelding 4.74. Het Instellingen Scherm, Schermen 2 Pagina**

Recursie toepassen op mappen die niet onder versiebeheer vallen

Wanneer deze optie is aangevinkt (standaard) wordt telkens wanneer de status van een map zonder versiebeheer wordt getoond in het **Toevoegen, Vastleggen of Kijk of er updates zijn** venster, ook elk onderliggend bestand en map getoond. Wanneer deze optie uit staat wordt alleen de bovenste map zonder versiebeheer getoond. Uitzetten van deze optie vermindert vervuiling in de dialogen. In het geval dat je een map zonder versiebeheer selecteert voor toevoegen, dan wordt deze recursief toegevoegd.

In het **Kijk of er updates zijn** scherm kun je er voor kiezen om genegeerde objecten te laten zien. Als deze optie aangevinkt is, dan worden als een genegeerd object gevonden wordt ook alle onderliggende objecten getoond.

Gebruik automatisch-aanvullen voor bestandslocaties en trefwoorden

De dialoog voor het vastleggen kent ook een mogelijkheid om een lijst met namen van de betreffende bestanden te kunnen verwerken. Wanneer je de eerste 3 letters van een item in de lijst intoetst verschijnt de "automatisch-aanvullen" pop-up en kun je Enter toetsen om de bestandsnaam aan te vullen. Vink dit vakje aan om deze functie te activeren.

Timeout (in seconden) waarna automatisch-aanvullen wordt gestopt

De automatische-aanvul functie kan erg langzaam zijn als er veel grote bestanden gecontroleerd moeten worden. Met deze timeout voorkom je dat het vastleg scherm te lang opgehouden wordt. Als je belangrijke automatische-aanvul gegevens mis, dan kun je de timeout verlengen.

Gebruik spellingscontrole alleen als `tsvn:projectlanguage` een waarde heeft

Als je de spellingscontrole niet voor elke vastlegging wilt gebruiken, vink dan deze optie aan. De spellingscontrole zal dan nog wel beschikbaar zijn als de project eigenschappen dat vereisen.



#### Maximum aantal boodschappen in de geschiedenis van logboek-boodschappen

Wanneer je een log opmerking intoetst in de vastleg-dialoog, slaat TortoiseSVN dit op voor eventueel hergebruik. Standaard zullen per archief de laatste 25 opmerkingen worden bewaard, maar dit aantal kan hier ingesteld worden. Wanneer je veel verschillende archieven hebt, wil je dit aantal wellicht verlagen om te voorkomen dat het register volloopt.

Merk op dat deze instelling uitsluitend betrekking op de opmerkingen die je intoetst op deze computer. Het heeft dus niets te maken met de log cache.

#### Selecteer items automatisch

Normaal gesproken worden in de vastleggingsvensters alle gewijzigde (en onder versiebeheer vallende) objecten al automatisch geselecteerd voor vastlegging. Wanneer je er echter liever hebt dat er niets geselecteerd is zodat je de items handmatig kunt selecteren, zet deze optie dan uit.

#### Dialoog heropenen na inchecken als er items niet ingecheckt zijn

This reopens the commit dialog automatically at the same directory after a successful commit. The dialog is reopened only if there still are items left to commit.

#### Leg contact met het archief bij het opstarten

De Kijk of er updates zijn dialoog vergelijkt standaard met de werkkopie, en vergelijkt alleen met de archief wanneer je op **Controleer archief** klikt. Wanneer je altijd de archief wilt controleren, dan kun je er met deze instelling voor zorgen, dat dit standaard zo is.

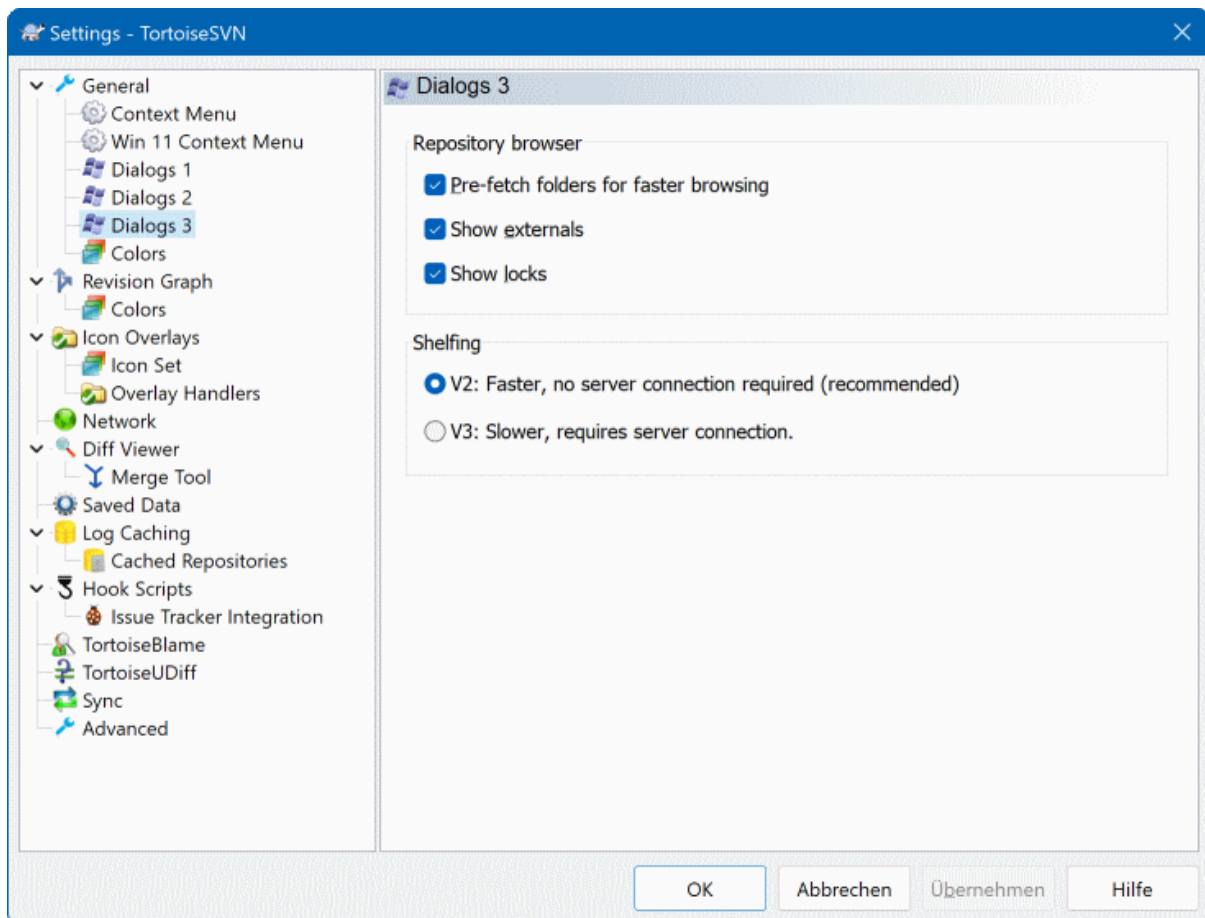
#### Toon de Blokkeer dialoog voordat bestanden geblokkeerd worden

Wanneer je een of meer bestanden selecteert en dan kiest voor TortoiseSVN → **Plaats blokkade** om deze bestanden te blokkeren, dan is het bij sommige projecten gebruikelijk om een blokkade-notitie achter te laten met de verklaring waarom je deze bestanden geblokkeerd hebt. Wanneer je deze blokkade-notities niet gebruikt, kun je deze optie uitschakelen om deze dialoog over te slaan en de bestanden meteen te blokkeren.

Wanneer je het blokkerr commando gebruik voor een map, dan wordt altijd de plaats blokkade dialoog getoond aangezien die de mogelijkheid geeft om bestanden voor de blokkade te selecteren.

Wanneer jouw project de `tsvn:lockmsgminsize` eigenschap gebruikt, dan zal de plaats blokkade dialoog getoond worden ongeacht deze instelling aangezien het project blokkade notities *verplicht*.

#### 4.31.1.4. TortoiseSVN Dialog Settings 3



#### Afbeelding 4.75. The Settings Dialog, Dialogs 3 Page

Settings for the repository browser:

Haal mappen vooraf op voor sneller navigeren

If this box is checked (default state), then the repository browser fetches information about shown folders in the background. That way as soon as you browse into one of those folders, the information is already available.

Some servers however can't handle the multiple requests this causes or when not configured correctly treat so many requests as something bad and start blocking them. In this case you can disable the pre-fetching here.

Toon externen

If this box is checked (default state), then the repository browser shows files and folders that are included with the `svn:externals` property as normal files and folders, but with an overlay icon to mark them as from an external source.

As with the pre-fetch feature explained above, this too can put too much stress on weak servers. In this case you can disable this feature here.

There are two versions of shelving implemented in SVN. Here you can select which version you want to use. Note that changing this setting might require an OS restart to take effect.

V2

this version is much faster than V3 and is the recommended version to use.

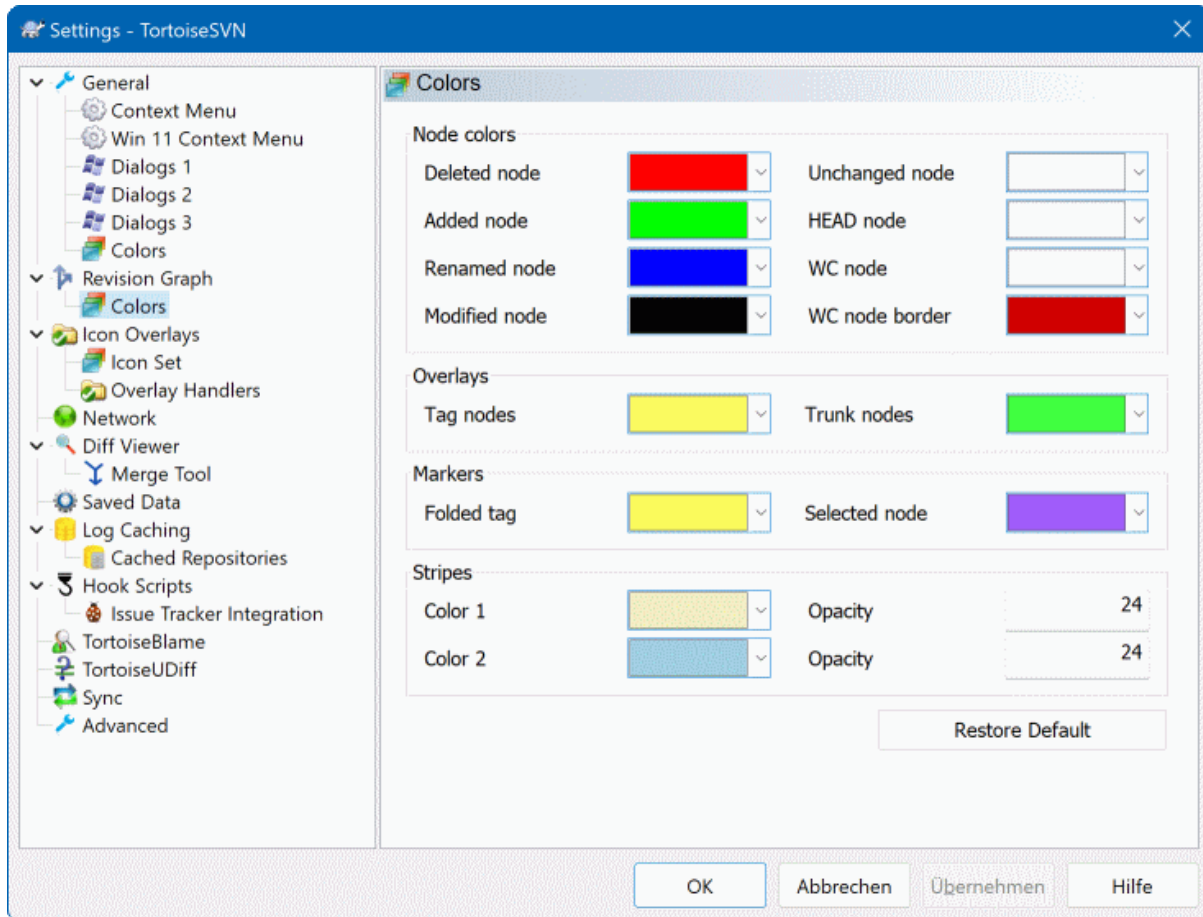
However, the speed comes at a prize: V2 does not handle directory changes, and can't handle copies and moves of files.

V3

this is the latest version of the shelving feature. It can handle changes to directories as well as file moves/copies.

However, V3 is much slower than V2 and can be unusably slow for big repositories or if you have a slow connection to the repository.

#### 4.31.1.5. TortoiseSVN kleurinstellingen



**Afbeelding 4.76. Het Instellingen Scherm, Kleuren Pagina**

Deze dialoog maakt het mogelijk om de tekstkleuren die in TortoiseSVN dialogen worden gebruikt naar eigen voorkeur in te stellen.

Mogelijk of echt conflict / geblokkeerd

Er is een conflict opgetreden tijdens een vastlegging of kan mogelijk optreden bij een samenvoeging. De vastlegging wordt geblokkeerd door een bestand/map zonder versiebeheer die dezelfde naam heeft als een met versiebeheer.

Deze kleur wordt tevens gebruikt voor foutmeldingen in de voortgangsvensers.

Toegevoegde bestanden

Aan het archief toegevoegde objecten.

Ontbreekt / verwijderd / vervangen

Items die verwijderd zijn uit het archief, ontbreken in de werkkopie of uit de werkkopie verwijderd en vervangen zijn door een gelijknamig bestand.

Samengevoegd

Wijzingen van het archief die met succes zijn samengevoegd in de werkkopie zonder dat er conflicten zijn opgetreden.

**Gewijzigd / gekopieerd**

Toevoegingen met historie, of gekopieerde paden in het archief. Wordt ook gebruikt in de log vensters voor onderdelen die gekopieerde objecten bevatten.

**Verwijderde knoop**

Een item dat is verwijderd uit het archief.

**Toegevoegde knoop**

Een item dat is toegevoegd aan het archief met behulp van een toevoeg, kopiëer of verplaats actie.

**Hernoemde knoop**

Een item dat is hernoemd in het archief.

**Vervangen knoop**

Het oorspronkelijke item is verwijderd en een nieuw item met dezelfde naam is ervoor in de plaats gekomen.

**Filter overeenkomst**

When using filtering in the log dialog, search terms are highlighted in the results using this colour.

other settings:

**Donker Thema**

The dialogs in TortoiseSVN can be shown in a dark mode on Windows 10 1809 and later. This feature also requires that dark mode for applications is enabled in the Windows 10 settings.

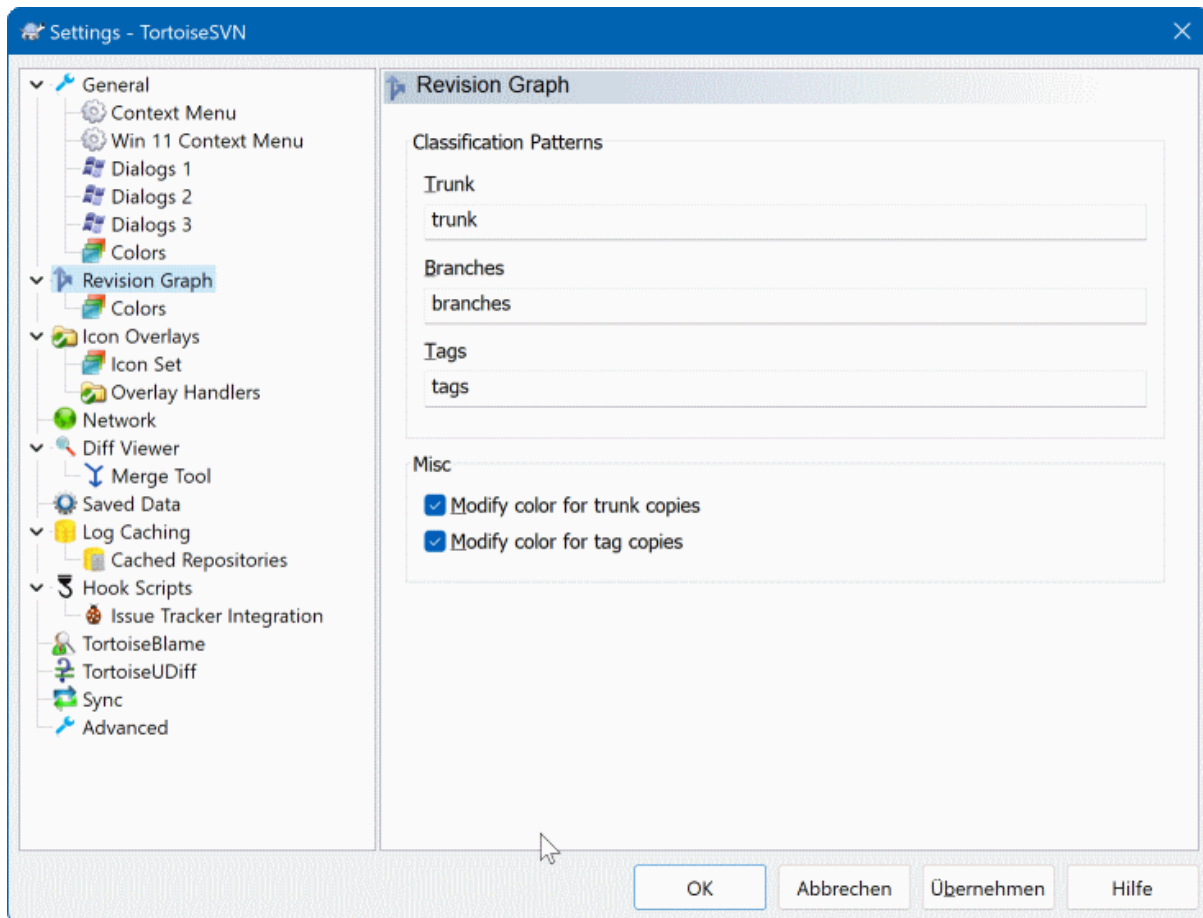


---

**Belangrijk**

Note that not all controls in all dialogs are shown in a dark theme.

## 4.31.2. Revisie Grafiek Instellingen



**Afbeelding 4.77. Het instellingen scherm, Revisie schema pagina**

### Classificatie patronen

Het revisieschema tracht een duidelijker beeld te geven van uw archiefstructuur door onderscheid te maken tussen basis, vertakkingen en labels. Omdat dergelijke classificaties niet in Subversion zijn ingebouwd, wordt deze informatie uit de namen van de paden gehaald. De standaard instellingen verwachten dat u conventionele engelse benamingen gebruikt zoals voorgesteld in de documentatie van Subversion, maar kan het gebruik hiervan afwijken.

Specificeer de patronen om deze paden te herkennen in de drie gegeven velden. De patronen zullen worden niet hoofdlettergevoelig worden vergeleken, maar u dient ze in kleine letters in te geven. Jokertekens \* en ? werken als gewoonlijk en u kunt ; gebruiken om meerdere patronen van elkaar te scheiden. Voeg geen extra witruimte in omdat dit wordt meegenomen in de vergelijkingspecificatie.



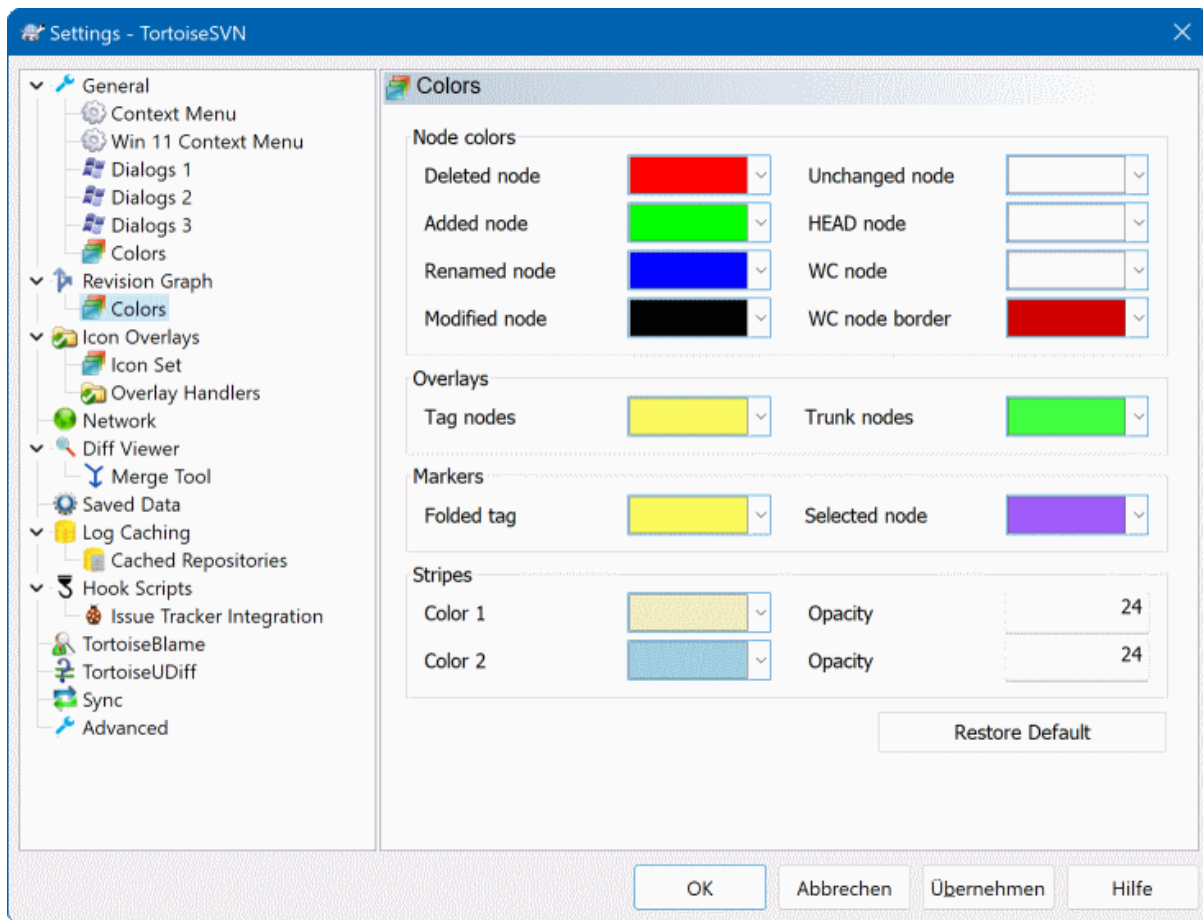
### Commit tag detection

Please note that these patterns are also used to detect commits to a tag, not just for the revision graph.

### Bewerk Kleuren

Kleuren worden in het revisieschema gebruikt om het type knoop aan te geven, bijv. of een knoop is toegevoegd, verwijderd of hernoemd. Om u te laten assisteren in het selecteren van de knoop classificatie kunt u het revisieschema toestaan om kleuren te mengen om een indicatie te geven van de combinatie knooptype en classificatie. Als het selectievak is aangevinkt wordt menging toegepast. Als het selectievak niet is aangevinkt wordt de kleur gebruikt om alleen het knooptype aan te geven. Gebruik het kleurselectiescherm om de gebruikte specifieke kleuren vast te leggen.

### 4.31.2.1. Revisie Grafiek Kleuren



**Afbeelding 4.78. Het instellingen scherm, Revisieschema kleuren pagina**

Deze pagina biedt u de mogelijkheid om de te gebruiken kleuren te configureren. Let op dat de aangegeven kleur de ondoorzichtige kleur is. De meeste knopen worden gekleurd door een mengsel van de knooptype kleur, achtergrondkleur en eventueel de classificatiekleur.

#### Verwijderde knoop

Items welke zijn verwijderd en niet naar elders zijn gekopieërd binnen dezelfde revisie.

#### Toegevoegde knoop

Nieuw toegevoegde of gekopieëerde (toevoegen met geschiedenis) items.

#### Hernoemde knoop

Items welke zijn verwijderd uit de ene locatie en toegevoegd aan een andere locatie binnen dezelfde revisie.

#### Aangepaste knoop

Eenvoudige wijzigingen zonder enige toevoeging of verwijdering.

#### Ongewijzigde knoop

Kan worden gebruikt om de revisie aan te geven welke wordt gebruikt als bron van een kopie, zelfs als er geen wijziging (aan het item dat schematisch wordt weergegeven) heeft plaatsgevonden binnen die revisie.

#### HEAD knoop

Huidige HEAD revisie in het archief.

#### Werkkopie knoop

Als u er voor kiest een extra knoop weer te geven voor uw aangepast werkkopie, gekoppeld aan zijn laatste vastlegging revisie in het schema, gebruik deze kleur.

**Werkkopie knoop rand**

Als u ervoor kiest om weer te geven of een werkkopie is aangepast, gebruik deze gekleurde rand voor de werkkopieknoop wanneer aanpassingen worden gevonden.

**Label knopen**

Knopen welke worden geclassificeerd als labels kunnen worden gemengt met deze kleur.

**Basislijn knopen**

Knopen welke worden geclassificeerd als basislijn kunnen worden gemengt met deze kleur.

**Ingeklapt label merktekens**

Als u labels inklapt om ruimte te winnen worden de labels gemarkeerd in de bronkopie door een blok in deze kleur.

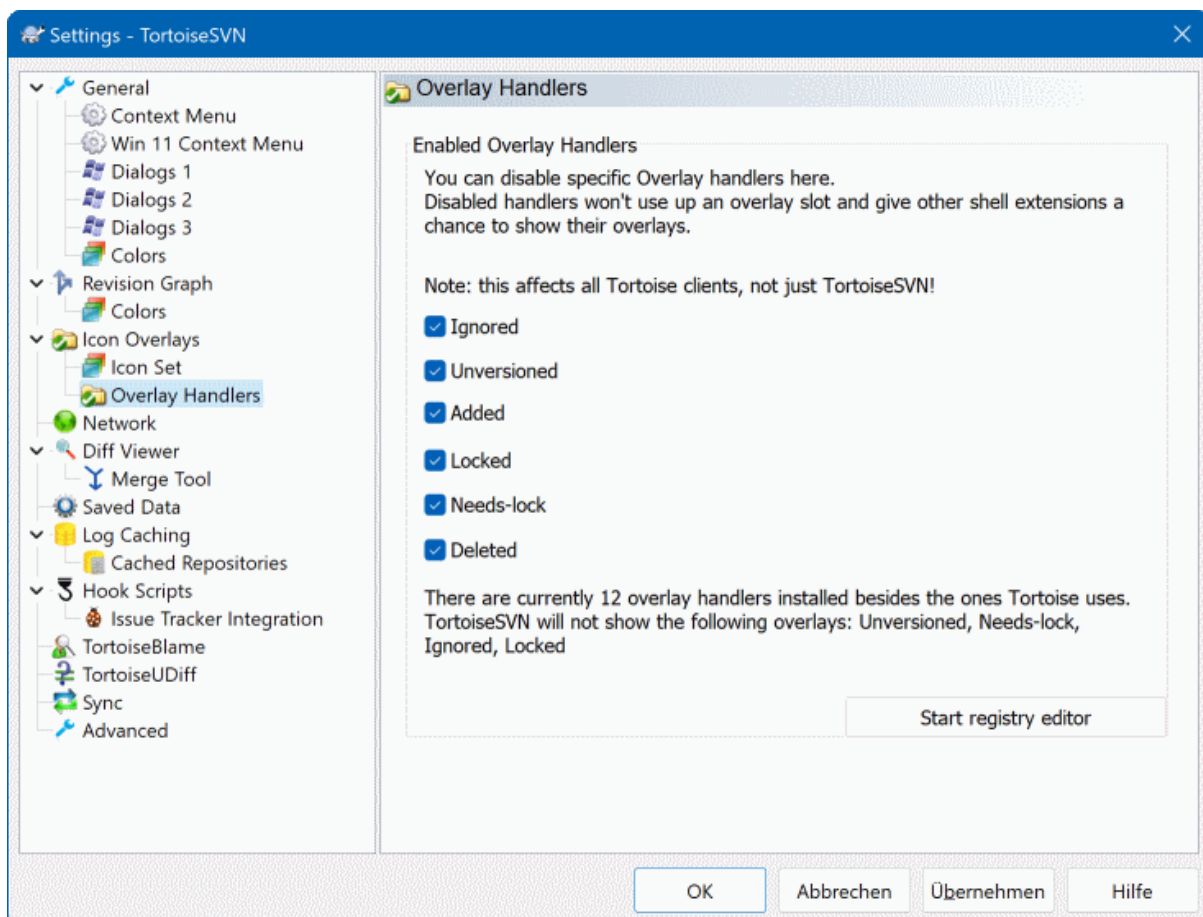
**Geselecteerde knoop markering**

Als u met links op een knoop klikt om deze te selecteren, is het merkteken dat wordt gebruikt om de selectie aan te geven een blok in deze kleur.

**Strepen**

Deze kleuren worden gebruikt wanneer het schema wordt opgesplitst in sub-boomstructuren en de achtergrond wordt gekleurd als afwisselende strepen om te helpen de verschillende boomstructuren te onderscheiden.

### 4.31.3. Overlappende Pictogram Instellingen



**Afbeelding 4.79. Het instellingen scherm, Overlappend pictogram pagina**

Deze pagina biedt u de mogelijkheid om de items te kiezen waarvoor TortoiseSVN de overlappende pictogrammen weergeeft.

Omdat het een behoorlijke tijd kost om de status van een werkkopie op te halen, gebruikt TortoiseSVN een cache om de status in op te slaan, zodat de verkenner niet teveel stukt tijdens het tonen van de overlappende pictogrammen. U kunt het type cache hier kiezen dat TortoiseSVN moet gebruiken volgens uw systeem en grootte van werkkopie:

#### Standaard

Plaats alle statusinformatie in een apart proces (`TSVNCache.exe`) in de cache. Dat proces observeert alle schijven op wijzigingen en haalt de status opnieuw op wanneer bestanden binnen een werkkopie worden gewijzigd. Het proces wordt met de laagst mogelijke prioriteit uitgevoerd zodat andere programma's er niet door worden gehinderd. Dat betekent ook dat de statusinformatie niet *realtime* is, maar het enige seconden kan duren voordat de overlappende pictogrammen wijzigen.

Voordeel: de overlappende pictogrammen tonen de status recursief, bijv. als een bestand diep in een werkkopie wordt aangepast, zullen alle mappen tot aan de basis van de werkkopie ook het aangepaste overlappend pictogram tonen. En doordat het proces berichten naar de shell kan sturen worden de overlappende pictogrammen in het linker boomstructuur venster gewoonlijk ook wijzigen.

Nadeel: het proces wordt continu uitgevoerd, zelfs als u niet aan uw projecten werkt. Het gebruikt ook zo'n 10-50 Mb RAM, afhankelijk van het aantal en grootte van uw werkkopieën.

#### Shell

Cachen wordt direct binnen de shell-extensie dll uitgevoerd, maar alleen voor de op dat moment zichtbare map. Iedere keer dat u naar een andere map navigeert wordt de statusinformatie opnieuw opgehaald.

Voordeel: heeft slechts weinig geheugen nodig (rond de 1Mb RAM) en kan de status in *realtime* tonen.

Nadeel: Doordat slechts één map in de cache wordt geplaatst, geven de overlappende pictogrammen de status niet recursief weer. Voor grote werkkopieën kan het meer tijd kosten om een map in de verkenner te tonen dan met de standaard cache. Ook is de mime-type kolom niet beschikbaar.

#### Geen

Met deze instelling haalt TortoiseSVN geen enkele status op in Verkenner. Hierdoor krijgen bestanden geen overlappend pictogram en mappen krijgen alleen een 'normaal' overlappend pictogram als zij onder versiebeheer vallen. Andere overlappende pictogrammen worden niet getoond en extra kolommen zijn ook niet beschikbaar.

Voordeel: gebruikt absoluut geen extra geheugen en vertraagt de Verkenner niet tijdens bladeren.

Nadeel: Statusinformatie van bestanden en mappen wordt in de Verkenner niet getoond. Om te zien of uw werkkopieën zijn aangepast dient u het "Kijk of er wijzigingen zijn" scherm te raadplegen.

Standaard zullen de overlappende pictogrammen en contextmenu's in alle openen/opslaan schermen alsook in Windows Verkenner verschijnen. Als u wilt dat ze *alleen* in Windows Verkenner verschijnen, vink dan het **Toon overlays en contextmenu alleen in verkenner** selectievak aan.

You can force the status cache to *None* for elevated processes by checking the **Disable status cache for elevated processes** box. This is useful if you want to prevent another `TSVNCache.exe` process getting created with elevated privileges.

U kunt er ook voor kiezen om mappen aan te merken als aangepast als deze items zonder versiebeheer bevatten. Dit kan handig zijn om u eraan te herinneren dat u nieuwe bestanden heeft aangemaakt welke nog niet onder versiebeheer vallen. Deze optie is enkel beschikbaar als u de *standaard* status cache optie (zie hieronder) gebruikt.

If you have files in the `ignore-on-commit` changelist, you can chose to make those files not propagate their status to the parent folder. That way if only files in that changelist are modified, the parent folder still shows the unmodified overlay icon.

De volgende groep biedt de mogelijkheid om de aan te geven welke opslagklassen overlappende pictogrammen moeten weergeven. Standaard zijn alleen harde schijven geselecteerd. U kunt zelfs alle overlappende pictogrammen uitzetten, maar wat kan is daar de lol van zijn?



Netwerkschijven kunnen erg traag zijn, dus worden de overlappende pictogrammen standaard niet voor werkkopieën, die zich op netwerkschijven bevinden, getoond.

USB Flash schijven blijken een speciaal geval te vormen doordat het schijftype door het apparaat zelf wordt geïdentificeerd. Sommige doen zich voor als harde schijf en andere als verwijderbare schijf.

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, or if you only want TortoiseSVN to look in specific folders.

Any path you specify here is assumed to apply recursively, so none of the child folders will show overlays either. If you want to exclude *only* the named folder, append ? after the path.

Hetzelfde geldt voor de **Inclusief paden**. Behalve dan dat voor die paden de overlappende pictogrammen worden getoond, zelfs als de overlappende pictogrammen zijn uitgeschakeld voor dat specifieke schijftype, of door een uitgesloten pad zoals hierboven aangegeven.

Users sometimes ask how these three settings interact. For any given path check the include and exclude lists, seeking upwards through the directory structure until a match is found. When the first match is found, obey that include or exclude rule. If there is a conflict, a single directory spec takes precedence over a recursive spec, then inclusion takes precedence over exclusion.

An example will help here:

```
Exclude:
C:
C:\develop\?
C:\develop\tsvn\obj
C:\develop\tsvn\bin
```

```
Include:
C:\develop
```

These settings disable icon overlays for the C: drive, except for `c:\develop`. All projects below that directory will show overlays, except the `c:\develop` folder itself, which is specifically ignored. The high-churn binary folders are also excluded.

TSVNCache.exe gebruikt deze paden ook om zijn doorzoeken van de paden te beperken. Als u wilt dat slechts in bepaalde mappen wordt gekeken, schakel dan alle schijftypes uit en includeer alleen die mappen welke u ook daadwerkelijk wilt laten doorzoeken.



## Uitsluiten van SUBST schijven

It is often convenient to use a SUBST drive to access your working copies, e.g. using the command

```
subst T: C:\TortoiseSVN\trunk\doc
```

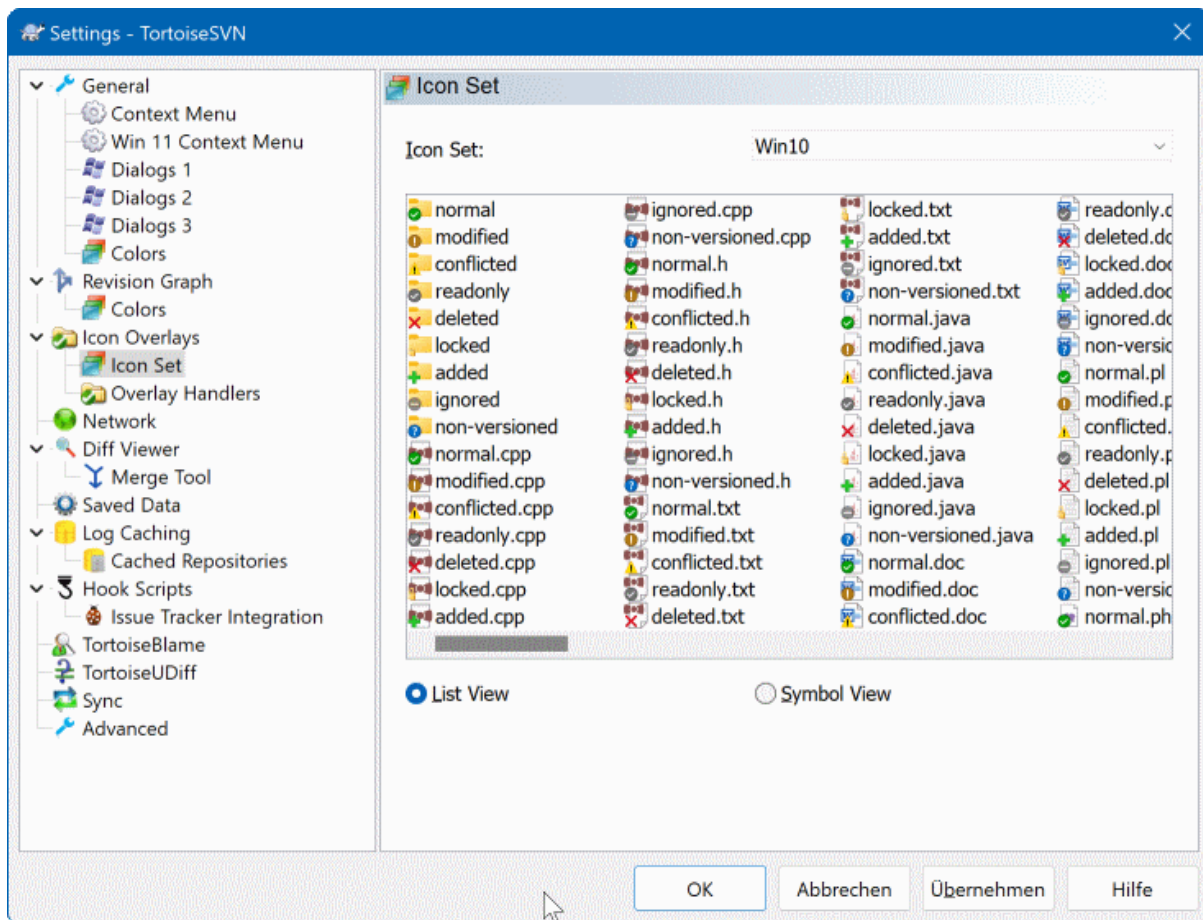
However this can cause the overlays not to update, as TSVNCache will only receive one notification when a file changes, and that is normally for the original path. This means that your overlays on the subst path may never be updated.

Een gemakkelijke manier om dit te omzeilen is door het originele pad uit te sluiten voor het tonen van overlappende pictogrammen, zodat, in de plaats hiervan, de overlappende pictogrammen worden getoond op het subst pad.

Sometimes you will exclude areas that contain working copies, which saves TSVNCache from scanning and monitoring for changes, but you still want a visual indication that a folder contains a working copy. The Show excluded root folders as 'normal' checkbox allows you to do this. With this option, working copy root folders in any excluded area (drive type not checked, or specifically excluded) will show up as normal and up-to-date, with a green check mark. This reminds you that you are looking at a working copy, even though the folder overlays may not be correct. Files do not get an overlay at all. Note that the context menus still work, even though the overlays are not shown.

Een speciale uitzondering op de regel is dat voor de schijven A : en B : nooit de optie Toon uitgesloten mappen als 'normaal' zal worden toegekend. Dit is omdat Windows wordt gedwongen om op de schijf te kijken, wat kan resulteren in een vertraging van meerdere seconden wanneer Verkenner wordt gestart, zelfs als uw PC een floppy-drive bezit.

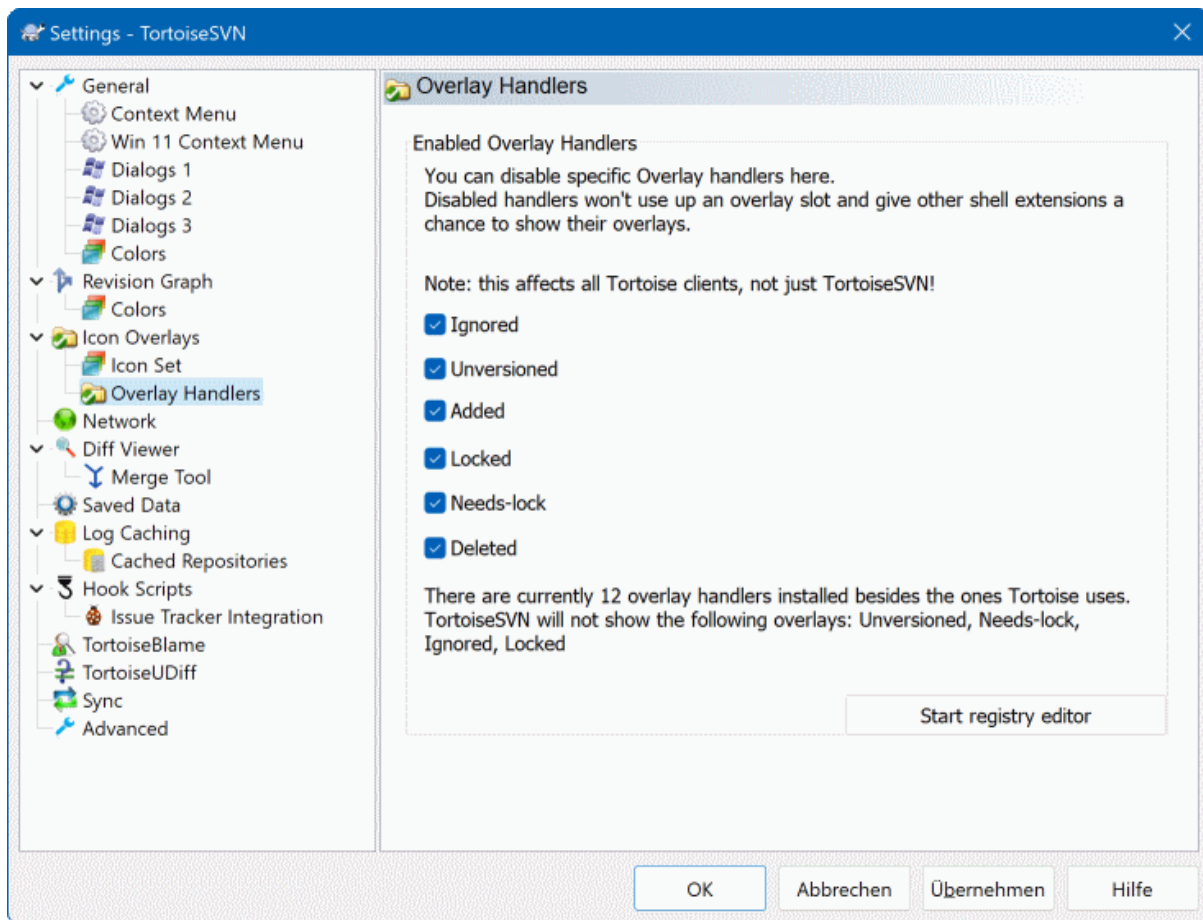
#### 4.31.3.1. Pictogram Set Selectie



**Afbeelding 4.80. Het instellingen scherm, Pictogramset pagina**

U kunt de overlappende pictogrammen set wijzigen naar degene van uw wens. Onthoud dat wanneer u de overlappende pictogrammen set wijzigt, u mogelijk uw computer opnieuw moet opstarten om de wijzigingen te effectueren.

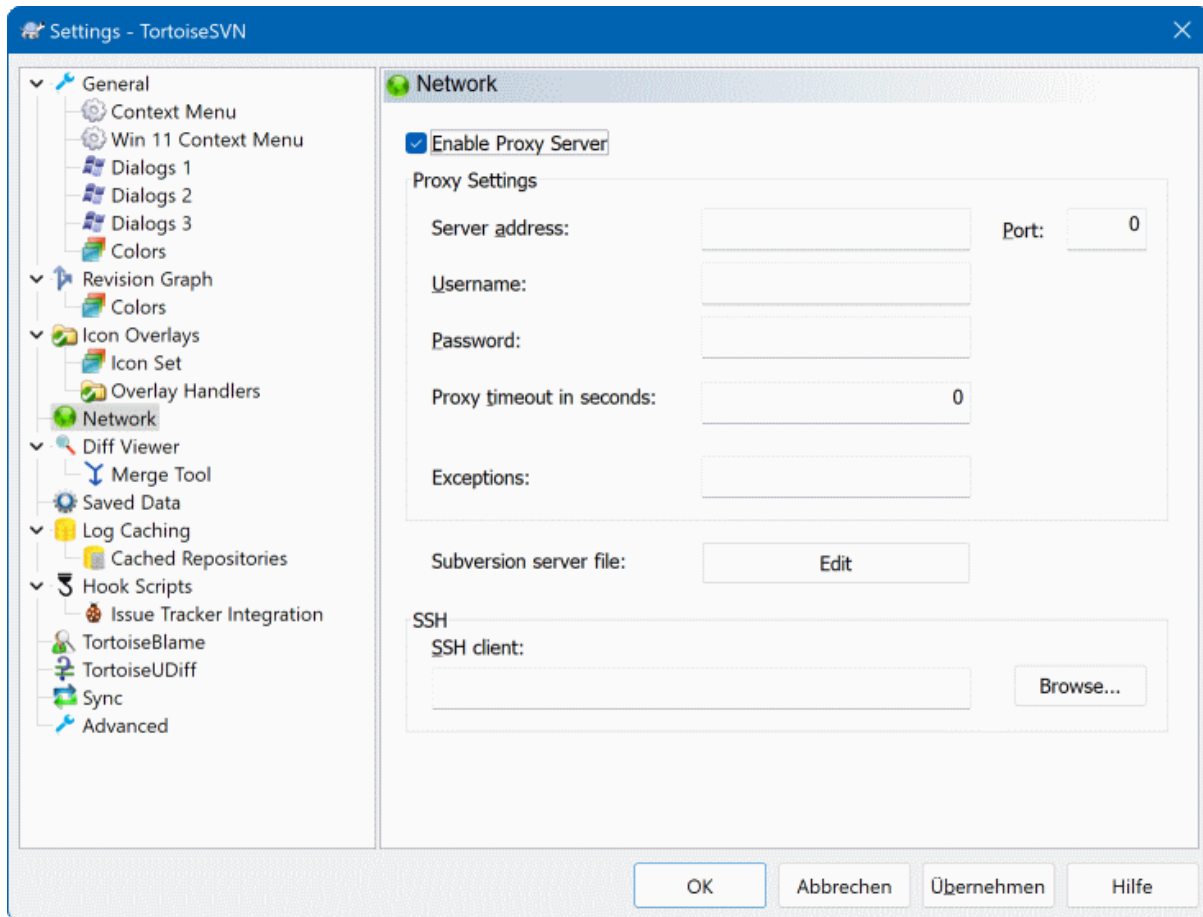
### 4.31.3.2. Ingeschakelde overlay afhandeling



**Afbeelding 4.81. The Settings Dialog, Icon Handlers Page**

Because the number of overlays available is severely restricted, you can choose to disable some handlers to ensure that the ones you want will be loaded. Because TortoiseSVN uses the common TortoiseOverlays component which is shared with other Tortoise clients (e.g. TortoiseCVS, TortoiseHg) this setting will affect those clients too.

### 4.31.4. Network Instellingen



**Afbeelding 4.82. Het instellingen scherm, Netwerk pagina**

Hier kunt u uw proxy-server configureren als u er een nodig heeft om door uw bedrijfs-firewall te geraken.

If you need to set up per-repository proxy settings, you will need to use the Subversion `servers` file to configure this. Use `Edit` to get there directly. Consult the [Runtime Configuration Area](http://svnbook.red-bean.com/en/1.8/svn.advanced.confarea.html) [http://svnbook.red-bean.com/en/1.8/svn.advanced.confarea.html] for details on how to use this file.

U kunt ook aangeven welk programma TortoiseSVN moet gebruiken om een veilige verbinding te leggen met een `svn+ssh` archief. We raden aan dat u `TortoisePlink.exe` gebruikt. Dit is een versie van het populaire `Plink` programma en wordt meegeleverd met TortoiseSVN, maar is gecompileerd als een schermloze applicatie, dus u krijgt geen DOS-venster dat tevoorschijn komt bij elke keer dat u zich authenticceerd.

You must specify the full path to the executable. For `TortoisePlink.exe` this is the standard TortoiseSVN bin directory. Use the `Browse` button to help locate it. Note that if the path contains spaces, you must enclose it in quotes, e.g.

```
"C:\Program Files\TortoiseSVN\bin\TortoisePlink.exe"
```

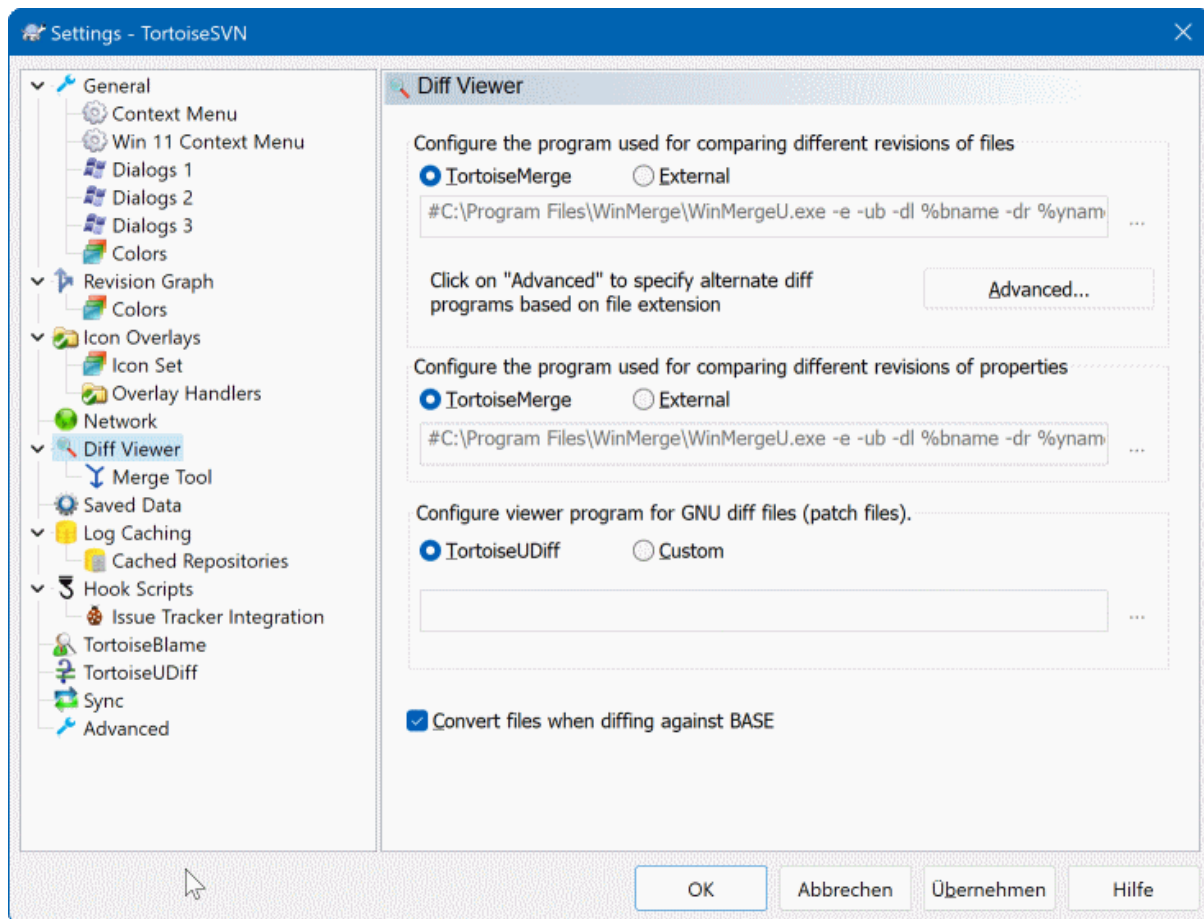
Een bijwerking van het niet hebben van een scherm is dat foutberichten nergens heen kunnen, dus als authenticatie mislukt zal u gewoon een melding krijgen welke iets aangeeft als “Kan niet schrijven naar standaard uitvoer”. Hiervoor adviseren we dat u eerst het standaard `Plink` instelt. Als alles werkt, kunt u `TortoisePlink` met exact dezelfde parameters gebruiken.

`TortoisePlink` does not have any documentation of its own because it is just a minor variant of `Plink`. Find out about command line parameters from the [PuTTY website](https://www.chiark.greenend.org.uk/~sgtatham/putty/) [https://www.chiark.greenend.org.uk/~sgtatham/putty/].

Om te voorkomen dat u telkens om een wachtwoord wordt gevraagd, kunt u er ook voor kiezen om een wachtwoord caching hulpmiddel te gebruiken zoals Pageant. Dit is ook beschikbaar voor binnenhalen vanaf de PuTTY website.

Finally, setting up SSH on server and clients is a non-trivial process which is beyond the scope of this help file. However, you can find a guide in the TortoiseSVN FAQ listed under [Subversion/TortoiseSVN SSH How-To](https://tortoisesvn.net/ssh_howto.html) [https://tortoisesvn.net/ssh\_howto.html].

#### 4.31.5. Externe Programma Instellingen



**Abbeelding 4.83. Het instellingen scherm, Diff viewer pagina**

Hier kunt u uw eigen verschil/samenvoeg programma's aangeven welke TortoiseSVN moet gebruiken. De standaard instelling is om TortoiseMerge te gebruiken, welke tesamen met TortoiseSVN is geïnstalleerd.

Lees [Paragraaf 4.11.6, "Externe Verschil/Samenvoeg gereedschappen"](#) voor een lijst met enkele externe verschil/samenvoeg programma's welke door mensen worden gebruikt in combinatie met TortoiseSVN.

##### 4.31.5.1. Diff Viewer

Er kan een extern verschil-programma worden gebruikt om de verschillende revisies van bestanden met elkaar te vergelijken. Het externe programma dient de bestandsnamen van de commandoregel uit te lezen, tesamen met eventuele andere commandoregel opties. TortoiseSVN gebruikt substitutie parameters voorafgegaan door een % . Als deze één van deze parameters tegenkomt, zal de parameter worden vervangen door de geschikte waarde. De volgorde van de parameters hangt van het door u gebruikte verschil-programma af.

%base

Het originele bestand zonder uw wijzigingen

**%bname**  
De schermtitel voor het basis bestand

**%nqbyname**  
The window title for the base file, without quotes

**%mine**  
Uw eigen bestand, met uw wijzigingen

**%yname**  
De schermtitel voor uw bestand

**%nqyname**  
The window title for your file, without quotes

**%burl**  
The URL of the original file, if available

**%nqburl**  
The URL of the original file, if available, without quotes

**%yurl**  
The URL of the second file, if available

**%nqyurl**  
The URL of the second file, if available, without quotes

**%brev**  
The revision of the original file, if available

**%nqbrev**  
The revision of the original file, if available, without quotes

**%yrev**  
The revision of the second file, if available

**%nqyrev**  
The revision of the second file, if available, without quotes

**%peg**  
The peg revision, if available

**%nqpeg**  
The peg revision, if available, without quotes

**%fname**  
The name of the file. This is an empty string if two different files are diffed instead of two states of the same file.

**%nqfname**  
The name of the file, without quotes

The window titles are not pure filenames. TortoiseSVN treats that as a name to display and creates the names accordingly. So e.g. if you're doing a diff from a file in revision 123 with a file in your working copy, the names will be `filename : revision 123` and `filename : working copy`.

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
```

or with UltraCompare:

```
C:\Path-To\uc.exe %base %mine -title1 %bname -title2 %yname
```

or with DiffMerge:

```
C:\Path-To\DiffMerge.exe -nosplash -t1=%bname -t2=%yname %base %mine
```

Als u de `svn:keywords` eigenschap gebruikt om trefwoorden uit te breiden, en dan vooral de *revisie* van een bestand, kan er een verschil bestaan tussen bestanden welke slechts wordt veroorzaakt door de huidige waarde van het trefwoord. Ook wanneer u `svn:eol-style = native` gebruikt, heeft het BASE bestand pure LF regeleinden terwijl uw bestand CR-LF regeleinden zal hebben. TortoiseSVN zal normaal gesproken deze verschillen automatisch verbergen door eerst het BASE bestand te doorlopen om trefwoorden uit te breiden en de regeleinden aan te passen, voordat de verschil-operatie wordt uitgevoerd. Dit kan echter veel tijd kosten bij grote bestanden. Als Bestanden converteren bij het vergelijken met BASE niet is aangevinkt zal TortoiseSVN het vooraf bewerken van de bestanden overslaan.

U kunt ook een ander verschil-gereedschap aangeven om te gebruiken bij Subversion eigenschappen. Omdat deze gewoonlijk korte eenvoudige tekenreeksen zijn, zou u een eenvoudiger en kleinere viewer kunnen gebruiken.

Als u een alternatief verschil-gereedschap heeft geconfigureerd, kunt u TortoiseMerge *en* het gereedschap van derden via het contextmenu benaderen. Contextmenu → Verschil gebruikt het primaire verschil-gereedschap, en **Shift+** Contextmenu → Verschil gebruikt het secundaire verschil-gereedschap.

At the bottom of the dialog you can configure a viewer program for unified-diff files (patch files). No parameters are required. The Default setting is to use TortoiseUDiff which is installed alongside TortoiseSVN, and colour-codes the added and removed lines.

Since Unified Diff is just a text format, you can use your favourite text editor if you prefer.

#### 4.31.5.2. Merge Tool

Een extern samenvoegprogramma dat wordt gebruikt om conflicterende bestanden op te lossen. Er wordt op dezelfde manier van parameter substitutie gebruik gemaakt als bij het verschil-programma.

`%base`  
het originele bestand zonder uw of andermans wijzigingen

`%bname`  
De schermtitel voor het basis bestand

`%nqbname`  
The window title for the base file, without quotes

`%mine`  
uw eigen bestand, met uw wijzigingen

`%yname`  
De schermtitel voor uw bestand

`%nqyname`  
The window title for your file, without quotes

`%theirs`  
het bestand zoals het is in het archief

`%tname`  
de schermtitel voor het bestand in het archief

`%nqtname`  
The window title for the file in the repository, without quotes

`%merged`  
het conflicterende bestand, het resultaat van de samenvoeg-operatie

`%mname`  
de schermtitel voor het samengevoegde bestand

`%nqmname`  
The window title for the merged file, without quotes

`%fname`  
The name of the conflicted file

`%nqfname`  
The name of the conflicted file, without quotes

For example, with Perforce Merge:

```
C:\Path-To\P4Merge.exe %base %theirs %mine %merged
```

or with KDiff3:

```
C:\Path-To\kdiff3.exe %base %mine %theirs -o %merged  
--L1 %bname --L2 %yname --L3 %tname
```

or with Araxis:

```
C:\Path-To\compare.exe /max /wait /3 /title1:%tname /title2:%bname
```



```
/title3:%yname %theirs %base %mine %merged /a2
```

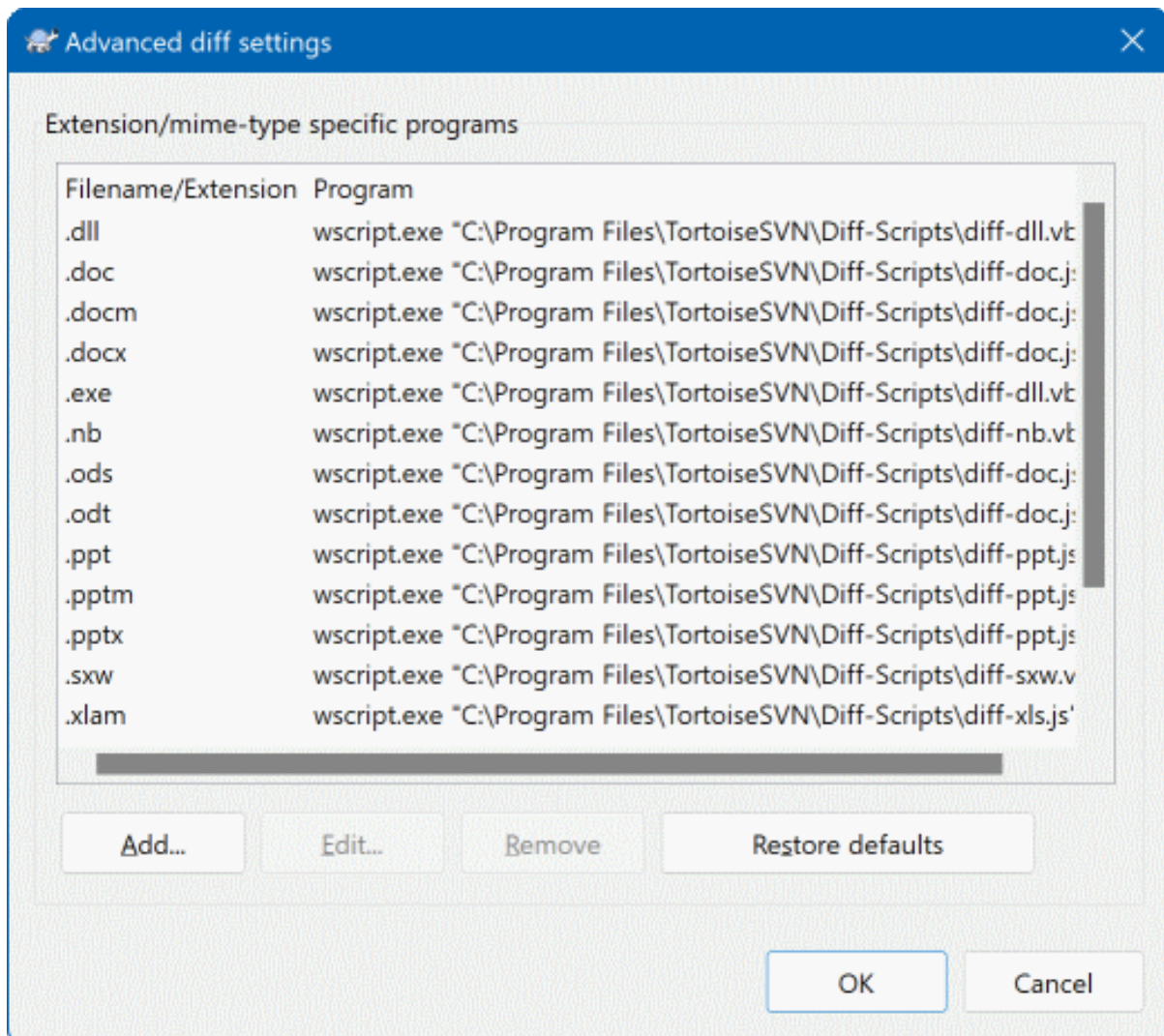
or with WinMerge (2.8 or later):

```
C:\Path-To\WinMerge.exe %merged
```

or with DiffMerge:

```
C:\Path-To\DiffMerge.exe -caption=%mname -result=%merged -merge
-nosplash -t1=%yname -t2=%bname -t3=%tname %mine %base %theirs
```

### 4.31.5.3. Verschil/Samenvoeg uitgebreide instellingen

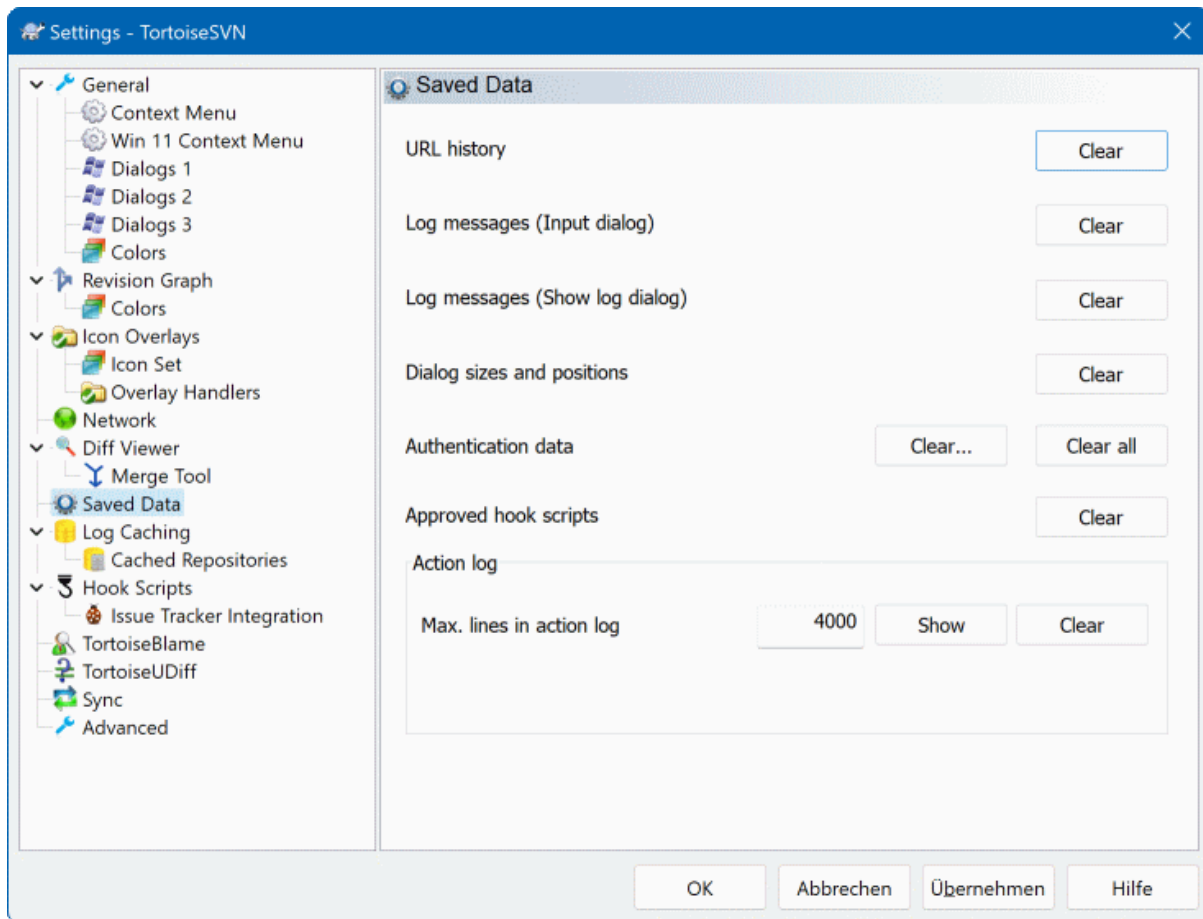


**Afbeelding 4.84. Het instellingen scherm, Verschil/Samenvoeg uitgebreid scherm**

In de uitgebreide instellingen kun u voor elke bestandsextensie een ander verschil-programma en samenvoeg-programma opgeven. U kunt bijvoorbeeld Photoshop associëren als het “verschil”-programma voor .jpg bestanden :-). U kunt ook de `svn:mime-type` eigenschap associëren met een verschil- of samenvoegprogramma.

Om middels een bestandsextensie te associëren dient u de extensie op te geven. Gebruik `.bmp` om Windows bitmap bestanden aan te geven. Om te associëren middels de `svn:mime-type` eigenschap, geef dan het mime-type (inclusief een slash) op. Bijvoorbeeld `text/xml`.

#### 4.31.6. Bewaarde Gegevens Instellingen



**Afbeelding 4.85. Het instellingen scherm, Opgeslagen gegevens pagina**

Voor uw gemak bewaart TortoiseSVN vele van de door u gebruikte eigenschappen en onthoudt waar u laatst bent geweest. Als u de cache van gegevens wilt schonen, kunt u dat hier doen.

##### URL geschiedenis

Telkens als u een werkkopie binnenhaalt, wijzigingen samenvoegt of de archief-verkenner gebruikt, houdt TortoiseSVN een lijst van de recent gebruikte URL's bij en stelt deze beschikbaar in een keuzelijst. Soms raakt die lijst vervuult met oude en ongeldige URL's, het is hierom raadzaam de lijst met regelmaat te legen.

Als u een enkel item uit een van de keuzelijsten wilt verwijderen kunt u dat ter plekke doen. Klik hiervoor op de pijl om de keuzelijst uit te klappen, beweeg de muis over het item dat u wilt verwijderen en gebruik de **Shift + Del** combinatie.

##### Recente boodschappen (Invoer dialoog)

TortoiseSVN bewaart de door u recent ingegeven logberichten bij vastleggen. Deze worden per archief opgeslagen, dus wanneer u veelarchieven benadert kan de lijst tot een behoorlijke grootte aangroeien.

##### Logboek boodschappen (Toon log dialoog)

TortoiseSVN houdt de uit het Toon log scherm verkregen logberichten bij in cache om tijd uit te sparen wanneer u de volgende keer de log bekijkt. Als iemand anders een logbericht aanpast terwijl u dat bericht al in de cache heeft staan, zal u die aanpassing niet zien totdat u de cache heeft geleegd. Het cachen van logberichten wordt aangezet op de Log cache tab.

### Venster afmetingen en posities

Veel vensters onthouden de afmetingen en vensterpositie die u het laatste heeft gebruikt.

### Authenticatie gegevens

Als u zich authenticceerd met een Subversion server worden de gebruikersnaam en wachtwoord lokaal in de cache geplaatst zodat u deze niet elke keer hoeft in te geven. U kunt ze willen verwijderen om veiligheids redenen of omdat u het archief onder een andere gebruikersnaam wilt benaderen ... weet Jan dat u zijn PC gebruikt ?

If you want to clear authentication data for one particular server only, use the **Clear...** instead of the **Clear all** button.

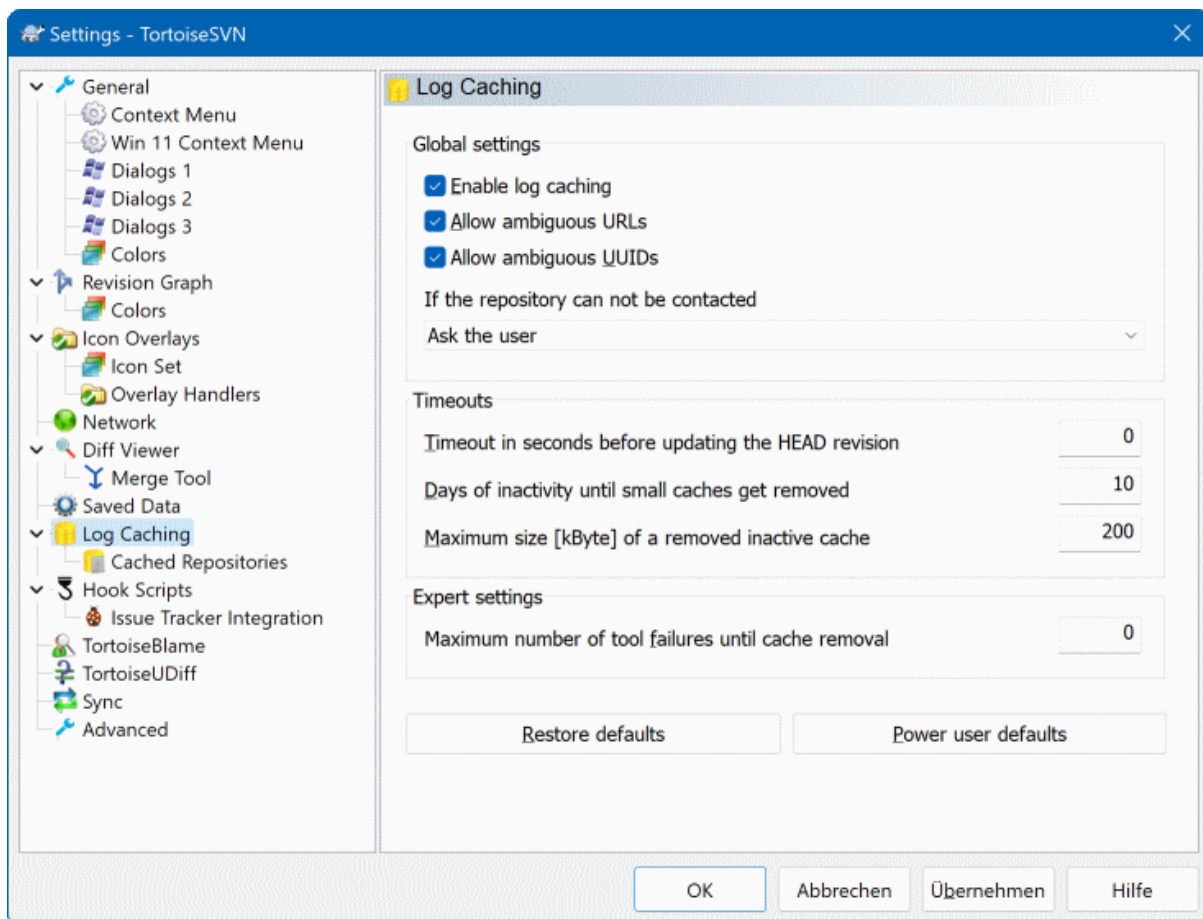
### Actie-log

TortoiseSVN houdt een log bij van alles wat naar het voortgangsscherm wordt geschreven. Dit kan handig zijn voor het geval dat u bijvoorbeeld wilt controleren wat er in een recent bijwerk-commando is gebeurd.

Het logbestand is in lengte begrensd en als het te groot wordt, worden de oudste gegevens verwijderd. Standaard worden 4000 regels bewaard, maar u kunt dit aantal zelf aanpassen.

Van hieruit kunt u de inhoud van het logbestand bekijken en ook wissen.

## 4.31.7. Log Caching



**Afbeelding 4.86. Het Instellingen Scherm, Log Cache Pagina**

In dit scherm kun je de log cache functionaliteit van TortoiseSVN instellen, waarmee een lokale kopie van logboek boodschappen en gewijzigde paden gemaakt wordt om lange wachttijden voor het ophalen van de gegevens van een

server te voorkomen. De log cache kan het gebruik van het logboek venster en het revisie schema sterk versnellen. Een andere nuttige toepassing is dat logboek boodschappen ook nog bekeken kunnen worden in offline mode.

#### Aanzetten van log caching

Maakt het gebruik van de log cache mogelijk als er logboek gegevens opgevraagd worden. Indien aangevinkt, worden berichten uit de cache gebruikt, als ze daarin beschikbaar zijn, en anders worden de berichten van de server gehaald en toegevoegd aan de cache.

Als caching uitgeschakeld is, dan worden de gegevens altijd direct vanaf de server opgehaald en wordt er niets lokaal opgeslagen.

#### Sta dubbelzinnige URLs toe

Occasionally you may have to connect to a server which uses the same URL for all repositories. Older versions of `svnbridge` would do this. If you need to access such repositories you will have to check this option. If you don't, unchecked it to improve performance.

#### Sta dubbelzinnige UUIDs toe

Some hosting services give all their repositories the same UUID. You may even have done this yourself by copying a repository folder to create a new one. For all sorts of reasons this is a bad idea - a UUID should be *unique*. However, the log cache will still work in this situation if you check this box. If you don't need it, unchecked it to improve performance.

#### Indien het archief niet bereikt kan worden

Als je offline werkt of als de server onbereikbaar is, dan kan de log cache nog steeds gebruikt worden voor het gebruiken van logboek boodschappen die al in de cache staan. Natuurlijk kan het zijn dat de cache niet de laatste stand van zaken weergeeft. Er zijn instellingen waarmee je in kunt stellen of deze functionaliteit gebruikt kan worden.

Als er logboek gegevens uit de cache worden gebruikt zonder contact te maken met de server, wordt in de titelbalk van het venster de offline status weergegeven.

#### Timeout in seconden voordat de HEAD revisie ververst wordt

Als je het logboek opent wil je eigenlijk ook dat er op de server gekeken wordt of er nieuwe logboek boodschappen zijn. Als de timeout hier op een niet-nul waarde is ingesteld, dan zal de server alleen gecontacteerd worden als de timeout tijd verstreken is sinds het laatste contact. Hiermee verlaag je de belasting van de server als je server langzaam is en je het logboek venster vaak opent, maar er wordt dus misschien niet altijd de laatste stand van zaken weergegeven. Als je deze functie wilt gebruiken, adviseren wij om een waarde van 300 (5 minuten) als compromis te gebruiken.

#### Dagen van inactiviteit totdat kleine caches verwijderd worden

Als je door veel verschillende archieven bladert, zal je veel log caches hebben. Als je deze niet actief gebruikt, zal de cache niet echt groot groeien. TortoiseSVN zal deze standaard na een bepaalde tijd wissen. Gebruik deze instelling om het wissen van cache te configureren.

#### Maximale grootte van een verwijderde inactieve cache

Grotere caches zijn duurder om opnieuw te vullen, daarom wist TortoiseSVN alleen kleinere caches. Je kunt de drempel met deze waarde verfijnen.

#### Maximum aantal fouten tot de cache verwijderd wordt

Soms gaat er weleens wat mis met de cache met een crash tot gevolg. Als dit gebeurt, wordt de cache normaliter automatisch verwijderd om herhaling van het probleem te voorkomen. Als je de minder stabiele nachtelijk versies gebruikt, kun je er voor kiezen om de cache toch te bewaren.

### 4.31.7.1. Opgeslagen (cache) Archieven

Op deze pagina zie je de lijst met archieven die in de lokale cache opgenomen zijn en de ruimte die gebruikt wordt voor de cache. Als je een van de archieven selecteert, kun je de volgende knoppen gebruiken.

Klik op de **Verversen** om de hele cache te verversen en de gaten op te vullen. Bij een groot archief kan dit veel tijd kosten, maar wel nuttig als je offline gaat en een goede cache beschikbaar wilt hebben.

Klik op de Exporteren knop om de gehele cache te exporteren naar een set van CSV bestanden. Dit kan handig zijn als je de logboek gegevens wilt bewerken met een extern programma, terwijl het hoofdzakelijk bedoeld is voor ontwikkelaars.

Klik op de Verwijderen om alle cache gegevens voor de geselecteerde archieven te verwijderen. Hiermee wordt de cache niet uitgeschakeld voor het archief. De volgende keer dat je logboek gegevens opvraagt, wordt er weer een nieuwe cache aangemaakt.

#### 4.31.7.2. Log Cache Statistieken

**Log Cache Statistics**

<b>Size and status</b>	<b>Age</b>
RAM [kB]: 861	last read: 8:01:01 PM
disk [kB]: 185	last update: 6:47:28 PM
connection: online	last HEAD update: 1/1/1970

<b>Container sizes</b>	
authors: 7	word tokens: 4502
path elements: 6785	pair tokens: 670
paths: 16032	text [token count]: 20175
skip ranges: 0	words [uncompressed]: 27050

<b>Revisions</b>	
max revision: 29337	merges total: 0
revision count: 2002	revisions: 0
	missing in: 2002
changes total: 31243	user revprops total: 0
revisions: 2002	revisions: 0
missing in: 0	missing in: 2002

OK

Afbeelding 4.87. Het Instellingen Scherm, Log Cache Statistieken

Klik op de **Details** knop om gedetailleerde statistieken te zien van een bepaalde cache. Veel van de hier getoonde velden zijn alleen interessant voor ontwikkelaars van TortoiseSVN, daarom zijn ze niet allemaal in detail beschreven.

#### RAM

De hoeveelheid geheugen die nodig is voor het gebruik van deze cache.

#### Schijf

De diskruimte die in gebruik is voor de cache. Gegevens worden gecomprimeerd, gebruik van ruimte is dus beperkt.

#### Verbinding

Geeft weer of het archief beschikbaar was op het moment dat de cache voor het laatst gebruikt werd.

#### Laatste verversing

De laatste keer dat de inhoud van de cache veranderd is.

#### Laatste hoofd verversing

De laatste keer dat de HEAD revisie is opgevraagd bij de server.

#### Auteurs

Het aantal verschillende auteurs waarvan berichten zijn opgeslagen in de cache.

#### Paden

Het aantal paden opgenomen in de lijst, zoals je die zou zien met `svn log -v`.

#### Overgeslagen bereiken

Het aantal revisie bereiken die we niet opgehaald hebben, omdat ze eenvoudigweg niet opgevraagd zijn. Dit is een maatstaf voor het aantal gaten in de cache.

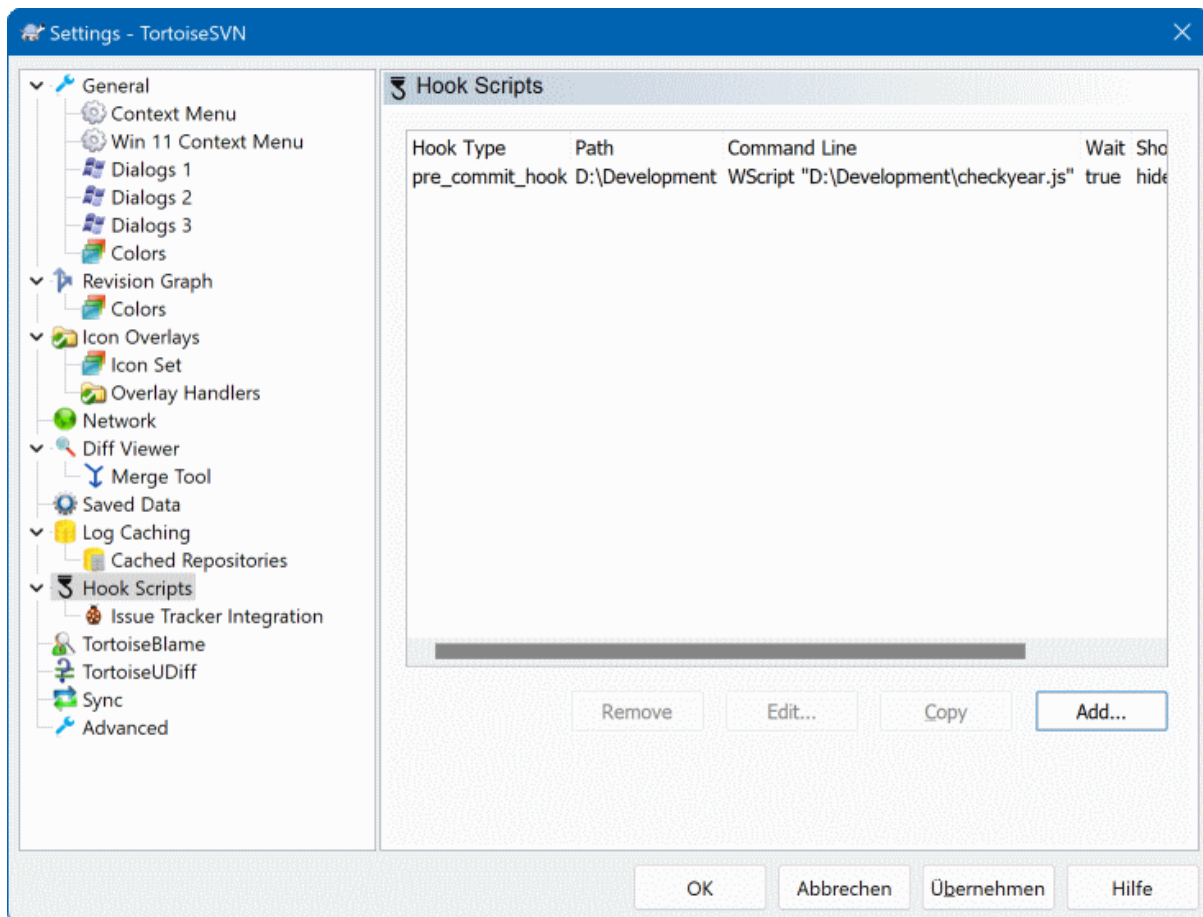
#### Maximale revisie

Het hoogste revisie nummer opgeslagen in de cache.

#### Revisie aantal

Het aantal revisies opgeslagen in de cache. Dit is een andere maatstaf voor de compleetheid van de cache.

### 4.31.8. Hook Scripts op de Client



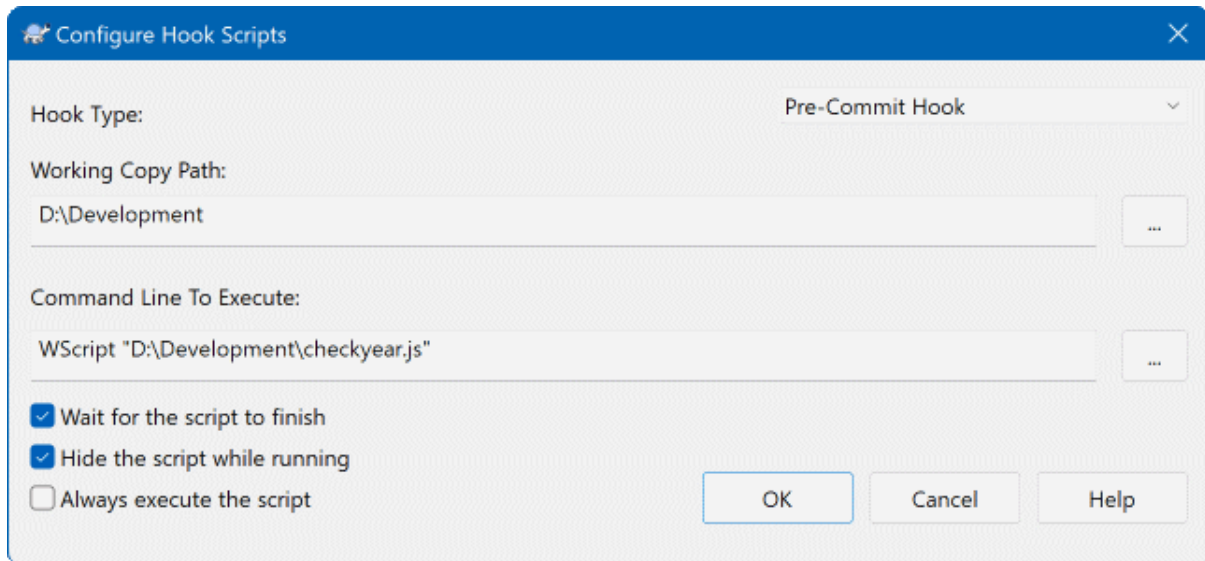
**Afbeelding 4.88. Het Instellingen Scherm, Hook Scripts Pagina**

In dit scherm kun je hook scripts instellen, welke automatisch uitgevoerd worden als een bepaalde Subversion actie plaatsvindt. In tegenstelling tot de hook scripts behandeld in [Paragraaf 3.3, “Server gesitueerde hook scripts”](#), worden deze scripts lokaal op de client uitgevoerd.

Een mogelijke toepassing voor zulke hooks kan zijn het aanroepen van een programma zoals `SubWCRev.exe` om de revisie nummers bij te kunnen werken na een vastlegging, en misschien ook om het bouwproces op te triggeren.

Note that you can also specify such hook scripts using special properties on your working copy. See the section [Paragraaf 4.18.2, “TortoiseSVN Project Eigenschappen”](#) for details.





**Afbeelding 4.89. Het Instellingen Scherm, Hook Scripts Configureren**

Klik op **Toevoegen** om een nieuw hook script toe te voegen en voer de details in.

There are currently these types of hook script available

#### Start-commit

Wordt aangeroepen voordat het vastleg venster getoond wordt. Je wilt deze mogelijk gebruiken als de hook een bestand wijzigt, wat gevolg kan hebben voor de lijst met bestanden die vastgelegd moeten worden en/of voor het vastleg bericht. Hou wel in de gaten dat, omdat deze hook vroeg in het proces wordt aangeroepen, nog niet de hele lijst met geselecteerde objecten voor de vastlegging beschikbaar is.

#### Manual Pre-commit

If this is specified, the commit dialog shows a button **Run Hook** which when clicked runs the specified hook script. The hook script receives a list of all checked files and folders and the commit message if there was one entered.

#### Check-commit

Called after the user clicks **OK** in the commit dialog, and before the commit dialog closes. This hook gets a list of all the checked files. If the hook returns an error, the commit dialog stays open.

If the returned error message contains paths on newline separated lines, those paths will get selected in the commit dialog after the error message is shown.

#### Pre-commit

Wordt aangeroepen nadat de gebruiker op de **OK** klikt in het vastleg venster en nog voordat de echte vastlegging start. Deze hook heeft een lijst met wat er precies vastgelegd gaat worden.

#### Post-commit

Called after the commit finishes successfully.

#### Start-update

Wordt aangeroepen voordat het verversen-naar-revisie scherm weergegeven wordt.

#### Pre-update

Called before the actual Subversion update or switch begins.

#### Post-update

Called after the update, switch or checkout finishes (whether successful or not).



**Pre-verbinden**

Called before an attempt to contact the repository. Called at most once in five minutes.

**Pre-lock**

Called before an attempt to lock a file.

**Post-lock**

Called after a file has been locked.

Een hook wordt gedefinieerd voor een bepaalde werkkopie. Je hoeft alleen het pad van de basismap op te geven; als je een actie uitvoert in een sub-map, dan zoekt TortoiseSVN automatisch in de bovenliggende mappen naar het bijbehorende pad.

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, e.g. a perl script. Note that the script must not be specified using a UNC path as Windows shell execute will not allow such scripts to run due to security restrictions.

De opdrachtregel bevat verschillende parameters, die door TortoiseSVN ingevuld worden. Welke parameters doorgegeven worden hangt af van de aangeroepen hook. Elke hook heeft zijn eigen parameters, welke doorgegeven worden in de volgende volgorde:

**Start-commit**

PATHMESSAGEFILECWD

**Manual Pre-commit**

PATHMESSAGEFILECWD

**Check-commit**

PATHMESSAGEFILECWD

**Pre-commit**

PATHDEPTHMESSAGEFILECWD

**Post-commit**

PATHDEPTHMESSAGEFILEREVISIONERRORCWD

**Start-update**

PATHCWD

**Pre-update**

PATHDEPTHREVISIONCWD

**Post-update**

PATHDEPTHREVISIONERRORCWDRESULTPATH

**Pre-verbinden**

no parameters are passed to this script. You can pass a custom parameter by appending it to the script path.

**Pre-lock**

PATHLOCKFORCEMESSAGEFILECWD

**Post-lock**

PATHLOCKFORCEMESSAGEFILEERRORCWD

De betekenis van elke parameter is hier beschreven:

**PATH**

A path to a temporary file which contains all the paths for which the operation was started in UTF-8 encoding. Each path is on a separate line in the temp file.

Note that for operations done remotely, e.g. in the repository browser, those paths are not local paths but the urls of the affected items.

#### DEPTH

De diepte waarmee de vastlegging/verversing uitgevoerd wordt.

Mogelijke waarden zijn:

- 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

Pad naar een bestand met het logboek bericht van de vastlegging. Het bestand bevat de tekst met een UTF-8 codering. Na succesvolle uitvoering van de start-vastlegging hook, wordt het logboek bericht teruggelezen, waardoor de hook de kans krijgt het te wijzigen.

#### REVISION

De revisie van het archief naar welke de verversing uitgevoerd zal moeten worden of nadat een vastlegging gereed is.

#### LOCK

Either `true` when locking, or `false` when unlocking.

#### FORCE

Either `true` or `false`, depending on whether the operation was forced or not.

#### ERROR

Pad naar een bestand met daarin de foutmelding. Als er geen fout was, dan zal het bestand leeg zijn.

#### CWD

De huidige werk map waarmee het script gedraaid wordt. Deze is ingesteld op de gemeenschappelijke hoofdmap voor alle betrokken paden.

#### RESULTPATH

A path to a temporary file which contains all the paths in UTF-8 encoding which were somehow touched by the operation. Each path is on a separate line in the temp file.

Merk op dat ondanks dat we deze parameters namen hebben gegeven voor het gemak, je niet naar deze namen hoeft te refereren in de hook instellingen. Alle parameters die beschikbaar zijn voor een bepaalde hook worden altijd doorgegeven, of je ze nu wilt hebben of niet ;-)

Als je wilt dat het Subversion proces wacht totdat de hook klaar is, selecteer dan **Wacht op het beëindigen van het script**.

Normally you will want to hide ugly DOS boxes when the script runs, so **Hide the script while running** is checked by default. Also you need to check this if your hook script might return an error that should stop the operation.

The `force` flag can be set if the user must not proceed with the operation without running the script, i.e. the script must always run. If the `force` flag is not checked, then the user is shown a button `Retry without hooks` to retry the operation without running the hook script.

Sample client hook scripts can be found in the `contrib` folder in the *TortoiseSVN repository* [<https://svn.code.sf.net/p/tortoisesvn/code/trunk/contrib/hook-scripts>]. (Paragraaf 3, “Licentie” explains how to access the repository.)

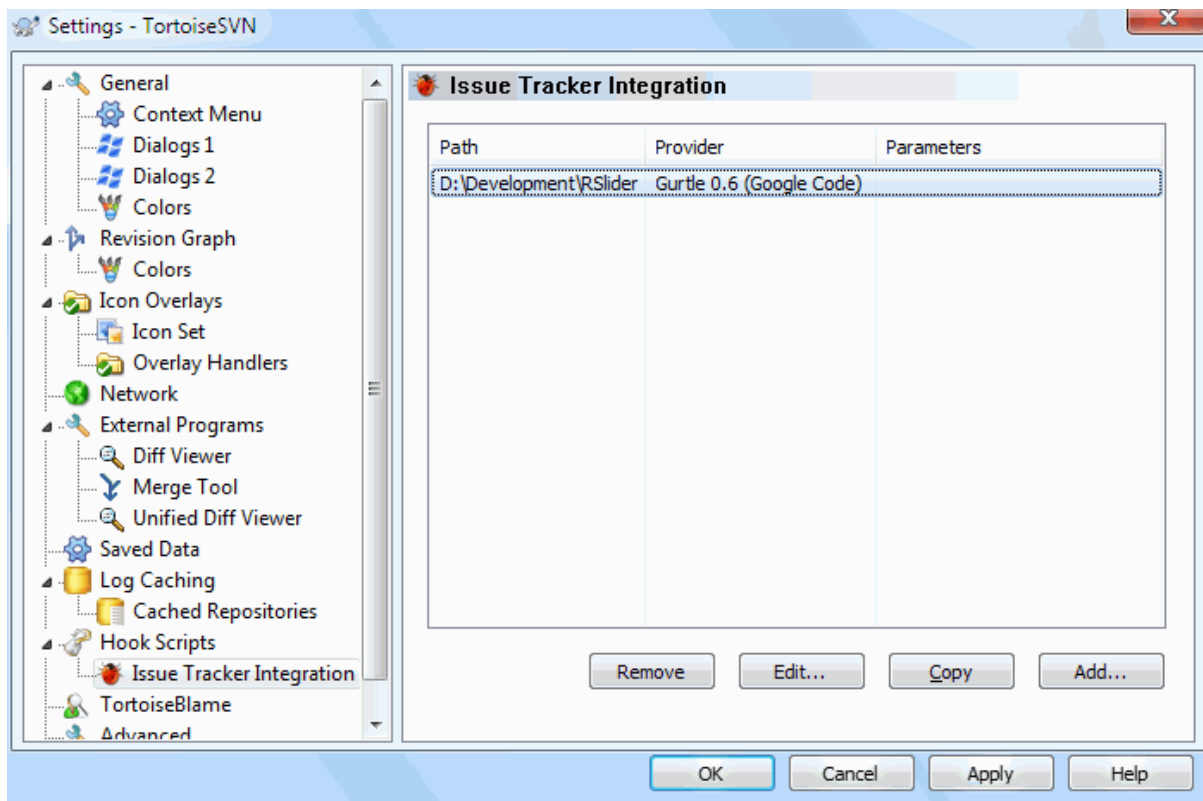
When debugging hook scripts you may want to echo progress lines to the DOS console, or insert a pause to stop the console window disappearing when the script completes. Because I/O is redirected this will not normally work. However you can redirect input and output explicitly to CON to overcome this. e.g.

```
echo Checking Status > con
pause < con > con
```

A small tool is included in the TortoiseSVN installation folder named `ConnectVPN.exe`. You can use this tool configured as a pre-connect hook to connect automatically to your VPN before TortoiseSVN tries to connect to a repository. Just pass the name of the VPN connection as the first parameter to the tool.

#### 4.31.8.1. Issue Tracker Integratie

TortoiseSVN kan gebruik maken van een COM plugin om te communiceren met issue trackers vanuit het vastleg scherm. Het gebruik van plugins is beschreven in Paragraaf 4.29.2, “Informatie Verkrijgen vanuit de Issue Tracker”. Als jouw systeembeheerder een plugin beschikbaar heeft gesteld, die je al geïnstalleerd en geregistreerd hebt, dan kun je hier specificeren hoe deze moet integreren met je werkkopie.



**Afbeelding 4.90. Het Instellingen Scherm, Issue Tracker Integratie Pagina**

Klik op de `Toevoegen...` om de plugin met een bepaalde werkkopie te gebruiken. Je kunt hier het pad van de werkkopie specificeren, de kiezen welke plugin gebruikt moet worden uit een drop down lijst met alle geregistreerde issue tracker plugins en de parameters die meegegeven moeten worden. De parameters zijn specifiek

voor een plugin, maar kunnen je gebruikersnaam voor je issue tracker bevatten, zodat de plugin problemen kan vinden dan aan jou zijn toegekend.

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`, `bugtraq:provideruuid64` and `bugtraq:providerparams`.

`bugtraq:provideruuid`

This property specifies the COM UUID of the IBugtraqProvider, for example {91974081-2DC7-4FB1-B3BE-0DE1C8D6CE4E}. (This example is the UUID of the *Gurtle bugtraq provider* [<http://code.google.com/p/gurtle/>], which is a provider for the *Google Code* [<http://code.google.com/hosting/>] issue tracker.)

`bugtraq:provideruuid64`

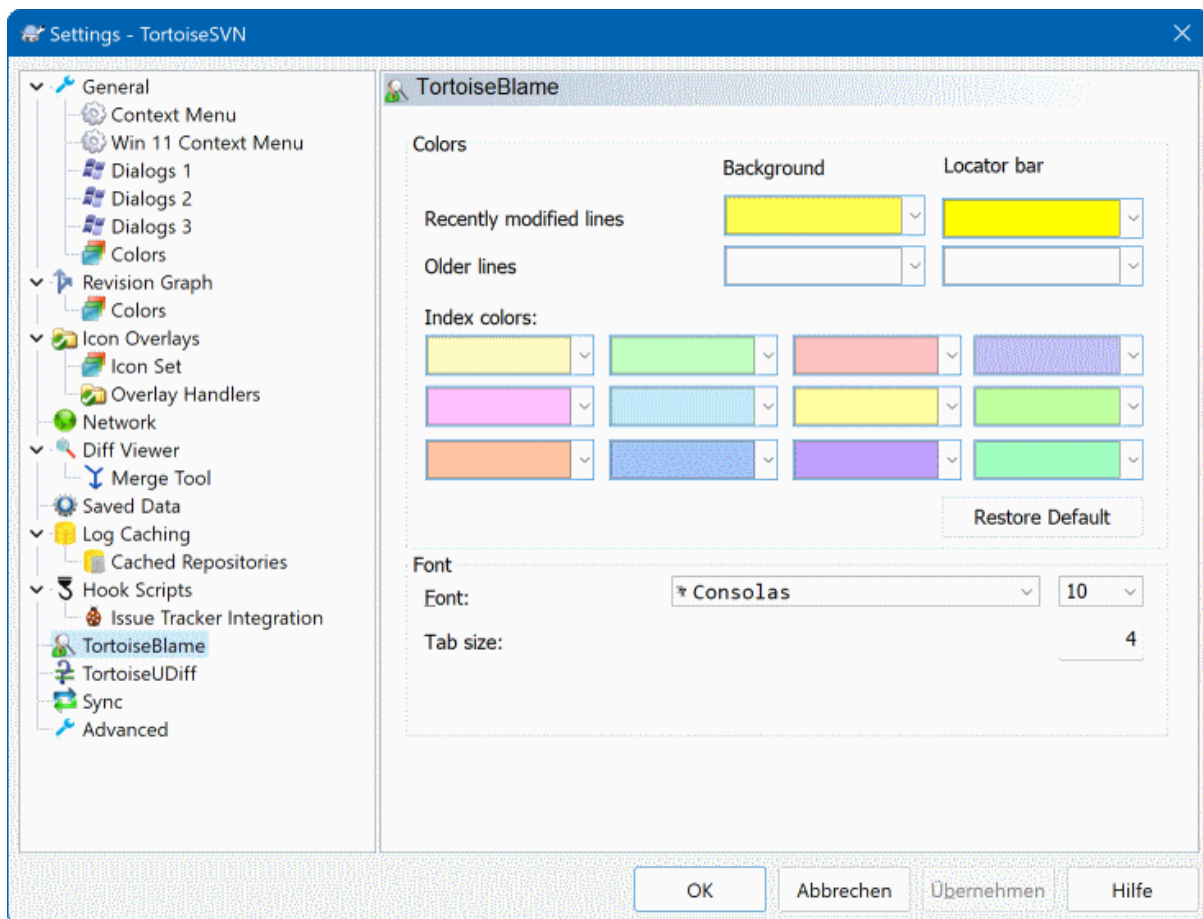
This is the same as `bugtraq:provideruuid`, but for the 64-bit version of the IBugtraqProvider.

`bugtraq:providerparams`

Deze eigenschap de eigenschappen die doorgegeven moeten worden aan de IBugtraqProvider.

Please check the documentation of your IBugtraqProvider plugin to find out what to specify in these two properties.

### 4.31.9. TortoiseBlame Instellingen



**Afbeelding 4.91. Het Instellingen Scherm, TortoiseBlame pagina**

De instellingen voor TortoiseBlame worden gedaan vanuit het hoofd context menu, niet direct in TortoiseBlame zelf.

**Kleuren**

TortoiseBlame kan gebruik maken van achtergrond kleur om de leeftijd van de regels in een bestand aan te geven. Je kunt de eindpunten aangeven door de kleuren voor de nieuwste en oudste revisies te kiezen.

TortoiseBlame zal een lineaire interpolatie gebruiken tussen deze twee kleuren volgens de revisie van het archief voor elke regel.

You can specify different colours to use for the locator bar. The default is to use strong contrast on the locator bar while keeping the main window background light so that you can still read the text.

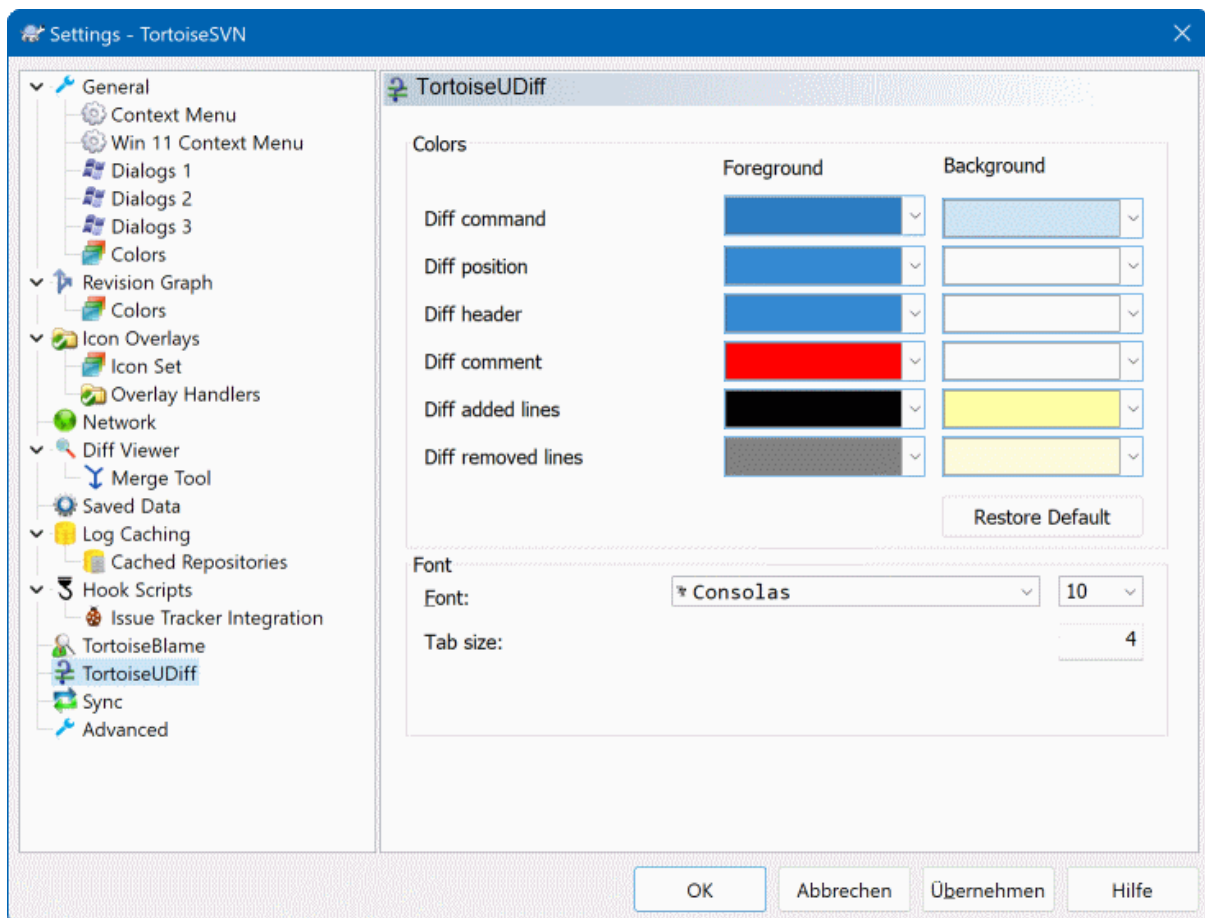
#### Lettertype

Je kunt het lettertype en grootte selecteren voor tekst die weergegeven wordt. Deze instelling wordt gebruikt voor inhoud van een bestand en de auteur en revisie informatie in het linker deelvenster.

#### Tabs

Definieert hoeveel spaties gebruikt moeten worden als een tab karakter gevonden is in de inhoud van een bestand.

### 4.31.10. TortoiseUDiff Settings



#### Afbeelding 4.92. The Settings Dialog, TortoiseUDiff Page

The settings used by TortoiseUDiff are controlled from the main context menu, not directly with TortoiseUDiff itself.

#### Kleuren

The default colors used by TortoiseUDiff are usually ok, but you can configure them here.

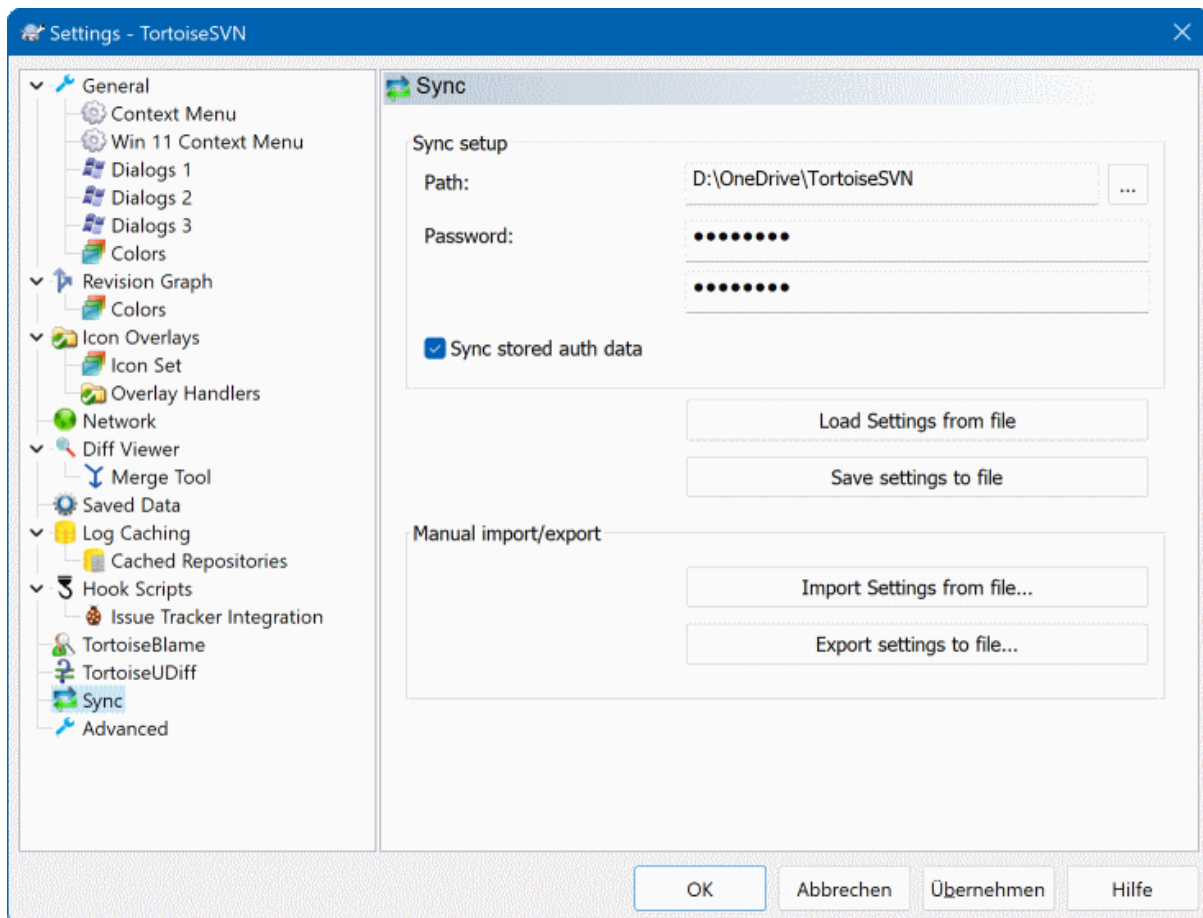
#### Lettertype

You can select the font used to display the text, and the point size to use.

#### Tabs

Defines how many spaces to use for expansion when a tab character is found in the file diff.

### 4.31.11. Exporting TSVN Settings



#### Afbeelding 4.93. The Settings Dialog, Sync Page

You can sync all TortoiseSVN settings to and from an encrypted file. The file is encrypted with the password you enter so you don't have to worry if you store that file on a cloud folder like OneDrive, GDrive, DropBox, ...

When a path and password is specified, TortoiseSVN will sync all settings automatically and keep them in sync.

You can also export/import an encrypted files with all the settings manually. When you do that, you're asked for the path of the file and the password to encrypt/decrypt the settings file.

When exporting the settings manually, you can also optionally include all local settings which are not included in a normal export or in a sync. Local settings are settings which include local paths which usually vary between computers. These local settings include the configured diff and merge tools and hook scripts.

### 4.31.12. Geavanceerde Instellingen

Enkele niet vaak gebruikte instellingen zijn alleen beschikbaar in de geavanceerde pagina van het instellingen scherm. Deze instellingen wijzigen het register en je moet weten waarvoor elke instelling gebruikt wordt en wat die doet. Wijzig deze instellingen niet, tenzij je zeker weet dat je ze wijzigen moet.

#### AllowAuthSave

Sometimes multiple users use the same account on the same computer. In such situations it's not really wanted to save the authentication data. Setting this value to `false` disables the `save authentication` button in the authentication dialog.

#### AllowUnversionedObstruction

Als een verversing aan een werkkopie een nieuw bestand uit het archief toevoegt, van welke al een kopie aanwezig is in de werkkopie als bestand zonder versiebeheer, dan wordt standaard het lokale bestand behouden

en aangegeven als (mogelijk) gewijzigd bestand. Als je prefereert dat TortoiseSVN hiervoor het bestand als conflict markeert in zulke situaties, stel deze waarde dan in op `false`.

#### AlwaysExtendedMenu

Net zoals de verkenner, toont TortoiseSVN extra commando's als de **Shift** knop is ingedrukt als het context menu geopend is. Om TortoiseSVN altijd die uitgebreide commando's te laten zien, stel je deze waarde in op `true`.

#### AutoCompleteMinChars

The minimum amount of chars from which the editor shows an auto-completion popup. The default value is 3.

#### AutocompleteRemovesExtensions

De auto-aanvul lijst die weergegeven wordt in het vastleg bericht scherm, toont de namen van bestanden die meegenomen worden bij de vastlegging. Om deze name ook mee te nemen zonder de extensies, stel je deze waarde in op `true`.

#### BlockPeggedExternals

File externals that are pegged to a specific revision are blocked by default from being selected for a commit. This is because a subsequent update would revert those changes again unless the pegged revision of the external is adjusted.

Set this value to `false` in case you still want to commit changes to such external files.

#### BlockStatus

Als je niet wilt dat de verkenner de overlappende status pictogrammen bijwerkt terwijl er een TortoiseSVN commando loopt (bijv. Verversen, Vastleggen, ...), stel deze waarde dan in op `true`.

#### CacheTrayIcon

Stel deze waarde in op `true` om een cache taakbalk pictogram toe te laten voegen voor het TSVNCache programma. Dit wordt echt alleen maar door ontwikkelaars gebruikt omdat het de mogelijkheid geeft het programma netjes af te sluiten.

#### ColumnsEveryWhere

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, set this value to `true`. Note that the extra columns are only available in XP. Vista and later doesn't support that feature any more. However some third-party explorer replacements do support those even on Windows versions later than XP.

#### ConfigDir

Hier kun je een andere map specificeren voor het Subversion configuratiebestand. Dit heeft effect op alle TortoiseSVN acties.

#### CtrlEnter

In de meeste schermen van TortoiseSVN kun je **Ctrl+Enter** gebruiken om het scherm te sluiten alsof je op de OK knop gedrukt hebt. Als je dit niet wilt, stel deze waarde dan in op `false`.

#### Debug

Set this to `true` if you want a dialog to pop up for every command showing the command line used to start TortoiseProc.exe.

#### DebugOutputString

Set this to `true` if you want TortoiseSVN to print out debug messages during execution. The messages can be captured with special debugging tools only.

#### DialogTitles

The default format (value of 0) of dialog titles is `url/path - name of dialog - TortoiseSVN`. If you set this value to 1, the format changes to `name of dialog - url/path - TortoiseSVN`.

### DiffBlamesWithTortoiseMerge

TortoiseSVN geeft de mogelijkheid om een extern diff programma in te stellen. De meeste programma's zijn echter niet geschikt voor het verklaren van verschillen (Paragraaf 4.24.2, “Verklaar Verschillen”), waardoor je misschien terug wilt vallen op TortoiseMerge in die gevallen. Stel hiervoor deze waarde in op `true`

### DlgStickySize

This value specifies the number of pixels a dialog has to be near a border before the dialog sticks to it. The default value is 3. To disable this value set the value to zero.

### FixCaseRenames

Some apps change the case of filenames without notice but those changes aren't really necessary nor wanted. For example a change from `file.txt` to `FILE.TXT` wouldn't bother normal Windows applications, but Subversion is case sensitive in these situations. So TortoiseSVN automatically fixes such case changes.

If you don't want TortoiseSVN to automatically fix such case changes for you, you can set this value to `false`.

### FullRowSelect

De besturing van de status lijst, welke in verschillende schermen wordt gebruikt (bijv. vastleggen, kijk-of-er-updates-zijn, toevoegen, ongedaan maken, ...), selecteert altijd hele rijen (dat wil zeggen, als je een entry selecteert de hele rij geselecteerd wordt, niet alleen de eerste kolom). Op zich is dat niet erg, maar de geselecteerde rij bedekt dan ook het achtergrond plaatje aan de rechter onderzijde. Dit kan er lelijk uitzien. Om de selectie van de gehele rij uit te schakelen, stel je deze waarde in op `false`

### GroupTaskbarIconsPerRepo

This option determines how the Win7 taskbar icons of the various TortoiseSVN dialogs and windows are grouped together. This option has no effect on Vista!

1. The default value is 0. With this setting, the icons are grouped together by application type. All dialogs from TortoiseSVN are grouped together, all windows from TortoiseMerge are grouped together, ...



**Afbeelding 4.94. Taskbar with default grouping**

2. If set to 1, then instead of all dialogs in one single group per application, they're grouped together by repository. For example, if you have a log dialog and a commit dialog open for repository A, and a check-for-modifications dialog and a log dialog for repository B, then there are two application icon groups shown in the Win7 taskbar, one group for each repository. But TortoiseMerge windows are not grouped together with TortoiseSVN dialogs.



**Afbeelding 4.95. Taskbar with repository grouping**

3. If set to 2, then the grouping works as with the setting set to 1, except that TortoiseSVN, TortoiseMerge, TortoiseBlame, TortoiseIDiff and TortoiseUDiff windows are all grouped together. For example, if you have the commit dialog open and then double click on a modified file, the opened TortoiseMerge diff window will be put in the same icon group on the taskbar as the commit dialog icon.





**Afbeelding 4.96. Taskbar with repository grouping**

4. If set to 3, then the grouping works as with the setting set to 1, but the grouping isn't done according to the repository but according to the working copy. This is useful if you have all your projects in the same repository but different working copies for each project.
5. If set to 4, then the grouping works as with the setting set to 2, but the grouping isn't done according to the repository but according to the working copy.

#### HideExternalInfo

If this is set to `false`, then every `svn:externals` is shown during an update separately.

If it is set to `true` (the default), then update information for externals is only shown if the externals are affected by the update, i.e. changed in some way. Otherwise nothing is shown as with normal files and folders.

#### GroupTaskbarIconsPerRepoOverlay

This has no effect if the option `GroupTaskbarIconsPerRepo` is set to 0 (see above).

If this option is set to `true`, then every icon on the Win7 taskbar shows a small colored rectangle overlay, indicating the repository the dialogs/windows are used for.



**Afbeelding 4.97. Taskbar grouping with repository color overlays**

#### IncludeExternals

Standaard neemt TortoiseSVN ook altijd de externen mee bij het verversen. Dit voorkomt consistentie problemen met werkkopieën. Als je echter veel externen hebt ingesteld, kan een verversing erg lang duren. Stel deze waarde in op `false` om standaard bij de verversing de externen niet mee te nemen. Om dan externen toch te verversen gebruik je `Verversen naar revisie...` of stel deze waarde weer in op `true`.

#### LogFindCopyFrom

When the log dialog is started from the merge wizard, already merged revisions are shown in gray, but revisions beyond the point where the branch was created are also shown. These revisions are shown in black because those can't be merged.

If this option is set to `true` then TortoiseSVN tries to find the revision where the branch was created from and hide all the revisions that are beyond that revision. Since this can take quite a while, this option is disabled by default. Also this option doesn't work with some SVN servers (e.g., Google Code Hosting, see [issue #5471](http://code.google.com/p/support/issues/detail?id=5471) [<http://code.google.com/p/support/issues/detail?id=5471>]).

#### LogMultiRevFormat

A format string for the log messages when multiple revisions are selected in the log dialog.

You can use the following placeholders in your format string:

`%1!ld!`  
gets replaced with the revision number text

`%2!s!`  
gets replaced with the short log message of the revision

#### LogStatusCheck

Het logboek venster toont de revisie van het pad van de werkkopie vetgedrukt, maar hiervoor moet het logboek venster de status van het pad van de werkkopie ophalen. Omdat dit voor grote werkkopieën lang kan duren, kun je deze waarde op `false` instellen, zodat deze functie uitgeschakeld wordt.

#### MaxHistoryComboItems

Comboboxes for URLs and paths show a history of previously used URLs/paths if possible. This settings controls how many previous items are saved and shown. The default is 25 items.

#### MergeLogSeparator

Als je revisies van een andere tak samenvoegt en samenvoeg traceer informatie is beschikbaar, dan worden de logboek berichten van de samen te voegen revisies opgehaald om een nieuw vastleg bericht te maken. Een vooraf gedefinieerde tekst wordt gebruikt om de individuele berichten van de revisies te scheiden. Je kunt deze waarde aanpassen naar een tekst of indeling die je zelf wilt.

#### NumDiffWarning

If you want to show the diff at once for more items than specified with this settings, a warning dialog is shown first. The default is 10.

#### OldVersionCheck

TortoiseSVN controleert één keer per week of er een nieuwe versie beschikbaar is. Als een nieuwe versie gevonden is, dan wordt dit in het vastleg scherm aangegeven. Als je het oude gedrag terug wilt, waarbij er een apart scherm wordt weergegeven, stel deze waarde dan in op `true`.

#### RepoBrowserTrySVNParentPath

The repository browser tries to fetch the web page that's generated by an SVN server configured with the `SVNParentPath` directive to get a list of all repositories. To disable that behavior, set this value to `false`.

#### ScintillaBidirectional

This option enables the bidirectional mode for the commit message edit box. If enabled, right-to-left language text editing is done properly. Since this feature is expensive, it is disabled by default. You can enable this by setting this value to `true`.

#### ScintillaDirect2D

This option enables the use of Direct2D accelerated drawing in the Scintilla control which is used as the edit box in e.g. the commit dialog, and also for the unified diff viewer. With some graphic cards however this sometimes doesn't work properly so that the cursor to enter text isn't always visible. If that happens, you can turn this feature off by setting this value to `false`.

#### OutOfDateRetry

This parameter specifies how TortoiseSVN behaves if a commit fails due to an out-of-date error:

- 0  
The user is asked whether to update the working copy or not, and the commit dialog is not reopened after the update.
- 1  
This is the default. The user is asked whether to update the working copy or not, and the commit dialog is reopened after the update so the user can proceed with the commit right away.
- 2  
Similar to 1, but instead of updating only the paths selected for a commit, the update is done on the working copy root. This helps to avoid inconsistent working copies.

3

The user is not asked to update the working copy. The commit simply fails with the out-of-date error message.

#### PlaySound

If set to `true`, TortoiseSVN will play a system sound when an error or warning occurs, or another situation which is important and requires your attention. Set this to `false` if you want to keep TortoiseSVN quiet. Note that the project monitor has its own setting for playing sounds, which you can configure in its settings dialog.

#### ShellMenuAccelerators

TortoiseSVN gebruikt sneltoetsen voor de opties in de context menu's. Dit kan leiden tot dubbele toewijzingen van sneltoetsen (bijv. de `SVN Vastleggen` het de **Alt-C** sneltoets, maar de `Kopieer` optie van de verkenner gebruikt die ook). Als je de sneltoetsen niet wilt gebruiken of nodig hebt voor de TortoiseSVN opties, stel deze waarde dan in op `false`.

#### ShowContextMenuIcons

Dit kan nuttig zijn als je iets anders gebruikt dan de Windows verkenner of als je problemen hebt met verkeerd weergegeven context menu's. Stel deze waarde in op `false` als je wilt dat TortoiseSVN geen pictogrammen weergeeft in het context menu. Stel deze waarde in op `true` om de pictogrammen weer te laten zien.

#### ShowAppContextMenuIcons

Als je niet wilt dat TortoiseSVN pictogrammen laat zien voor de context menu's in zijn eigen schermen, stel deze waarde dan in op `false`.

#### ShowNotifications

Set this value to `false` if you don't want the project monitor to show notification popups when new commits are detected.

#### StyleCommitMessages

Het vastleg en logboek venster gebruiken stijlen (bijv. vet, schuingedrukt) bij vastleg berichten (lees [Paragraaf 4.4.5, "Vastleg Log Meldingen"](#) voor meer details). Als je dit niet wilt hebben, stel deze waarde dan in op `false`.

#### UpdateCheckURL

Deze waarde bevat de URL waar vandaan TortoiseSVN een tekstbestand probeert op te halen om te achterhalen of er nieuwe versies beschikbaar zijn. Dit kan handig zijn voor beheerders van bedrijven, die willen voorkomen dat gebruikers hun TortoiseSVN bijwerken voordat zij dat geaccepteerd hebben.

#### UseCustomWordBreak

The standard edit controls do not stop on forward slashes like they're found in paths and urls. TortoiseSVN uses a custom word break procedure for the edit controls. If you don't want that and use the default instead, set this value to 0. If you only want the default for edit controls in combo boxes, set this value to 1.

#### VersionCheck

TortoiseSVN controleert één keer per week of er een nieuwe versie beschikbaar is. Als je niet wilt dat TortoiseSVN deze controle uitvoert, stel deze waarde dan in op `false`.

## 4.32. Laatste stap

### Doneer!

Even though TortoiseSVN and TortoiseMerge are free, you can support the developers by sending in patches and playing an active role in the development. You can also help to cheer us up during the endless hours we spend in front of our computers.

While working on TortoiseSVN we love to listen to music. And since we spend many hours on the project we need a *lot* of music. Therefore we have set up some wish-lists with our favourite music CDs and DVDs: <https://tortoisesvn.net/donate.html> Please also have a look at the list of people who contributed to the project by sending in patches or translations.

# Hoofdstuk 5. Project Monitor

The project monitor is a helpful tool that monitors repositories and notifies you in case there are new commits.

The projects can be monitored via a working copy path or directly via their repository URLs.

The project monitor scans each project in a configurable interval, and every time new commits are detected a notification popup is shown. Also the icon that is added to the system tray changes to indicate that there are new commits.

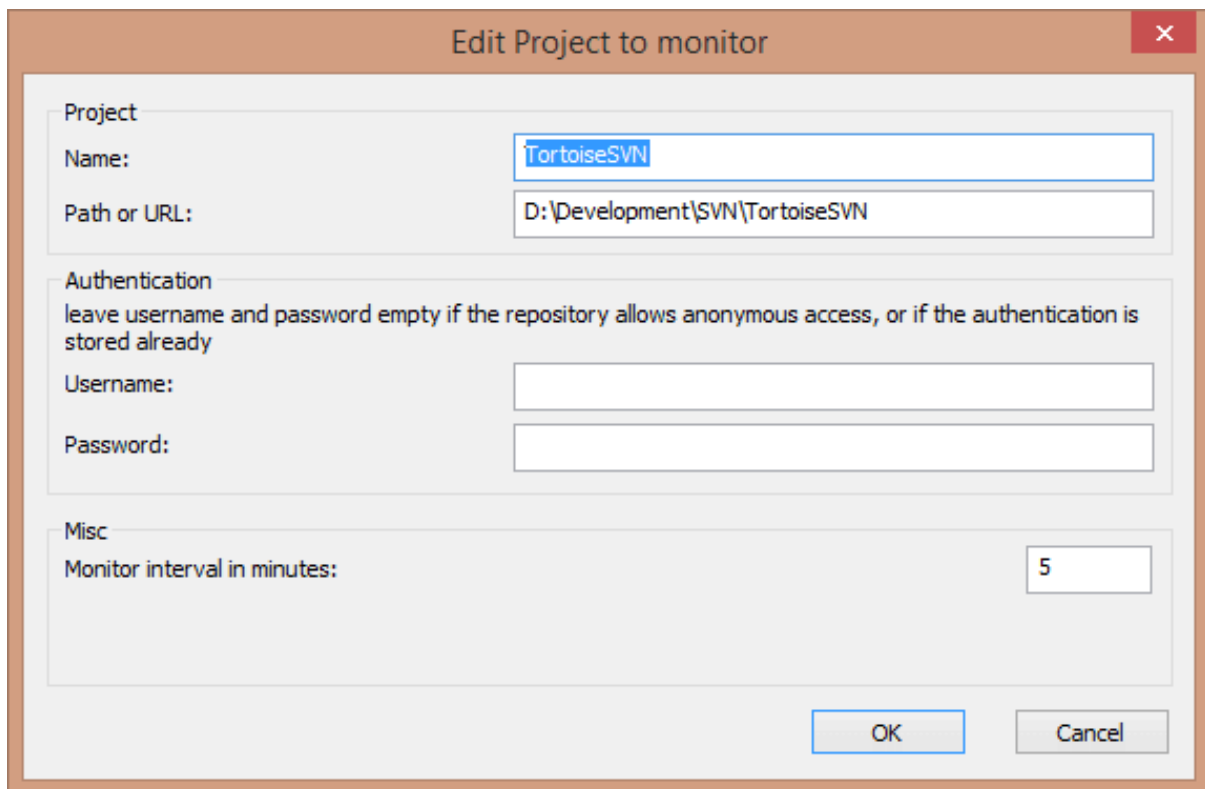


## Snarl

If *Snarl* [<https://snarl.fullphat.net/>] is installed and active, then the project monitor automatically uses Snarl to show the notifications about newly detected commits.

## 5.1. Adding projects to monitor

If you first start the project monitor, the tree view on the left side is empty. To add projects, click on the button at the top of the dialog named Add Project.



### Afbeelding 5.1. The edit project dialog of the project monitor

To add a project for monitoring, fill in the required information. The name of the project is not optional and must be filled in, all other information is optional.

If the box for `Path or URL` is left empty, then a folder is added. This is useful to group monitored projects.

If you want to monitor all repositories served via the *SVNParentPath* [<http://svnbook.red-bean.com/en/1.8/svn.serverconfig.httpd.html#svn.serverconfig.httpd.basic>] directive, enter the root `Url` for your repositories and check the box `Url points to SVNParentPath list`.

The fields `Username` and `Password` should only be filled in if the repository does not provide anonymous read access, and only if the authentication is not stored by Subversion itself. If you're accessing the monitored repository with TortoiseSVN or other svn clients and you've stored the authentication already, you should leave this empty: you won't have to edit those projects manually if the password changes.

The `Monitor interval in minutes` specifies the minutes to wait in between checks. The smallest interval is one minute.

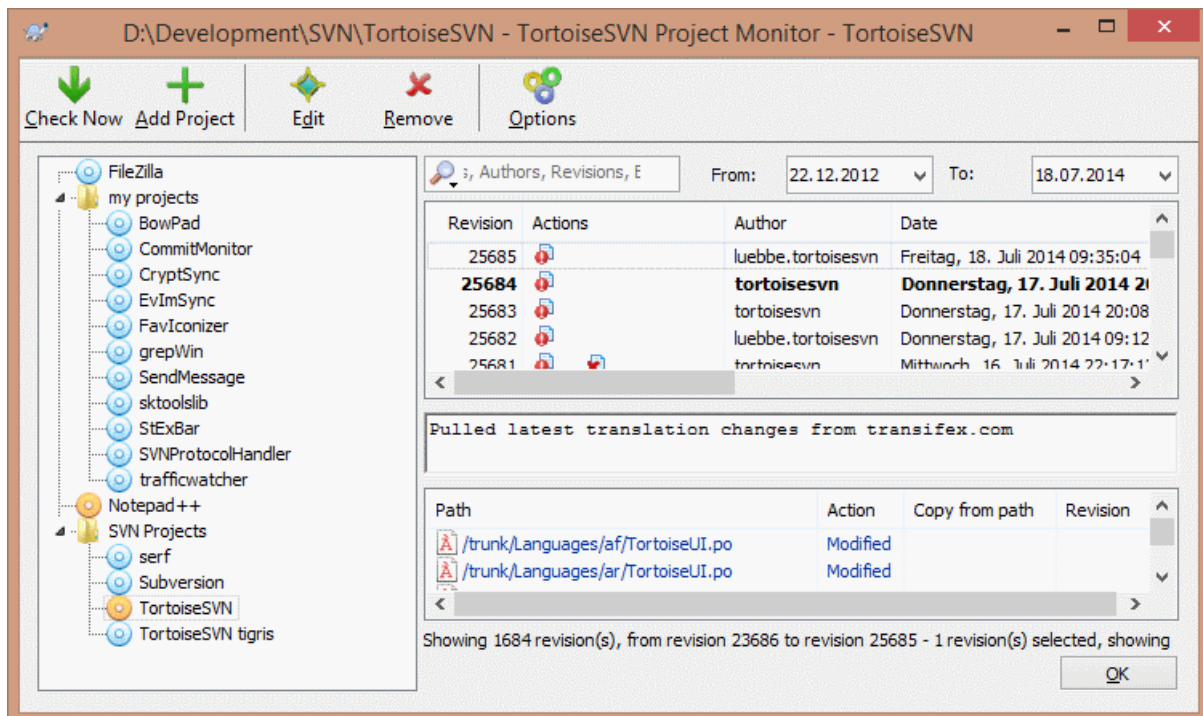


## check interval

If there are a lot of users monitoring the same repository and the bandwidth on the server is limited, a repository admin can set the minimum for check intervals using an `svnrobots.txt` file. A detailed explanation on how this works can be found on the project monitor website:

<https://tools.stefankueng.com/svnrobots.html> [https://tools.stefankueng.com/svnrobots.html]

## 5.2. Monitor dialog



**Afbeelding 5.2. The main dialog of the project monitor**

The project monitor shows all monitored projects on the left in a tree view. The projects can be moved around, for example one project can be moved below another project, making it a child/subproject.

A click on a project shows all the log messages of that project on the right.

Projects that have updates are shown in bold, with the number of new commits in brackets at the right. A click on a project marks it automatically as read.

### 5.2.1. Main operations

The toolbar at the top of the dialog allows to configure and operate the project monitor.

Controleer nu

While each monitored project is checked according to the interval that's set up, clicking this button will force a check of all projects immediately. Note that if there are updates, the notification won't show up until all projects have been checked.

Project toevoegen

Opens a new dialog to set up a new project for monitoring.

Wijzigen

Opens the configuration dialog for the selected project.

Verwijderen

Removes the selected project after a confirmation dialog is shown.

Mark all as read

Marks all revisions in all projects as read. Note that if you select a project with unread revisions, those revisions are automatically marked as read when you select another project.

If you hold down the **Shift** key when clicking the button, all error states are also cleared if there are any.

Update all

Runs an **Update** on all monitored working copies. Projects that are monitored via an url are not updated, only those that are set up with a working copy path.

Opties

Shows a dialog to configure the behavior of the project monitor.

---

# Hoofdstuk 6. Het SubWCRev Programma

SubWCrev is een Windows opdrachtprompt programma, welke gebruikt kan worden voor het uitlezen van de status van een Subversion werkkopie en optioneel trefwoordvervangning met een template bestand kan uitvoeren. Dit wordt meestal gebruikt als onderdeel van het bouwproces om informatie van de werkkopie mee te nemen in het object wat je aan het bouwer bent. Meestal wordt het gebruikt om het revisienummer op te nemen in het "Over" scherm.

## 6.1. De SubWCRev Command Line

SubWCRev reads the Subversion status of all files in a working copy, excluding externals by default. It records the highest commit revision number found, and the commit timestamp of that revision, it also records whether there are local modifications in the working copy, or mixed update revisions. The revision number, update revision range and modification status are displayed on stdout.

SubWCRev.exe is called from the command line or a script, and is controlled using the command line parameters.

```
SubWCRev WorkingCopyPath [SrcVersionFile DstVersionFile] [-nmdfe]
```

WorkingCopyPath is het pad van de werkkopie die gecontroleerd moet worden. Je kunt SubWCRev alleen op werkkopieën gebruiken, niet direct op het archief. Het pad mag absoluut en relatief ten opzichte van de huidige werkmap zijn.

Als je SubWCRev trefwoordvervangning wilt laten uitvoeren, zodat bijvoorbeeld de velden archief revisie en URL in een bestand opgeslagen worden, dan moet je een template bestand SrcVersionFile en een resultaat bestand DstVersionFile voor de ingevulde resultaten specificeren.

You can specify ignore patterns for SubWCRev to prevent specific files and paths from being considered. The patterns are read from a file named `.subwcrevignore`. The file is read from the specified path, and also from the working copy root. If the file does not exist, no files or paths are ignored. The `.subwcrevignore` file can contain multiple patterns, separated by newlines. The patterns are matched against the paths relative to the repository root and paths relative to the path of the `.subwcrevignore` file. For example, to ignore all files in the `doc` folder of the TortoiseSVN working copy, the `.subwcrevignore` would contain the following lines:

```
/trunk/doc  
/trunk/doc/*
```

Or, assuming the `.subwcrevignore` file is in the working copy root which is checked out from trunk, using the patterns

```
doc  
doc/*
```

is the same as the example above.

To ignore all images, the ignore patterns could be set like this:

```
*.png  
*.jpg  
*.ico  
*.bmp
```



### Belangrijk

The ignore patterns are case-sensitive, just like Subversion is.



## Tip

To create a file with a starting dot in the Windows explorer, enter `.subwcrevignore..` Note the trailing dot.

There are a number of optional switches which affect the way SubWCRev works. If you use more than one, they must be specified as a single group, e.g. `-nm`, not `-n -m`.

Optie	Beschrijving
-n	Als deze optie gebruikt wordt, zal SubWCRev stoppen met <code>ERRORLEVEL 7</code> als de werkkopie lokale wijzigingen heeft. Dit kan gebruikt worden om te voorkomen dat er een pakket gebouwd wordt met niet vastgelegde wijzigingen.
-N	If this switch is given, SubWCRev will exit with <code>ERRORLEVEL 11</code> if the working copy contains unversioned items that are not ignored.
-m	Als deze optie gebruikt wordt, zal SubWCRev stoppen met <code>ERRORLEVEL 8</code> als de werkkopie meerdere revisies bevat. Dit kan gebruikt worden om te voorkomen dat er een pakket gebouwd wordt met een gedeeltelijk bijgewerkte werkkopie.
-d	Als deze optie gebruikt wordt, zal SubWCRev stoppen met <code>ERRORLEVEL 9</code> als het doel bestand al bestaat.
-f	Als deze optie gebruikt wordt, zal SubWCRev ook de laatst gewijzigde revisie van de mappen ophalen. Het standaard gedrag is zo dat alleen de revisie nummers van bestanden opgehaald worden.
-e	Als deze optie gebruikt wordt, zal SubWCRev mappen onderzoeken met <code>svn:externals</code> , maar alleen als ze van hetzelfde archief zijn. Standaard worden externen genegeerd.
-E	If this switch is given, same as <code>-e</code> , but it ignores the externals with explicit revisions, when the revision range inside of them is only the given explicit revision in the properties. So it doesn't lead to mixed revisions.
-x	Als deze optie gebruikt wordt, zal SubWCRev het revisie nummer in HEX formaat teruggeven.
-X	Als deze optie gebruikt wordt, zal SubWCRev het revisie nummer in HEX formaat teruggeven, vooraf gegaan door '0X'.
-F	If this switch is given, SubWCRev will ignore any <code>.subwcrevignore</code> files and include all files.
-q	If this switch is given, SubWCRev will perform the keyword substitution without showing working copy status on stdout.

**Tabel 6.1. Lijst met beschikbare opdrachtregel opties**

If there is no error, SubWCRev returns zero. But in case an error occurs, the error message is written to `stderr` and shown in the console. And the returned error codes are:

Fout code	Beschrijving
1	Syntax error. One or more command line parameters are invalid.
2	The file or folder specified on the command line was not found.
3	The input file could not be opened, or the target file could not be created.
4	Could not allocate memory. This could happen if e.g. the source file is too big.
5	The source file can not be scanned properly.
6	SVN error: Subversion returned with an error when SubWCRev tried to find the information from the working copy.
7	The working copy has local modifications. This requires the <code>-n</code> switch.



Fout code	Beschrijving
8	The working copy has mixed revisions. This requires the <code>-m</code> switch.
9	The output file already exists. This requires the <code>-d</code> switch.
10	The specified path is not a working copy or part of one.
11	The working copy has unversioned files or folders in it. This requires the <code>-N</code> switch.

**Tabel 6.2. List of SubWCRev error codes**

## 6.2. Trefwoordvervanging

Als een bron en doel bestand ondersteund wordt, kopieert SubWCRev de bron naar de bestemming, terwijl het trefwoordvervanging als volgt gebruikt:

Trefwoord	Beschrijving
<code>\$WCREV\$</code>	Vervangen door de hoogste vastgelegde revisie in de werkkopie.
<code>\$WCREV&amp;\$</code>	Replaced with the highest commit revision in the working copy, ANDed with the value after the <code>&amp;</code> char. For example: <code>\$WCREV&amp;0xFFFF\$</code>
<code>\$WCREV-\$</code> , <code>\$WCREV+\$</code>	Replaced with the highest commit revision in the working copy, with the value after the <code>+</code> or <code>-</code> char added or subtracted. For example: <code>\$WCREV-1000\$</code>
<code>\$WCDATE\$</code> , <code>\$WCDATEUTC\$</code>	Replaced with the commit date/time of the highest commit revision. By default, international format is used: <code>jjjj-mm-dd uu:mm:ss</code> . Alternatively, you can specify a custom format which will be used with <code>strftime()</code> , for example: <code>\$WCDATE=%a %b %d %I:%M:%S %p\$</code> . For a list of available formatting characters, look at the <a href="http://msdn.microsoft.com/en-us/library/fe06s4ak.aspx">online referentie</a> [http://msdn.microsoft.com/en-us/library/fe06s4ak.aspx].
<code>\$WCNOW\$</code> , <code>\$WCNOWUTC\$</code>	Vervangen door de huidige datum/tijd van het systeem. Dit kan gebruikt worden als de tijd van het bouwen van het pakket. Tijdsindeling kan gebruikt worden zoals beschreven in <code>\$WCDATE\$</code> .
<code>\$WCRANGES\$</code>	Replaced with the update revision range in the working copy. If the working copy is in a consistent state, this will be a single revision. If the working copy contains mixed revisions, either due to being out of date, or due to a deliberate update-to-revision, then the range will be shown in the form <code>100:200</code> .
<code>\$WCMIXED\$</code>	<code>\$WCMIXED?TText:FText\$</code> wordt vervangen door <code>TText</code> als er verschillende versies revisies zijn, <code>FText</code> als dat niet het geval is.
<code>\$WCMODS\$</code>	<code>\$WCMODS?TText:FText\$</code> wordt vervangen door <code>TText</code> als er lokale wijzigingen zijn, of <code>FText</code> als geen wijzigingen zijn.
<code>\$WCUNVER\$</code>	<code>\$WCUNVER?TText:FText\$</code> is replaced with <code>TText</code> if there are unversioned items in the working copy, or <code>FText</code> if not.
<code>\$WCEXTALLFIXED\$</code>	<code>\$WCEXTALLFIXED?TText:FText\$</code> is replaced with <code>TText</code> if all externals are fixed to an explicit revision, or <code>FText</code> if not.
<code>\$WCISTAGGED\$</code>	<code>\$WCISTAGGED?TText:FText\$</code> is replaced with <code>TText</code> if the repository URL contains the tags classification pattern, or <code>FText</code> if not.
<code>\$WCURL\$</code>	Vervangen door de URL van het archief van het pad van de werkkopie doorgeven aan SubWCRev.
<code>\$WCINSVNS\$</code>	<code>\$WCINSVN?TText:FText\$</code> wordt vervangen door <code>TText</code> als er voor deze optie versiebeheer gebruikt wordt, of <code>FText</code> als er geen versiebeheer gebruikt wordt.
<code>\$WCNEEDSLOCK\$</code>	<code>\$WCNEEDSLOCK?TText:FText\$</code> wordt vervangen door <code>TText</code> als voor de optie de <code>svn:needs-lock</code> eigenschap ingesteld is, of <code>FText</code> als deze niet ingesteld is.

Trefwoord	Beschrijving
\$WCISLOCKED\$	\$WCISLOCKED?TText:FText\$ wordt vervangen door TText als het object geblokkeerd is, of FText als het niet geblokkeerd is.
\$WCLOCKDATE\$, \$WCLOCKDATEUTC\$	Vervangen door de datum van de blokkade. Tijdsindeling kan gebruikt worden als beschreven voor \$WCDATE\$.
\$WCLOCKOWNER\$	Vervangen door de naam van de eigenaar van de blokkade.
\$WCLOCKCOMMENT\$	Vervangen door het commentaar van de blokkade.
\$WCUNVER\$	\$WCUNVER?TText:FText\$ is replaced with TText if there are unversioned files or folders in the working copy, or FText if not.

**Tabel 6.3. List of available keywords**

SubWCRev does not directly support nesting of expressions, so for example you cannot use an expression like:

```
#define SVN_REVISION    "$WCMIXED?$WCRANGE$: $WCREV$$"
```

But you can usually work around it by other means, for example:

```
#define SVN_RANGE      $WCRANGE$
#define SVN_REV        $WCREV$
#define SVN_REVISION   "$WCMIXED?SVN_RANGE:SVN_REV$"
```



### Tip

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\$.

## 6.3. Voorbeeld trefwoord

Dit voorbeeld laat zien hoe trefwoorden in een template bestand opgenomen worden in een resultaat bestand.

```
// Test file for SubWCRev

char *Revision      = "$WCREV$";
char *Revision16    = "$WCREV&0xFF$";
char *Revisionp100  = "$WCREV+100$";
char *Revisionm100  = "$WCREV-100$";
char *Modified       = "$WCMODS?Modified:Not modified$";
char *Unversioned    = "$WCUNVER?Unversioned items found:no unversioned items$";
char *Date           = "$WCDATE$";
char *CustDate       = "$WCDATE=%a, %d %B %Y$";
char *DateUTC        = "$WCDATEUTC$";
char *CustDateUTC    = "$WCDATEUTC=%a, %d %B %Y$";
char *TimeNow        = "$WCNOW$";
char *TimeNowUTC     = "$WCNOWUTC$";
char *RevRange       = "$WCRANGE$";
char *Mixed          = "$WCMIXED?Mixed revision WC:Not mixed$";
```

```

char *ExtAllFixed    = "$WCEXTALLFIXED?All externals fixed:Not all externals fixed$";
char *IsTagged      = "$WCISTAGGED?Tagged:Not tagged$";
char *URL           = "$WCURL$";
char *isInSVN      = "$WCINSVN?versioned:not versioned$";
char *needslock    = "$WCNEEDSLOCK?TRUE:FALSE$";
char *islocked     = "$WCISLOCKED?locked:not locked$";
char *lockdateutc  = "$WCLOCKDATEUTC$";
char *lockdate     = "$WCLOCKDATE$";
char *lockcustutc  = "$WCLOCKDATEUTC=%a, %d %B %Y$";
char *lockcust     = "$WCLOCKDATE=%a, %d %B %Y$";
char *lockown      = "$WCLOCKOWNER$";
char *lockcmt      = "$WCLOCKCOMMENT$";

```

```

#if $WCMODS?1:0$
#error Source is modified
#endif

```

```
// End of file
```

Nadat je SubWCRev.exe path\to\workingcopy testfile.tmpl testfile.txt hebt uitgevoerd, zal het resulterende bestand testfile.txt er zo uit zien:

```
// Test file for SubWCRev
```

```

char *Revision      = "22837";
char *Revision16   = "53";
char *Revisionpl100 = "22937";
char *Revisionml100 = "22737";
char *Modified     = "Modified";
char *Unversioned  = "no unversioned items";
char *Date         = "2012/04/26 18:47:57";
char *CustDate     = "Thu, 26 April 2012";
char *DateUTC      = "2012/04/26 16:47:57";
char *CustDateUTC  = "Thu, 26 April 2012";
char *TimeNow      = "2012/04/26 20:51:17";
char *TimeNowUTC   = "2012/04/26 18:51:17";
char *RevRange     = "22836:22837";
char *Mixed        = "Mixed revision WC";
char *ExtAllFixed  = "All externals fixed";
char *IsTagged     = "Not tagged";
char *URL          = "https://svn.code.sf.net/p/tortoisesvn/code/trunk";
char *isInSVN     = "versioned";
char *needslock   = "FALSE";
char *islocked    = "not locked";
char *lockdateutc = "1970/01/01 00:00:00";
char *lockdate    = "1970/01/01 01:00:00";
char *lockcustutc = "Thu, 01 January 1970";
char *lockcust    = "Thu, 01 January 1970";
char *lockown     = "";
char *lockcmt     = "";

```

```

#if 1
#error Source is modified
#endif

```

```
// End of file
```



## Tip

Een dergelijk bestand zal meegenomen worden in het bouwproces, je kunt dus verwachten dat ook onder versiebeheer valt. Zorg er voor dat alleen het template bestand van versiebeheer wordt voorzien, niet het gegenereerde bestand. Je moet anders elke keer dat je het versie bestand opnieuw genereert de wijziging vastleggen. Dat heeft dan weer tot gevolg dat je het versie bestand weer bij moet werken (omdat bijv. het revisienummer erin vermeld is).

## 6.4. COM interface

Als je Subversion revisie informatie vanuit andere programma's wilt gebruiken, dan kun je de COM interface van SubWCRev gebruiken. Het aan te maken object is `SubWCRev.object` en de volgende methodes kunnen gebruikt worden:

Methode	Beschrijving
<code>.GetWCInfo</code>	Deze methode doorzoekt de werkkopie en verzamelt de revisie informatie. Je moet deze natuurlijk eerst aanroepen voordat je de informatie gebruikt voor de andere methodes. De eerste parameter is het pad. De tweede parameter moet waar zijn als je de revisies van de mappen mee wilt nemen. Dit komt overeen met de <code>-f</code> opdrachtregel optie. De derde optie moet waar zijn als je <code>svn:externals</code> mee wilt nemen. Dit komt overeen met de <code>-e</code> opdrachtregel optie.
<code>.GetWCInfo2</code>	The same as <code>GetWCInfo()</code> but with a fourth parameter that sets the equivalent to the <code>-E</code> command line switch.
<code>.Revision</code>	The highest commit revision in the working copy. Equivalent to <code>\$WCREV\$</code> .
<code>.Date</code>	The commit date/time of the highest commit revision. Equivalent to <code>\$WCDATE\$</code> .
<code>.Author</code>	De auteur van de hoogste vastgelegde revisie, oftewel de laatste persoon die wijzigingen heeft vastgelegd.
<code>.MinRev</code>	De minimale bijgewerkte versie, zoals weergegeven in <code>\$WCRANGE\$</code>
<code>.MaxRev</code>	De maximale bijgewerkte revisie, zoals weergegeven in <code>\$WCRANGE\$</code>
<code>.HasModifications</code>	True, wanneer er lokale aanpassingen zijn
<code>.HasUnversioned</code>	True if there are unversioned items
<code>.Url</code>	Replaced with the repository URL of the working copy path used in <code>GetWCInfo</code> . Equivalent to <code>\$WCURL\$</code> .
<code>.IsSvnItem</code>	True, wanneer het item opgenomen is in versiebeheer.
<code>.NeedsLocking</code>	True, als het item de eigenschap <code>svn:needs-lock</code> heeft ingesteld.
<code>.IsLocked</code>	True, als het item vergrendeld is.
<code>.LockCreationDate</code>	De regel geeft de datum waarop de blokkade geplaatst is of is leeg als er geen blokkade is.
<code>.LockOwner</code>	De regel bevat de eigenaar van de blokkade of is leeg als het object niet geblokkeerd is.
<code>.LockComment</code>	Het bericht dat meegegeven werd toen een blokkade geplaatst is.

**Tabel 6.4. COM/automatiseringsmethoden worden ondersteund**

Het volgende voorbeeld geeft aan hoe je de interface zou kunnen gebruiken.

```
// 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.GetWCInfo2(
    filesystem.GetAbsolutePathName("../.."), 1, 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);
```

Het volgende voorbeeld beschrijft hoe je de SubWCRev COM objecten vanuit C# moet gebruiken:

```
using LibSubWCRev;
SubWCRev sub = new SubWCRev();
sub.GetWCInfo("C:\\PathToMyFile\\MyFile.cc", true, true);
if (sub.IsSvnItem == true)
{
    MessageBox.Show("versioned");
}
else
{
    MessageBox.Show("not versioned");
}
```

---

# Hoofdstuk 7. IBugtraqProvider interface

Om een betere integratie met issue trackers te krijgen dan alleen de `bugtraq`: te gebruiken, kan TortoiseSVN gebruik maken van COM plugins. Met deze plugins is het mogelijk om direct informatie uit het issue tracker systeem te halen, met de gebruiker te communiceren, om informatie aan TortoiseSVN terug te geven over open issues, door de gebruiker ingevoerde logboek berichten te controleren en zelfs het starten van acties na een succesvolle vastlegging, bijv, het sluiten van een issue.

We can't provide information and tutorials on how you have to implement a COM object in your preferred programming language, but we have example plugins in C++/ATL and C# in our repository in the `contrib/issue-tracker-plugins` folder. In that folder you can also find the required include files you need to build your plugin. (Paragraaf 3, "Licentie" explains how to access the repository.)



## Belangrijk

Je moet zowel een 32-bit als 64-bit versie van je plugin maken. De x64-versie van TortoiseSVN kan de 32-bit plugin niet gebruiken en vice versa.

## 7.1. Naming conventions

If you release an issue tracker plugin for TortoiseSVN, please do *not* name it *Tortoise<Something>*. We'd like to reserve the *Tortoise* prefix for a version control client integrated into the windows shell. For example: TortoiseCVS, TortoiseSVN, TortoiseHg, TortoiseGit and TortoiseBzr are all version control clients.

Please name your plugin for a Tortoise client *Turtle<Something>*, where *<Something>* refers to the issue tracker that you are connecting to. Alternatively choose a name that sounds like *Turtle* but has a different first letter. Nice examples are:

- Gurtle - An issue tracker plugin for Google code
- TurtleMine - An issue tracker plugin for Redmine
- VurtleOne - An issue tracker plugin for VersionOne

## 7.2. De IBugtraqProvider interface

TortoiseSVN 1.5 en later kunnen werken met plugins die gebruik maken van de IBugtraqProvider interface. Deze interface biedt enkele methodes die plugins kunnen gebruiken om de communiceren met een issue tracker.

```
HRESULT ValidateParameters (  
    // Hoofd scherm elke UI die weergegeven  
    // moet worden tijdens validatie.  
    [in] HWND hParentWnd,  
  
    // De parameter tekst die gevalideerd moet worden.  
    [in] BSTR parameters,  
  
    // Is de tekst geldig?  
    [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 (
    // Hoofdscherm voor elke (foutmeldings) UI die getoond moet worden.
    [in] HWND hParentWnd,

    // De parameter tekst, voor het geval je informatie moet geven aan je
    // web service (bijv.) om uit te vinden wat de juist tekst is.
    [in] BSTR parameters,

    // Welke tekst wilt je tonen?
    // Gebruik de huidige thread locale.
    [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 (
    // Hoofdscherm voor je provider's UI.
    [in] HWND hParentWnd,

    // Parameters voor je provider.
    [in] BSTR parameters,
    [in] BSTR commonRoot,
    [in] SAFEARRAY(BSTR) pathList,

    // De tekst die al in het vastleg venster staat.
    // Je provider moet deze tekst opnemen in het
    // nieuwe bericht, indien gewenst.
    [in] BSTR originalMessage,

    // De nieuwe vastleg tekst.
    // Deze vervangt het originele bericht.
    [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. For the branch/tag dialog, this is the path which is to be copied.

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.



### 7.3. De IBugtraqProvider2 interface

In TortoiseSVN 1.6 is een nieuwe interface toegevoegd, welke het gebruik van plugins mogelijk maakt. Deze IBugtraqProvider2 interface erft functionaliteit van IBugtraqProvider.

```
HRESULT GetCommitMessage2 (
    // Hoofd scherm voor je provider UI.
    [in] HWND hParentWnd,

    // Parameters voor jouw provider.
    [in] BSTR parameters,
    // De gemeenschappelijke URL van de vastlegging
    [in] BSTR commonURL,
    [in] BSTR commonRoot,
    [in] SAFEARRAY(BSTR) pathList,

    // De tekst die al aanwezig is in het vastleg bericht.
    // Je provider moet deze tekst opnemen in het nieuwe bericht,
    // indien gewenst.
    [in] BSTR originalMessage,

    // Je kunt eigen revisie eigenschappen toekennen aan een vastlegging
    // door de volgende twee parameters in te stellen.
    // noot: Beide Both safearrays moeten dezelfde lengte hebben.
    //       Elke eigenschas naam moet er een waarde hebben!

    // De inhoud van het bugID veld(indien weergegeven)
    [in] BSTR bugID,

    // Gewijzigde inhoud van het bugID veld
    [out] BSTR * bugIDOut,

    // De lijst van revision eigenschapsnamen.
    [out] SAFEARRAY(BSTR) * revPropNames,

    // De lijst van revisie eigenschap waardes.
    [out] SAFEARRAY(BSTR) * revPropValues,

    // De nieuwe tekst van het vastleg bericht.
    // Deze vervangt het originele bericht
    [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.

For the branch/tag dialog, the `commonURL` is the source URL of the copy, and `commonRoot` is set to the target URL of the copy.

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 (
    // Hoofdscherm voor elk (foutmelding) UI welke getoond moet worden.
    [in] HWND hParentWnd,

    // De gemeenschappelijke bron voor alle paden die vastgelegd worden.
    [in] BSTR commonRoot,

    // Alle paden die vastgelegd zijn.
    [in] SAFEARRAY(BSTR) pathList,

    // De reeds aanwezige tekst in het vastleg bericht.
    [in] BSTR logMessage,

    // De revisie van de vastlegging.
    [in] ULONG revision,

    // Een fout getoond aan de gebruiker als deze functie
    // iets anders teruggeeft dan 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(
    // Of de provider opties ondersteund
    [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(  
    // Parent window for the options dialog  
    [in] HWND hParentWnd,  
  
    // Parameters voor je provider.  
    [in] BSTR parameters,  
  
    // De parameter tekst  
    [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.

---

# Bijlage A. Veelgestelde Vragen (FAQ)

Because TortoiseSVN is being developed all the time it is sometimes hard to keep the documentation completely up to date. We maintain an *online FAQ* [<https://tortoisesvn.net/faq.html>] which contains a selection of the questions we are asked the most on the TortoiseSVN mailing lists <https://groups.google.com/forum/#!forum/tortoisesvn> and <https://groups.google.com/forum/#!forum/tortoisesvn-dev>

Als je een vraag hebt die nergens beantwoord wordt, dan kun je die vraag het beste stellen via één van de maillijsten:

- <https://groups.google.com/forum/#!forum/tortoisesvn> is the one to use if you have questions about using TortoiseSVN.
- If you want to help out with the development of TortoiseSVN, then you should take part in discussions on <https://groups.google.com/forum/#!forum/tortoisesvn-dev>

---

# Bijlage B. Hoe kan ik...

Deze appendix bevat oplossingen voor problemen/vragen die je mogelijk hebt bij het gebruik van TortoiseSVN.

## B.1. Verplaats een hoop bestanden in één keer

Verplaatsen/Kopiëren van enkele bestanden kan gedaan worden met TortoiseSVN → Hernoemen.... Als je echter veel bestanden wilt verplaatsen/kopiëren, dan is deze methode veel te langzaam en kost het te veel werk.

The recommended way is by right dragging the files to the new location. Simply right click on the files you want to move/copy without releasing the mouse button. Then drag the files to the new location and release the mouse button. A context menu will appear where you can either choose Context Menu → SVN Copy versioned files here. or Context Menu → SVN Move versioned files here.

### Afbeelding B.1. The TortoiseSVN right drag context menu for moving files

## B.2. Forceer gebruikers een log-bericht in te typen

Er zijn twee methodes om te voorkomen dat gebruikers wijzigingen vastleggen zonder logboek boodschap. Eentje is specifiek voor TortoiseSVN, de andere werkt op alle Subversion programma's, maar deze heeft dan weer directe toegang tot de server nodig.

### B.2.1. Hook-script op de server

Als je direct toegang tot de server hebt, kun je een vastleg hook script installeren, welke een vastlegging weigert als er een leeg of te kort logboek boodschap wordt gebruikt.

In de archief map op de server is een sub-map `hooks` aanwezig. Hierin staan enkele hook script voorbeelden, die je kunt gebruiken. Het bestand `pre-commit.tmpl` bevat een voorbeeld script waarmee een vastlegging geweigerd wordt als er geen logboek boodschap ingevuld is of als het bericht te kort is. In dit bestand staat ook omschreven hoe je het script moet installeren of gebruiken. Volg de instructies die in dat bestand staan.

Deze methode is de aanbevolen methode als je gebruikers ook andere Subversion programma's gebruiken dan TortoiseSVN. Het nadeel van deze methode is dat vastlegging geweigerd wordt door de server en de gebruikers dus een foutmelding krijgen. Het programma weet niet dat de vastlegging geweigerd wordt, voordat er contact is gemaakt met de server. Als je wilt dat TortoiseSVN de OK inactief maakt, totdat er een logboek boodschap is ingevuld die lang genoeg is, gebruik dan de methode die hieronder beschreven wordt.

### B.2.2. Project eigenschappen

TortoiseSVN maakt voor enkele functies gebruik van eigenschappen. Eén van die eigenschappen is de `tsvn:logminsize` eigenschap.

Als je deze eigenschap aan een map toekent, dan zal TortoiseSVN de OK inactief maken bij alle vastleg vensters, totdat de gebruiker een logboek boodschap heeft ingevoerd met de minimale lengte zoals aangegeven in de eigenschap.

For detailed information on those project properties, please refer to [Paragraaf 4.18, "Project Instellingen"](#).

## B.3. Update Geselecteerde Bestanden in het Archief

Normaliter ververs je je werkkopie met TortoiseSVN → Verversen. Je hebt een iets andere aanpak nodig, als je maar enkele nieuwe bestanden, die een collega heeft toegevoegd, wilt ophalen zonder meteen wijzigingen aan andere bestanden samen te laten voeren.

Gebruik TortoiseSVN → Kijk of er updates zijn en klik op Controleer archief om te zien wat er gewijzigd is in het archief. Selecteer de bestanden die je lokaal wilt verversen en gebruik dan het context menu om deze bestanden te verversen.

## B.4. Draai Revisies Terug in het Archief (ongedaan maken)

### B.4.1. Gebruik het logboek boodschappen scherm

By far the easiest way to revert the changes from one or more revisions, is to use the revision log dialog.

1. Selecteer het bestand of de map waarvoor je de wijzigingen ongedaan wilt maken. Als je alle wijzigingen ongedaan wilt maken, moet je de hoogste/basis map selecteren.
2. Selecteer TortoiseSVN → Toon logboek om de lijst met revisies weer te geven. Je moet misschien de Toon Alles of Volgende 100 knoppen gebruiken om de revisie(s) te zien, waarin je geïnteresseerd bent.
3. Select the revision you wish to revert. If you want to undo a range of revisions, select the first one and hold the **Shift** key while selecting the last one. If you want to pick out individual revisions and ranges, use the **Ctrl** key while selecting revisions. Right click on the selected revision(s), then select Context Menu → Revert changes from this revision.
4. Of als je een eerdere revisie de nieuwe HEAD revisie wilt maken, klik met rechts op de geselecteerde revisie en selecteer vervolgens Context Menu → Ongedaan maken tot deze revisie. Hiermee worden *alle* wijzigingen na de geselecteerde revisie ongedaan gemaakt.

Je hebt nu de wijzigingen in je werkkopie ongedaan gemaakt. Controleer het resultaat en leg de wijzigingen vast.

### B.4.2. Gebruik het samenvoeg scherm

If you want to enter revision numbers as a list, you can use the Merge dialog. The previous method uses merging behind the scenes; this method uses it explicitly.

1. Selecteer in je werkkopie TortoiseSVN → Samenvoegen.
2. In the Merge Type dialog select Merge a range of revisions.
3. In the From: field enter the full repository URL of your working copy folder. This should come up as the default URL.
4. In the Revision range to merge field enter the list of revisions to roll back (or use the log dialog to select them as described above).
5. Make sure the Reverse merge checkbox is checked.
6. In the Merge options dialog accept the defaults.
7. Click Merge to complete the merge.

You have reverted the changes within your working copy. Check that the results are as expected, then commit the changes.

### B.4.3. Gebruik svndumpfilter

Omdat TortoiseSVN nooit gegevens zal verliezen, bestaan je “ongedaan gemaakt” revisies nog steeds als tussentijdse revisies in het archief. De HEAD revisie is alleen veranderd naar een eerdere status. Als je wilt dat de revisies volledig uit het archief verdwijnen, inclusief alle informatie over het bestaan ervan, dan moet je extreme maatregelen nemen. Tenzij er hiervoor echt een hele goede reden is, wordt dit *met klem afgeraden*. Een mogelijke reden is bijvoorbeeld dat iemand een vertrouwelijk document in een publiek archief heeft vastgelegd.

The only way to remove data from the repository is to use the Subversion command line tool `svnadmin`. You can find a description of how this works in the *Repository Maintenance* [<http://svnbook.red-bean.com/en/1.8/svn.reposadmin.maint.html>].

## B.5. Vergelijk Twee Revisies van een Bestand of Map

Als je de geschiedenis van twee revisies van een object wilt vergelijken, bijvoorbeeld voor revisie 100 en 200 van hetzelfde bestand, gebruik je TortoiseSVN → Toon logboek om de geschiedenis van dat bestand te bekijken. Selecteer de twee revisies die je wilt vergelijken en gebruik dan Context Menu → Vergelijk Revisies.

Als je hetzelfde object in twee verschillende mapstructuren wilt bekijken, bijvoorbeeld in de basislijn en een tak, gebruik dan de archief-verkenner om beide boomstructuren te openen. Selecteer de bestanden in beide locaties en gebruik Context Menu → Vergelijk Revisies.

Als je twee boomstructuren wilt vergelijken om na te gaan wat er gewijzigd is, bijvoorbeeld de basislijn en een gelabelde vrijgave, gebruik dan het TortoiseSVN → Revisie schema. Selecteer de twee te vergelijken knopen en kies Context Menu → Vergelijk HEAD Revisies. Er wordt dan een lijst met gewijzigde bestanden getoond waarin je individuele bestanden kunt selecteren, waarvoor je de wijzigingen in detail wilt zien. Je kunt ook de mapstructuur met alle gewijzigde bestanden exporteren, of simpelweg een lijst met gewijzigde bestanden. Lees [Paragraaf 4.11.3, “Vergelijken van mappen”](#) voor meer informatie. Als alternatief kun je ook Context Menu → GNU diff van HEAD revisies om een samenvatting met minimale context van de wijzigingen te bekijken.

## B.6. Een veelgebruikt subproject gebruiken

Sometimes you will want to include another project within your working copy, perhaps some library code. There are at least 4 ways of dealing with this.

### B.6.1. svn:externals Gebruiken

Stel de `svn:externals` eigenschap in op een map in je project. Deze eigenschap kan één of meer regels hebben; elke regel bevat de naam van een sub-map welke je wilt gebruiken als de doelmap voor gemeenschappelijke broncode en de URL van het archief waar vandaan je de gegevens op wilt halen. Lees voor meer details [Paragraaf 4.19, “Externe Objecten”](#).

Leg de nieuwe map vast in het archief. Zodra je nu ververst, zal Subversion een kopie van dat project maken vanuit het archief naar jouw werkkopie. Sub-mappen worden automatisch aangemaakt als dat nodig is. Elke keer dat je werkkopie ververst, wordt ook de laatste wijzigingen van het externe project opgehaald.

Als het externe project zich in hetzelfde archief bevindt, dan worden de wijzigingen die je daarin aanbrengt meegenomen in de vastleglijst als je het hoofdproject vastlegt.

If the external project is in a different repository, any changes you make to the external project will be shown or indicated when you commit the main project, but you have to commit those external changes separately.

Van de drie beschreven methodes, is dit de enige methode die geen instellingen vereist aan de kant van de gebruikers. Als de externen eenmaal toegekend zijn aan de map-eigenschappen, zullen alle gebruikers automatisch de betreffende externe gegevens krijgen als zij hun werkkopie verversen.

### B.6.2. Een geneste werkkopie gebruiken

Create a new folder within your project to contain the common code, but do not add it to Subversion.

Selecteer TortoiseSVN → Ophalen voor de nieuwe map en haal een kopie van de gemeenschappelijke broncode op. Je hebt nu een aparte werkkopie genest in jouw hoofd werkkopie.

De twee werkkopieën zijn onafhankelijk. Als je wijzigingen vastlegt van de bovenliggende map, dan worden wijzigingen in de geneste werkkopie genegeerd. Ook als je de bovenliggende map ververst, wordt de geneste werkkopie niet ververst.

### B.6.3. Gebruik een relatieve lokatie

If you use the same common core code in several projects, and you do not want to keep multiple working copies of it for every project that uses it, you can just check it out to a separate location which is related to all the other projects which use it. For example:

```
C:\Projects\Proj1
C:\Projects\Proj2
C:\Projects\Proj3
C:\Projects\Common
```

and refer to the common code using a relative path, e.g. `..\..\Common\DSPcore`.

If your projects are scattered in unrelated locations you can use a variant of this, which is to put the common code in one location and use drive letter substitution to map that location to something you can hard code in your projects, e.g. Checkout the common code to `D:\Documents\Framework` or `C:\Documents` and `Settings\{login}\My Documents\framework` then use

```
SUBST X: "D:\Documents\framework"
```

to create the drive mapping used in your source code. Your code can then use absolute locations.

```
#include "X:\superio\superio.h"
```

Deze methode zal alleen een PC-omgeving en je moet nauwkeurig aangeven welke schijffletters er toegewezen moeten worden, zodat je team weer waar ze de mysterieuze bestanden kunnen vinden. Deze methode moet je alleen gebruiken in een gesloten ontwikkelomgeving en is niet aanbevolen voor generiek gebruik.

### B.6.4. Add the project to the repository

The maybe easiest way is to simply add the project in a subfolder to your own project working copy. However this has the disadvantage that you have to update and upgrade this external project manually.

To help with the upgrade, TortoiseSVN provides a command in the explorer right-drag context menu. Simply right-drag the folder where you unzipped the new version of the external library to the folder in your working copy, and then select Context Menu → SVN Vendorbranch here. This will then copy the new files over to the target folder while automatically adding new files and removing files that aren't in the new version anymore.

## B.7. Maak een snelkoppeling naar een archief

If you frequently need to open the repository browser at a particular location, you can create a desktop shortcut using the automation interface to TortoiseProc. Just create a new shortcut and set the target to:

```
TortoiseProc.exe /command:repobrowser /path:"url/to/repository"
```

Of course you need to include the real repository URL.

## B.8. Negeer bestanden die al opgenomen zijn in het versiebeheer

Stel, je hebt per ongeluk enkele bestanden toegevoegd aan het archief, terwijl die eigenlijk genegeerd zouden moeten worden. Hoe krijg je die bestanden dan weer uit het versiebeheer systeem? Je hebt bijvoorbeeld je eigen



IDE configuratie bestand die geen onderdeel is van het project, maar wat je wel veel tijd heeft gekost om alles naar jouw wens in te stellen.

If you have not yet committed the add, then all you have to do is use TortoiseSVN → Undo Add... to undo the add. You should then add the file(s) to the ignore list so they don't get added again later by mistake.

If the files are already in the repository, they have to be deleted from the repository and added to the ignore list. Fortunately TortoiseSVN has a convenient shortcut for doing this. TortoiseSVN → Unversion and add to ignore list will first mark the file/folder for deletion from the repository, keeping the local copy. It also adds this item to the ignore list so that it will not be added back into Subversion again by mistake. Once this is done you just need to commit the parent folder.

## B.9. Versiebeheer van een werkkopie verwijderen

If you have a working copy which you want to convert back to a plain folder tree without the .svn directory, you can simply export it to itself. Read [Paragraaf 4.27.1, “Versiebeheer van een werkkopie verwijderen”](#) to find out how.

## B.10. Een werkkopie verwijderen

If you have a working copy which you no longer need, how do you get rid of it cleanly? Easy - just delete it in Windows Explorer! Working copies are private local entities, and they are self-contained. Deleting a working copy in Windows Explorer does not affect the data in the repository at all.

---

# Bijlage C. Handige tips voor beheerders

Deze bijlage bevat oplossingen voor problemen/vragen die je zou kunnen hebben wanneer je verantwoordelijk bent voor het verspreiden van TortoiseSVN over meerdere client computers.

## C.1. Verspreid TotoiseSVN via groepsbeleid

De TortoiseSVN installatie wordt geleverd als een MSI-bestand, wat inhoudt dat je geen problemen zou moeten ondervinden bij het toevoegen van dat MSI-bestand aan het groepsbeleid van je domeincontroller.

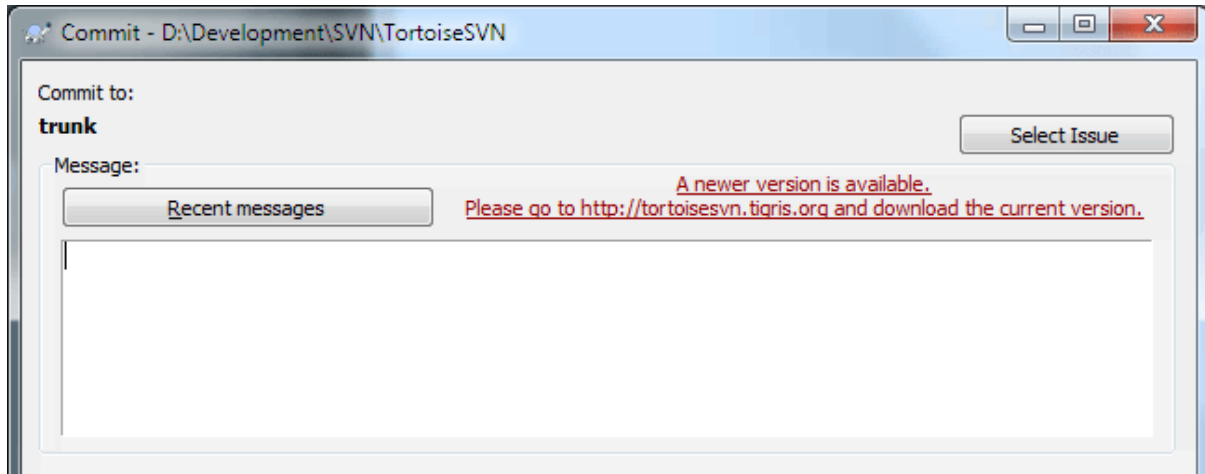
Een goede handleiding over hoe dat te doen kan gevonden worden in kennisbank artikel 314934 van Microsoft: <http://support.microsoft.com/?kbid=314934>.

TortoiseSVN must be installed under *Computer Configuration* and not under *User Configuration*. This is because TortoiseSVN needs the CRT and MFC DLLs, which can only be deployed *per computer* and not *per user*. If you really must install TortoiseSVN on a per user basis, then you must first install the MFC and CRT package version 12 from Microsoft on each computer you want to install TortoiseSVN as per user.

You can customize the MSI file if you wish so that all your users end up with the same settings. TSVN settings are stored in the registry under `HKEY_CURRENT_USER\Software\TortoiseSVN` and general Subversion settings (which affect all Subversion clients) are stored in config files under `%APPDATA%\Subversion`. If you need help with MSI customization, try one of the MSI transform forums or search the web for “MSI transform”.

## C.2. De update controle omleiden

TortoiseSVN checks if there's a new version available every few days. If there is a newer version available, a notification is shown in the commit dialog.



**Afbeelding C.1. De commit-dialoog terwijl deze de upgrade-notificatie weergeeft**

If you're responsible for a lot of users in your domain, you might want your users to use only versions you have approved and not have them install always the latest version. You probably don't want that upgrade notification to show up so your users don't go and upgrade immediately.

Versions 1.4.0 and later of TortoiseSVN allow you to redirect that upgrade check to your intranet server. You can set the registry key `HKCU\Software\TortoiseSVN\UpdateCheckURL` (string value) to an URL pointing to a text file in your intranet. That text file must have the following format:

```
1.9.1.6000
```

```
A new version of TortoiseSVN is available for you to download!
```

`http://192.168.2.1/downloads/TortoiseSVN-1.9.1.6000-svn-1.9.1.msi`

The first line in that file is the version string. You must make sure that it matches the exact version string of the TortoiseSVN installation package. The second line is a custom text, shown in the commit dialog. You can write there whatever you want. Just note that the space in the commit dialog is limited. Too long messages will get truncated! The third line is the URL to the new installation package. This URL is opened when the user clicks on the custom message label in the commit dialog. You can also just point the user to a web page instead of the MSI file directly. The URL is opened with the default web browser, so if you specify a web page, that page is opened and shown to the user. If you specify the MSI package, the browser will ask the user to save the MSI file locally.

### C.3. Zetten van de `SVN_ASP_DOT_NET_HACK` omgevingsvariable

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 confusion for users who always install *everything* no matter whether they know what it is for.

But the feature is still available in TortoiseSVN and other svn clients. To enable it you have to set the Windows environment variable named `ASPDOTNETHACK` to 1. Actually, the value of that environment variable doesn't matter: if the variable exists the feature is active.



#### Belangrijk

Please note that this hack is only necessary if you're still using VS.NET2002. All later versions of Visual Studio do *not* require this hack to be activated! So unless you're using that ancient tool, DO NOT USE THIS!

### C.4. Opties in het contextmenu uitschakelen

Vanaf versie 1.5.0 en hoger, staat TortoiseSvn je toe om opties van het context menu uit te schakelen (eigenlijk verbergen). Aangezien dit een eigenschap is dat niet lichtvaardig zou moeten worden gebruikt, maar alleen als er een dwingende reden voor is, is er geen GUI voor aanwezig en moet dit rechtstreeks in het register worden gedaan. Dit kan gebruikt worden om bepaalde commando's uit te schakelen voor gebruikers die deze niet zouden moeten gebruiken. Maar let er wel op dat alleen de context menu opties in de *Verkenner* zijn verborgen. De commando's zijn nog steeds beschikbaar via andere middelen, zoals bijvoorbeeld de opdrachtregel of zelfs andere dialogen in TortoiseSVN zelf!

De registersleutels die de informatie bevatten over welke contextmenus getoond moeten worden zijn `HKEY_CURRENT_USER\Software\TortoiseSVN\ContextMenuEntriesMaskLow` en `HKEY_CURRENT_USER\Software\TortoiseSVN\ContextMenuEntriesMaskHigh`.

Elk van deze registersleutels is een `DWORD` waarde, waarvan elke bit correspondeert met een specifieke menuoptie. Een bit dat gezet is, betekent dat de bijbehorende menuoptie is uitgeschakeld.

Waarde	Menuoptie
0x0000000000000001	Ophalen
0x0000000000000002	Verversen
0x0000000000000004	Vastleggen
0x0000000000000008	Toevoegen
0x0000000000000010	Ongedaan maken
0x0000000000000020	Schonen
0x0000000000000040	Oplossen
0x0000000000000080	Optie
0x0000000000000100	Importeren

Waarde	Menuoptie
0x0000000000000200	Exporteren
0x0000000000000400	Maak hier archief aan
0x0000000000000800	Tak/Label
0x0000000000001000	Samenvoegen
0x0000000000002000	Verwijderen
0x0000000000004000	Hernoemen
0x0000000000008000	Verversen naar revisie
0x0000000000010000	Vershil
0x0000000000020000	Toon logboek
0x0000000000040000	Bewerk conflicten
0x0000000000080000	Locatie wijzigen
0x0000000000100000	Controleer op wijzigingen
0x0000000000200000	Negeer
0x0000000000400000	Archief-verkenner
0x0000000000800000	Verklaren
0x0000000001000000	Maak patch
0x0000000002000000	Patch toepassen
0x0000000004000000	Revisie schema
0x0000000008000000	Blokkade
0x0000000010000000	Blokkade opheffen
0x0000000020000000	Eigenschappen
0x0000000040000000	Vershil met URL
0x0000000080000000	Verwijder niet-geversioneerde items
0x0000000100000000	Alles samenvoegen
0x0000000200000000	Vergelijk met vorige versie
0x0000000400000000	Plakken
0x0000000800000000	Upgrade werkkopie
0x0000001000000000	Vergelijk later
0x0000002000000000	Diff with 'filename'
0x0000004000000000	Unified diff
0x2000000000000000	Instellingen
0x4000000000000000	Help
0x8000000000000000	Over

**Tabel C.1. Menuopties en hun waarden**

Example: to disable the “Relocate” the “Delete unversioned items” and the “Settings” menu entries, add the values assigned to the entries like this:

```
0x0000000000008000
+ 0x0000000080000000
+ 0x2000000000000000
```

= 0x2000000080080000

The lower DWORD value (0x80080000) must then be stored in HKEY\_CURRENT\_USER\Software\TortoiseSVN\ContextMenuEntriesMaskLow, the higher DWORD value (0x20000000) in HKEY\_CURRENT\_USER\Software\TortoiseSVN\ContextMenuEntriesMaskHigh.

Voor het opnieuw inschakelen van de menuopties, kun je eenvoudig deze twee registersleutels verwijderen.

---

# Bijlage D. Automatisering van TortoiseSVN

Omdat alle commando's voor TortoiseSVN worden bestuurd via commandoregel parameters kunt u dit automatiseren door middel van batch scripts of specifieke commando's of schermen van andere programma's opstarten. (bijvoorbeeld uw favoriete tekstverwerker).



## Belangrijk

Onthoudt dat TortoiseSVN een GUI client is en deze automatiserings handleiding aangeeft hoe de TortoiseSVN schermen te tonen welke gebruikersinvoer verzamelen. Als u een script wilt schrijven dat geen enkele invoer van de gebruiker vereist, dient u de officiële Subversion commando-regel client gebruiken.

## D.1. TortoiseSVN commando's

Het TortoiseSVN GUI programma wordt `TortoiseProc.exe` genoemd. Alle commando's worden opgegeven met de parameter `/command:abcd` waarbij `abcd` de naam is van het vereiste commando. De meeste van deze commando's hebben ten minste één pad-argument nodig, welke wordt gegeven met `/path:"some\path"`. In de volgende tabel verwijst het commando naar de `/command:abcd` parameter en het pad verwijst naar de `/path:"some\path"` parameter.

There's a special command that does not require the parameter `/command:abcd` but, if nothing is specified on the command line, starts the project monitor instead. If `/tray` is specified, the project monitor starts hidden and only adds its icon to the system tray.

Omdat sommige commando's een lijst van doelpaden kunnen aannemen (bijv. een aantal specifieke bestanden vastleggen), kan de `/path` parameter meerdere paden, gescheiden door een `*` teken, bevatten.

You can also specify a file which contains a list of paths, separated by newlines. The file must be in UTF-16 format, without a *BOM* [[https://en.wikipedia.org/wiki/Byte-order\\_mark](https://en.wikipedia.org/wiki/Byte-order_mark)]. If you pass such a file, use `/pathfile` instead of `/path`. To have TortoiseProc delete that file after the command is finished, you can pass the parameter `/deletepathfile`. If you don't pass `/deletepathfile`, you have to delete the file yourself or the file gets left behind.

Het voortgangsvenster, dat wordt gebruikt voor vastleggingen, bewerkingen en vele andere commando's, blijft gewoonlijk geopend nadat het commando is beëindigd totdat de gebruiker op de OK knop klikt. Dit kan worden gewijzigd door de corresponderende optie in het instellingenvenster aan te vinken. Maar als die instelling wordt gebruikt, zal het voortgangsvenster sluiten hoe het commando ook vanuit het batch-bestand of vanuit de TortoiseSVN contextmenu wordt opgestart.

Om een andere locatie van het configuratiebestand aan te geven gebruikt u de parameter `/configdir:"path\to\config\directory"`. Dit zal het standaardpad, alsook elke registry instelling overschrijven.

Om het voortgangsscherm aan het einde van een commando automatisch te sluiten zonder de permanente instelling te gebruiken, kunt u de `/closeonend` parameter opgeven.

- `/closeonend:0` het venster niet automatisch sluiten
- `/closeonend:1` automatisch sluiten indien er geen fouten zijn
- `/closeonend:2` automatisch sluiten indien er geen fouten of conflicten zijn
- `/closeonend:3` automatisch sluiten indien er geen fouten, conflicten of samenvoegingen zijn

Om het voortgangsvenster voor locale operaties te sluiten als er geen fouten of conflicten zijn, geeft u de `/closeforlocal` parameter op.

De onderstaande tabel toont alle commando's welke kunnen worden benaderd met de TortoiseProc.exe commando regel. Zoals hierboven is beschreven, dienen deze in de vorm `/command:abcd` te worden gebruikt. In de tabel wordt het voorvoegsel `/command` weggelaten om ruimte te sparen.

Commando	Beschrijving
<code>:about</code>	Toont het 'over' scherm. Dit wordt ook getoond als geen enkel commando is opgegeven.
<code>:log</code>	<p>Opens the log dialog. The <code>/path</code> specifies the file or folder for which the log should be shown. Additional options can be set:</p> <ul style="list-style-type: none"> <li>• <code>/startrev:xxx</code>,</li> <li>• <code>/endrev:xxx</code>,</li> <li>• <code>/limit:xxx</code> limits the amount of fetched messages</li> <li>• <code>/strict</code> enables the 'stop-on-copy' checkbox,</li> <li>• <code>/merge</code> enables the 'include merged revisions' checkbox,</li> <li>• <code>/datemin: "{datestring}"</code> sets the start date of the filter, and</li> <li>• <code>/datemax: "{datestring}"</code> sets the end date of the filter. The date format is the same as used for svn date revisions.</li> <li>• <code>/findstring: "filterstring"</code> fills in the filter text,</li> <li>• <code>/findtext</code> forces the filter to use text, not regex, or</li> <li>• <code>/findregex</code> forces the filter to use regex, not simple text search, and</li> <li>• <code>/findtype:X</code> with X being a number between 0 and 511. The numbers are the sum of the following options: <ul style="list-style-type: none"> <li>• <code>/findtype:0</code> filtert op alles</li> <li>• <code>/findtype:1</code> filtert op berichten</li> <li>• <code>/findtype:2</code> filtert op pad</li> <li>• <code>/findtype:4</code> filtert op auteurs</li> <li>• <code>/findtype:8</code> filtert op revisies</li> <li>• <code>/findtype:16</code> not used</li> <li>• <code>/findtype:32</code> filtert op bug ID</li> <li>• <code>/findtype:64</code> not used</li> <li>• <code>/findtype:128</code> filter by date</li> <li>• <code>/findtype:256</code> filter by date range</li> </ul> </li> <li>• If <code>/outfile:path\to\file</code> is specified, the selected revisions are written to that file when the log dialog is closed. The revisions are written in the same format as is used to specify revisions in the merge dialog.</li> </ul> <p>An svn date revision can be in one of the following formats:</p> <ul style="list-style-type: none"> <li>• <code>{2006-02-17}</code></li> </ul>

Commando	Beschrijving
	<ul style="list-style-type: none"> <li>• {15:30}</li> <li>• {15:30:00.200000}</li> <li>• {"2006-02-17 15:30"}</li> <li>• {"2006-02-17 15:30 +0230"}</li> <li>• {2006-02-17T15:30}</li> <li>• {2006-02-17T15:30Z}</li> <li>• {2006-02-17T15:30-04:00}</li> <li>• {20060217T1530}</li> <li>• {20060217T1530Z}</li> <li>• {20060217T1530-0500}</li> </ul>
:checkout	<p>Opens the checkout dialog. The <code>/path</code> specifies the target directory and the <code>/url</code> specifies the URL to checkout from. If you specify the key <code>/blockpathadjustments</code>, the automatic checkout path adjustments are blocked. The <code>/revision:XXX</code> specifies the revision to check out.</p> <p>If you specify <code>/outfile:"path/to/file"</code> the specified file will contain three lines after a checkout. The first line is the checkout path, the second line the url and the third the revision.</p>
:import	<p>Opens the import dialog. The <code>/path</code> specifies the directory with the data to import. You can also specify the <code>/logmsg</code> switch to pass a predefined log message to the import dialog. Or, if you don't want to pass the log message on the command line, use <code>/logmsgfile:path</code>, where <code>pad</code> points to a file containing the log message.</p>
:update	<p>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>, <code>/ignoreexternals</code> and <code>/includeexternals</code>. The <code>/stickydepth</code> indicates that the specified depth should be sticky, creating a sparse checkout. The <code>/skipprechecks</code> can be set to skip all checks that are done before an update. If this is specified, then the <code>Toon logboek</code> button is disabled, and the context menu to show diffs is also disabled after the update.</p>
:commit	<p>Opent het vastleggen venster. De <code>/path</code> geeft de doelmap of de lijst met bestanden op om vast te leggen. U kunt ook de <code>/logmsg</code> optie aangeven om een voorgedefiniëerd logbericht aan het vastleggen venster door te geven. Of, als u geen logbericht op de commandoregel wilt doorgeven gebruikt u dan <code>/logmsgfile:path</code>, waarbij <code>pad</code> een bestand aangeeft dat het logbericht bevat. Om het bug ID veld van tevoren te vullen (voor het geval dat u integratie met bug trackers naar behoren heeft ingesteld) kunt u de <code>/bugid:"het id van de bug hier"</code> gebruiken.</p>
:add	<p>Voegt de bestanden in <code>/path</code> toe aan versiebeheer.</p>
:revert	<p>Draait alle locale aanpassingen aan de werkkopie terug. De <code>/path</code> geeft aan welke items terug te draaien.</p>
:cleanup	<p>Cleans up interrupted or aborted operations and unlocks the working copy in <code>/path</code>. You also have to pass the <code>/cleanup</code> to actually do the cleanup. Use <code>/noui</code> to prevent the result dialog from popping up (either telling about the cleanup being finished or showing an error message). <code>/noprogessui</code> also disables the progress dialog. <code>/nodlg</code> disables showing the cleanup dialog where the user can</p>



Commando	Beschrijving
	choose what exactly should be done in the cleanup. The available actions can be specified with the options <code>/cleanup</code> for status cleanup, <code>/breaklocks</code> to break all locks, <code>/revert</code> to revert uncommitted changes, <code>/delunversioned</code> , <code>/delignored</code> , <code>/refreshshell</code> , <code>/externals</code> , <code>/fixtimestamps</code> and <code>/vacuum</code> .
<code>:resolve</code>	Markeert een conflicterend bestand dat wordt aangegeven in <code>/path</code> als opgelost. Als <code>/noquestion</code> is ingesteld, dan wordt het probleem opgelost zonder de gebruiker van tevoren te raadplegen of het daadwerkelijk moet worden uitgevoerd.
<code>:reprocreate</code>	Maakt een archief aan in <code>/path</code>
<code>:switch</code>	Opens the switch dialog. The <code>/path</code> specifies the target directory and <code>/url</code> the URL to switch to.
<code>:export</code>	Exports the working copy in <code>/path</code> to another directory. If the <code>/path</code> points to an unversioned directory, a dialog will ask for an URL to export to the directory in <code>/path</code> . If you specify the key <code>/blockpathadjustments</code> , the automatic export path adjustments are blocked.
<code>:dropexport</code>	Exports the working copy in <code>/path</code> to the directory specified in <code>/droptarget</code> . This exporting does not use the export dialog but executes directly. The option <code>/overwrite</code> specifies that existing files are overwritten without user confirmation, and the option <code>/autorename</code> specifies that if files already exist, the exported files get automatically renamed to avoid overwriting them. The option <code>/extended</code> can specify either <code>localchanges</code> to only export files that got changed locally, or <code>unversioned</code> to also export all unversioned items as well.
<code>:dropvendor</code>	Copies the folder in <code>/path</code> recursively to the directory specified in <code>/droptarget</code> . New files are added automatically, and missing files get removed in the target working copy, basically ensuring that source and destination are exactly the same. Specify <code>/noui</code> to skip the confirmation dialog, and <code>/noprogessui</code> to also disable showing the progress dialog.
<code>:merge</code>	Opens the merge dialog. The <code>/path</code> specifies the target directory. For merging a revision range, the following options are available: <code>/fromurl:URL</code> , <code>/revrange:string</code> . For merging two repository trees, the following options are available: <code>/fromurl:URL</code> , <code>/tourl:URL</code> , <code>/fromrev:xxx</code> and <code>/torev:xxx</code> .
<code>:mergeall</code>	Opent het samenvoeg scherm. De <code>/path</code> geeft de doeldirectory aan.
<code>:copy</code>	Brings up the branch/tag dialog. The <code>/path</code> is the working copy to branch/tag from. And the <code>/url</code> is the target URL. If the url starts with a <code>^</code> it is assumed to be relative to the repository root. To already check the option <code>Switch working copy to new branch/tag</code> you can pass the <code>/switchaftercopy</code> switch. To check the option <code>Tussenliggende mappen aanmaken</code> pass the <code>/makeparents</code> switch. You can also specify the <code>/logmsg</code> switch to pass a predefined log message to the branch/tag dialog. Or, if you don't want to pass the log message on the command line, use <code>/logmsgfile:path</code> , where <code>pad</code> points to a file containing the log message.
<code>:settings</code>	Opent het instellingen scherm.
<code>:remove</code>	Verwijdert één of meerdere bestanden van versiebeheer.
<code>:rename</code>	Hernoemt het bestand in <code>/path</code> . De nieuwe naam voor het bestand wordt in een venster gevraagd. Om de vraag om gelijksoortige bestanden in dezelfde handeling te hernoemen te voorkomen, geeft u <code>/noquestion</code> op.
<code>:diff</code>	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. If the specified file also has property modifications,

Commando	Beschrijving
	the external diff tool is also started for each modified property. To prevent that, pass the option <code>/ignoreprops</code> . To explicitly set the revision numbers use <code>/startrev:xxx</code> and <code>/endrev:xxx</code> , and for the optional peg revision use <code>/pegrevision: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. The parameter <code>/line:xxx</code> specifies the line to jump to when the diff is shown.
<code>:shelve</code>	Shelves the specified paths in a new shelf. The option <code>/shelfname:name</code> specifies the name of the shelf. An optional log message can be specified with <code>/logmsg:message</code> . If option <code>/checkpoint</code> is passed, the modifications of the files are kept.
<code>:unshelve</code>	Applies the shelf with the name <code>/shelfname:name</code> to the working copy path. By default the last version of the shelf is applied, but you can specify a version with <code>/version:X</code> .
<code>:showcompare</code>	<p>Afhankelijk van de URL's en de revisienummers welke moeten worden vergeleken, toont dit of een unified diff (als de optie <code>unified</code> is ingesteld), of een lijst met bestanden welke zijn gewijzigd, of start de vergelijkingskijker als de URL's verwijzen naar twee bestanden.</p> <p>De opties <code>url1</code>, <code>url2</code>, <code>revision1</code> en <code>revision2</code> dienen te zijn opgegeven. De opties <code>pegrevision</code>, <code>ignoreancestry</code>, <code>verklaren</code> en <code>unified</code> zijn optioneel.</p> <p>If the specified url also has property modifications, the external diff tool is also started for each modified property. To prevent that, pass the option <code>/ignoreprops</code>.</p> <p>If a unified diff is requested, an optional <code>prettyprint</code> option can be specified which will show the merge-info properties in a more user readable format.</p>
<code>:conflicteditor</code>	Start de conflict bewerker, welke is ingesteld in de TortoiseSVN instellingen, met de correcte bestanden voor het conflicterende bestand in <code>/path</code> .
<code>:relocate</code>	Opent het verhuis scherm. De <code>/path</code> geeft de werkkopie aan om te verhuizen.
<code>:help</code>	Opent het help bestand.
<code>:repostatus</code>	Opent het scherm voor controle op wijzigingen. De <code>/path</code> geeft de map van de werkkopie aan. Als <code>/remote</code> is opgegeven maakt het scherm bij opstarten direct contact met het archief alsof de gebruiker op de Controleer archief knop heeft geklikt.
<code>:repobrowser</code>	<p>Starts the repository browser dialog, pointing to the URL of the working copy given in <code>/path</code> or <code>/path</code> points directly to an URL.</p> <p>An additional option <code>/rev:xxx</code> can be used to specify the revision which the repository browser should show. If the <code>/rev:xxx</code> is omitted, it defaults to HEAD.</p> <p>If <code>/path</code> points to an URL, the <code>/projectpropertiespath:path/to/wc</code> specifies the path from where to read and use the project properties.</p> <p>If <code>/outfile:path\to\file</code> is specified, the selected URL and revision are written to that file when the repository browser is closed. The first line in that text file contains the URL, the second line the revision in text format.</p>
<code>:ignore</code>	Voegt alle doelen in <code>/path</code> toe aan de negeren lijst. Oftewel, voegt de <code>svn:ignore</code> eigenschap toe aan die bestanden.
<code>:blame</code>	Opent het verklaar scherm voor het bestand dat wordt gegeven in <code>/path</code> .

Commando	Beschrijving
	<p>Als de opties <code>/startrev</code> en <code>/endrev</code> zijn ingesteld wordt niet het scherm dat het te verklaren bereik vraagt getoond, maar worden de waarden van deze opties in de revisie gebruikt.</p> <p>Als de optie <code>/line:nnn</code> is ingesteld zal TortoiseBlame worden geopend met het aangegeven regelnummer in beeld.</p> <p>De opties <code>/ignoreeol</code>, <code>/ignorespaces</code> en <code>/ignoreallspaces</code> worden ook ondersteund.</p>
<code>:cat</code>	Slaat een bestand op vanuit een URL of werkkopiepad gegeven in <code>/path</code> naar de locatie gegeven in <code>/savepath:path</code> . De revisie wordt gegeven in <code>/revision:xxx</code> . Dit kan worden gebruikt om een bestand met van een bepaalde revisie op te halen.
<code>:createpatch</code>	Creates a patch file for the path given in <code>/path</code> . To skip the file Save-As dialog you can pass <code>/savepath:path</code> to specify the path where to save the patch file to directly. To prevent the unified diff viewer from being started showing the patch file, pass <code>/noview</code> . If a unified diff is requested, an optional <code>prettyprint</code> option can be specified which will show the merge-info properties in a more user readable format.
<code>:revisiongraph</code>	<p>Shows the revision graph for the path given in <code>/path</code>.</p> <p>To create an image file of the revision graph for a specific path, but without showing the graph window, pass <code>/output:path</code> with the path to the output file. The output file must have an extension that the revision graph can actually export to. These are: <code>.svg</code>, <code>.wmf</code>, <code>.png</code>, <code>.jpg</code>, <code>.bmp</code> and <code>.gif</code>.</p> <p>Since the revision graph has many options that affect how it is shown, you can also set the options to use when creating the output image file. Pass these options with <code>/options:XXXX</code>, where <code>XXXX</code> is a decimal value. The best way to find the required options is to start the revision graph the usual way, set all user-interface options and close the graph. Then the options you need to pass on the command line can be read from the registry <code>HKCU\Software\TortoiseSVN\RevisionGraphOptions</code>.</p>
<code>:lock</code>	Blokkeert één of meerdere bestanden in de in <code>/path</code> gegeven map. Het 'blokkeer' venster wordt getoond zodat de gebruiker commentaar op de blokkade kan ingeven.
<code>:unlock</code>	Verwijdert de blokkering van één of meerdere bestanden in de in <code>/path</code> gegeven map.
<code>:rebuildiconcache</code>	Bouwt de windows pictogrammen cache opnieuw op. Gebruik dit alleen voor het geval dat de windows pictogrammen corrupt zijn geraakt. Een bijwerking hiervan is (wat niet kan worden voorkomen) dat de pictogrammen op het werkblad worden herschikt. Geef <code>/noquestion</code> mee om het meldingsscherm te onderdrukken.
<code>:properties</code>	<p>Shows the properties dialog for the path given in <code>/path</code>.</p> <p>For dealing with versioned properties this command requires a working copy.</p> <p>Revision properties can be viewed/changed if <code>/path</code> is an URL and <code>/rev:xxx</code> is specified.</p> <p>To open the properties dialog directly for a specific property, pass the property name as <code>/property:name</code>.</p>
<code>:sync</code>	Exports/imports settings, either depending on whether the current settings or the exported settings are newer, or as specified.

Commando	Beschrijving
	<p>If a path is passed with <code>/path</code>, then the path is used to store or read the settings from.</p> <p>The parameter <code>/askforpath</code> will show a file open/save dialog for the user to chose the export/import path.</p> <p>If neither <code>/load</code> nor <code>/save</code> is specified, then TortoiseSVN determines whether to export or import the settings by looking at which ones are more recent. If the export file is more recent than the current settings, then the settings are loaded from the file. If the current settings are more recent, then the settings are exported to the settings file.</p> <p>If <code>/load</code> is specified, the settings are imported from the settings file.</p> <p>If <code>/save</code> is specified, the current settings are exported to the settings file.</p> <p>The parameter <code>/local</code> forces a settings export to include local settings, i.e. settings that refer to local paths.</p>

**Tabel D.1. Lijst van alle beschikbare commando's en opties**

Voorbeelden (welke op één regel dienen te worden ingegeven):

```
TortoiseProc.exe /command:commit
                 /path:"c:\svn_wc\file1.txt*c:\svn_wc\file2.txt"
                 /logmsg:"test logbericht" /closeonend:0
```

```
TortoiseProc.exe /command:update /path:"c:\svn_wc\" /closeonend:0
```

```
TortoiseProc.exe /command:log /path:"c:\svn_wc\file1.txt"
                 /startrev:50 /endrev:60 /closeonend:0
```

## D.2. Tsvncmd URL verwerker

Door speciale URL's te gebruiken is het ook mogelijk om TortoiseProc vanuit een webpagina aan te roepen.

TortoiseSVN registreert een nieuw protocol `tsvncmd`: wat kan worden gebruikt om snelkoppelingen aan te maken welke TortoiseSVN commando's uitvoeren. De commando's en parameters zijn dezelfde als wanneer TortoiseSVN vanaf de commandoregel wordt geautomatiseerd.

De opbouw van de `tsvncmd`: URL is als volgt:

```
tsvncmd:command:cmd?parameter:parameterwaarde?parameter:parameterwaarde
```

with `cmd` being one of the allowed commands, `parameter` being the name of a parameter like `path` or `revision`, and `paramvalue` being the value to use for that parameter. The list of parameters allowed depends on the command used.

De volgende commando's zijn toegestaan met `tsvncmd`: URL's:

- `:update`
- `:commit`
- `:diff`
- `:repobrowser`

- :checkout
- :export
- :blame
- :repostatus
- :revisiongraph
- :showcompare
- :log
- :properties

A simple example URL might look like this:

```
<a href="tsvncmd:command:update?path:c:\svn_wc?rev:1234">Update</a>
```

or in a more complex case:

```
<a href="tsvncmd:command:showcompare?
url1:https://svn.code.sf.net/p/stefanstools/code/trunk/StExBar/src/setup/Setup.wxs?
url2:https://svn.code.sf.net/p/stefanstools/code/trunk/StExBar/src/setup/Setup.wxs?
revision1:188?revision2:189">compare</a>
```

### D.3. TortoiseIDiff Commando's

Het vergelijkingshulpprogramma voor afbeeldingen heeft een aantal commando-regel opties welke kunnen worden gebruikt om te bepalen hoe het hulpprogramma wordt opgestart. Het programma wordt `TortoiseIDiff.exe` genoemd.

De onderstaande tabel toont alle opties welke op de commando-regel kunnen worden doorgegeven aan het vergelijkingshulpprogramma voor afbeeldingen.

Optie	Beschrijving
:left	Pad naar het bestand dat links wordt getoond.
:lefttitle	Een titel tekst. Deze tekst wordt gebruikt in de titel van de afbeeldingsweergave in plaats van het volledige pad naar het afbeeldingsbestand.
:right	Pad naar het bestand dat rechts wordt getoond.
:righttitle	Een titel tekst. Deze tekst wordt gebruikt in de titel van de afbeeldingsweergave in plaats van het volledige pad naar het afbeeldingsbestand.
:overlay	Indien opgegeven schakelt het vergelijkingshulpprogramma voor afbeeldingen over naar de overlap modus (alfa mengen).
:fit	Indien opgegeven maakt het vergelijkingshulpprogramma voor afbeeldingen beide afbeeldingen aan elkaar gelijk qua grootte.
:showinfo	Toont het informatieblok voor de afbeelding.

#### Tabel D.2. Lijst van beschikbare opties.

Voorbeeld (welke op één regel dient te worden ingegeven):

```
TortoiseIDiff.exe /left:"c:\images\img1.jpg" /lefttitle:"image 1"
```

```
/right:"c:\images\img2.jpg" /righttitle:"image 2"  
/fit /overlay
```

## D.4. TortoiseUDiff Commands

The unified diff viewer has only two command line options:

Optie	Beschrijving
:patchfile	Path to the unified diff file.
:p	Activates pipe mode. The unified diff is read from the console input.

### Tabel D.3. Lijst van beschikbare opties.

Examples (which should be entered on one line):

```
TortoiseUDiff.exe /patchfile:"c:\diff.patch"
```

If you create the diff from another command, you can use TortoiseUDiff to show that diff directly:

```
svn diff | TortoiseUDiff.exe /u
```

this also works if you omit the /p parameter:

```
svn diff | TortoiseUDiff.exe
```

---

# Bijlage E. Opdrachtprompt Interface Cross Reference

Deze handleiding verwijst soms naar de Subversion documentatie, die de Subversion opdrachtprompt interface (Command Line Interface, CLI) beschrijft. Om een indruk te geven van wat TortoiseSVN op de achtergrond doet, hebben we een lijst gemaakt van de equivalente CLI commando's voor elke TortoiseSVN GUI actie.

## Opmerking

Onthoudt dat, ondanks dat er CLI equivalenten zijn voor wat TortoiseSVN doet, TortoiseSVN de bibliotheken direct benadert en *niet* de CLI commando's gebruikt.

Als je denkt dat je een bug in TortoiseSVN hebt gevonden, kan het zijn dat we je vragen om het probleem te reproduceren met CLI commando's. We kunnen dan makkelijk een onderscheid maken tussen TortoiseSVN en Subversion problemen. Dit overzicht verteld je welke commando's je moet gebruiken.

## E.1. Conventies en Basis Regels

In the descriptions which follow, the URL for a repository location is shown simply as URL, and an example might be `https://svn.code.sf.net/p/tortoisesvn/code/trunk/`. The working copy path is shown simply as PATH, and an example might be `C:\TortoiseSVN\trunk`.



## Belangrijk

Omdat TortoiseSVN een Windows Shell uitbreiding is, heeft het geen idee van de huidige werkkopie map. Alle paden van werkkopieën moeten als absolute paden opgegeven worden, niet als relatieve paden.

Bepaalde onderdelen zijn optioneel. Deze zijn dan ook meestal als selectievak en radiobutton in TortoiseSVN beschikbaar. Deze opties worden weergegeven tussen [blokhaken] in de definitie van de opdrachtregel.

## E.2. TortoiseSVN commando's

### E.2.1. Ophalen

```
svn checkout [-depth ARG] [--ignore-externals] [-r rev] URL PATH
```

De diepte combo box objecten hebben betrekking op het `-depth` optie.

Als Externen uitsluiten aangevinkt is, gebruik dan de `--ignore-externals` optie.

Als je een specifieke revisie ophaalt, specificeer deze dan na de URL met de `-r` optie.

### E.2.2. Verversen

```
svn info URL_of_WC
svn update [-r rev] PATH
```

Het verversen van meerdere objecten is op dit moment niet een enkele actie in Subversion. TortoiseSVN zoekt eerst de HEAD revisie van het archief en ververset dan alle objecten naar die specifieke revisie om te voorkomen dat er een werkkopie ontstaat met verschillende revisies.

Als maar één object geselecteerd is voor het verversen of de geselecteerde objecten zijn niet allemaal uit hetzelfde archief, dan zal TortoiseSVN gewoon naar de HEAD revisie verversen.

Er worden geen opdrachtregel opties gebruikt hierbij. Verversen naar revisie gebruikt ook het verversen commando, maar biedt meer opties.

### E.2.3. Verversen naar Revisie

```
svn info URL_of_WC
svn update [-r rev] [-depth ARG] [--ignore-externals] PATH
```

De diepte combo box objecten hebben betrekking op het `-depth` optie.

Als Externen uitsluiten aangevinkt is, gebruik dan de `--ignore-externals` optie.

### E.2.4. Vastleggen

Het vastleg venster van TortoiseSVN gebruikt meerdere Subversion commando's. In de eerste fase wordt een status controle uitgevoerd om te bepalen welke objecten in je werkkopie mogelijk vastgelegd moeten worden. Je kunt de lijst bekijken, bestanden vergelijken ten opzichte van de BASE en de objecten selecteren die je mee wilt nemen bij het vastleggen.

```
svn status -v PATH
```

Als Toon bestanden zonder versiebeheer aangevinkt is, zal TortoiseSVN alle bestanden en mappen zonder versiebeheer in je werkkopie laten zien, rekening houdend met de neger regels. Deze functie heeft geen direct equivalent in Subversion, omdat het `svn status` commando niet afdalt in mappen zonder versiebeheer.

Als je bestanden en mappen zonder versiebeheer controleert, dan zullen die objecten eerst aan je werkkopie worden toegevoegd.

```
svn add PATH...
```

Als je op OK klikt, zal de Subversion vastlegging plaatsvinden. Als je alle bestandsselectievakken in de standaard stand hebt laten staan, zal TortoiseSVN een enkele recursieve vastlegging van je werkkopie uitvoeren. Als je enkele bestanden selecteert, dan moet een niet-recursieve vastlegging (`-N`) gebruikt worden en moet elke pad apart gespecificeerd worden in de vastlegging opdrachtregel.

```
svn commit -m "LogMessage" [-depth ARG] [--no-unlock] PATH...
```

LogMessage vertegenwoordigt hier de inhoud van het logboek boodschap invoerveld. Deze kan leeg gelaten worden.

Als Blokkade handhaven aangevinkt is, gebruik dan de `--no-unlock` optie.

### E.2.5. Verschil

```
svn diff PATH
```

Als je Verschil gebruikt vanaf het hoofd context menu, dan ben je een gewijzigd bestand aan het vergelijken met de BASE revisie. De uitkomst van bovenstaande CLI commando voert ook deze actie uit en levert het resultaat



op in een unified-diff formaat. Dit is echter niet wat TortoiseSVN gebruikt. TortoiseSVN gebruikt TortoiseMerge (of een ander vergelijk programma waarvoor jij gekozen hebt) voor het visueel weergeven van verschillen tussen volledig op tekst gebaseerde bestanden. Er is hier dus geen CLI equivalent voor.

Je kunt ook 2 willekeurige bestanden vergelijken met TortoiseSVN, onafhankelijk of deze versiebeheer hebben. TortoiseSVN geeft de twee bestanden door aan het gekozen vergelijkingsprogramma en laat dat programma de verschillen uitzoeken.

## E.2.6. Toon logboek

```
svn log -v -r 0:N --limit 100 [--stop-on-copy] PATH
or
svn log -v -r M:N [--stop-on-copy] PATH
```

Standaard beperkt TortoiseSVN het aantal logboek boodschappen tot 100 met de `--limit` methode. Als de instellingen aangeven om oude API's te gebruiken, dan wordt de tweede indeling gebruikt voor het ophalen van de logboek boodschappen voor 100 archief revisies.

Als Stoppen bij kopie/hernoemd bestand is aangevinkt, gebruik dan de `--stop-on-copy` optie.

## E.2.7. Controleer op Wijzigingen

```
svn status -v PATH
or
svn status -u -v PATH
```

De initiële controle van de statussen kijkt alleen naar je werkkopie. Als je op **Controleer archief** klikt, dan wordt er ook in het archief gekeken welke bestanden er veranderen zullen bij het verversen. Hiervoor wordt de `-u` optie gebruikt.

Als **Toon bestanden zonder versiebeheer** aangevinkt is, zal TortoiseSVN alle bestanden en mappen zonder versiebeheer in je werkkopie laten zien, rekening houdend met de neger regels. Deze functie heeft geen directe equivalent in Subversion, omdat het `svn status` commando niet afdaalt in mappen zonder versiebeheer.

## E.2.8. Revisie schema

Het revisie schema is een functie van TortoiseSVN. Er is een CLI equivalent.

What TortoiseSVN does is an

```
svn info URL_of_WC
svn log -v URL
```

where URL is the repository *root* and then analyzes the data returned.

## E.2.9. Archief Verkenner

```
svn info URL_of_WC
svn list [-r rev] -v URL
```

Je kunt `svn info` gebruiken om de basis map van het archief te vinden, dit is de bovenste map die in de archief-verkenner wordt weergegeven. Je kunt niet Hoger dan deze map komen in de mapstructuur. Dit commando geeft ook alle blokkade informatie weer die in de archief-verkenner wordt weergegeven.

De `svn list` opdracht zal een lijst met de inhoud van een map opleveren, van de ingegeven URL en revisie.

### E.2.10. Bewerk conflicten

Dit commando heeft geen CLI equivalent. Het roept TortoiseMerge of een ander extern 3-weg vergelijkingsprogramma om de naar de betrokken bestanden te kunnen kijken en te bepalen welke regels er gebruikt moeten worden.

### E.2.11. Opgelost

```
svn resolved PATH
```

### E.2.12. Hernoemen

```
svn rename CURR_PATH NEW_PATH
```

### E.2.13. Verwijderen

```
svn delete PATH
```

### E.2.14. Ongedaan maken

```
svn status -v PATH
```

De eerste fase is een status controle, welke de objecten zoekt die mogelijk ongedaan gemaakt kunnen worden. Je kunt de lijst bekijken, bestanden vergelijken ten opzichte van de BASE en de objecten selecteren die je ongedaan wilt maken.

Als je op OK klikt, zal Subversion de actie uitvoeren. Als je alle bestandsselectievakken in de standaard stand hebt laten staan, zal TortoiseSVN een enkele recursieve (-R) ongedaanmaking van je werkkopie uitvoeren. Als je enkele bestanden de-selecteert, dan moet elk pad apart gespecificeerd worden in de opdrachtregel.

```
svn revert [-R] PATH...
```

### E.2.15. Schonen

```
svn cleanup PATH
```

### E.2.16. Plaats Blokkade

```
svn status -v PATH
```

De eerste fase is een status controle, welke de objecten zoekt in je werkkopie die mogelijk geblokkeerd kunnen worden. Je kunt de objecten die je wilt blokkeren selecteren.

```
svn lock -m "LockMessage" [--force] PATH...
```

Hierbij is `LockMessage` het bericht in het invoerveld. Dit kan leeg gelaten worden.

Als `Steel de blokkades` is aangevinkt, gebruik dan de `--force` optie.

### E.2.17. Blokkade Opheffen

```
svn unlock PATH
```

### E.2.18. Tak/Label

```
svn copy -m "LogMessage" URL URL
or
svn copy -m "LogMessage" URL@rev URL@rev
or
svn copy -m "LogMessage" PATH URL
```

Het Tak/Label scherm voert een kopieer actie uit op het archief. Er zijn drie radioknop opties:

- HEAD revisie in het archief
- Specifieke revisie in het archief
- Werkkopie

welke overeenkomen met de 3 eerder genoemde commando varianten.

`LogMessage` vertegenwoordigt hier de inhoud van het logboek boodschap invoerveld. Deze kan leeg gelaten worden.

### E.2.19. Optie

```
svn info URL_of_WC
svn switch [-r rev] URL PATH
```

### E.2.20. Samenvoegen

```
svn merge [--dry-run] --force From_URL@revN To_URL@revM PATH
```

De `Test samenvoegen` knop voert dezelfde samenvoeging uit, maar dan met de `--dry-run` optie.

```
svn diff From_URL@revN To_URL@revM
```

De `Voer Unified Diff` knop toont de verschil actie, welke gebruikt wordt om de samenvoeging uit te voeren.

### E.2.21. Exporteren

```
svn export [-r rev] [--ignore-externals] URL Export_PATH
```

Deze formattering wordt gebruikt als de actie op een map zonder versiebeheer wordt opgestart en de map dus als doel map gebruikt wordt.

Het exporteren van een werkkopie naar een andere locatie wordt zonder de Subversion bibliotheek te gebruiken uitgevoerd. Er is dus geen CLI equivalent.

TortoiseSVN kopieert alle bestanden naar de nieuwe locatie terwijl de voortgang wordt weergegeven. Bestanden/mappen zonder versiebeheer kunnen optioneel ook geëxporteerd worden.

Als in één van beide gevallen de Externen uitsluiten aangevinkt is, gebruik dan de `--ignore-externals` optie.

### E.2.22. Locatie wijzigen

```
svn switch --relocate From_URL To_URL
```

### E.2.23. Maak Archief Hier

```
svnadmin create --fs-type fsfs PATH
```

### E.2.24. Toevoegen

```
svn add PATH...
```

Als je een map geselecteerd hebt, zal TortoiseSVN deze eerst recursief controleren op objecten die toegevoegd kunnen worden.

### E.2.25. Importeren

```
svn import -m LogMessage PATH URL
```

LogMessage vertegenwoordigt hier de inhoud van het logboek boodschap invoerveld. Deze kan leeg gelaten worden.

### E.2.26. Verklaren

```
svn blame -r N:M -v PATH  
svn log -r N:M PATH
```

Als je TortoiseBlame gebruikt om de Verklaren informatie te bekijken, dan is het bestand logboek ook nodig om de logboek boodschappen in een tooltip te laten zien. Als je de Verklaren informatie als een tekstbestand bekijkt, dan is deze informatie niet nodig.

### E.2.27. Voeg aan Negeerlijst toe

```
svn propget svn:ignore PATH > tempfile  
{edit new ignore item into tempfile}
```

```
svn propset svn:ignore -F tempfile PATH
```

Omdat de `svn:ignore` eigenschap vaak meerdere regels heeft, wordt het hier weergegeven alsof dit gewijzigd is met een tekstbestand in plaats van direct mee te geven in de opdrachtregel.

### **E.2.28. Maak patch**

```
svn diff PATH > patch-file
```

TortoiseSVN maakt een patch bestand aan in unified diff indeling door de werkkopie met de BASE revisie te vergelijken.

### **E.2.29. Patch toepassen**

Het toepassen van patches is een riskante activiteit, tenzij de patch en de werkkopie van dezelfde revisie zijn. Gelukkig kun je TortoiseMerge gebruiken, waarvoor geen direct equivalent beschikbaar is in Subversion.

---

# Bijlage F. Implementatie details

Deze bijlage bevat een gedetailleerde verhandeling over de implementatie van enkele mogelijkheden van TortoiseSVN.

## F.1. Overlappende Pictogrammen

Ieder bestand en map heeft een Subversion statuswaarde zoals gerapporteerd door de Subversion bibliotheek. In de commandoregel client worden deze vertegenwoordigd door enkelvoudige karakter codes, maar in TortoiseSVN worden ze grafisch als een overlappend pictogram weergegeven. Omdat het aantal overlappende pictogrammen erg beperkt is, vertegenwoordigd elk overlappend pictogram één van meerdere statuswaarden.



Het *Conflicterend* overlappende pictogram vertegenwoordigt de `conflict` status, waar een bijwerking of wisseling leidt tot conflicten tussen lokale wijzigingen en wijzigingen welke vanuit het archief zijn binnen gehaald. Het wordt ook gebruikt om de *belemmering* status aan te duiden, welke zich kan voordoen wanneer een activiteit niet kan worden voltooid.



Het *Aangepast* overlappende pictogram vertegenwoordigt de *aangepast* status, waar u lokale wijzigingen heeft aangebracht, de *samengevoegd* status, waar wijzigingen vanuit het archief zijn samen gevoegd met de lokale wijzigingen en de *vervange* status, waar een bestand is verwijderd en vervangen door een ander bestand met dezelfde naam.



Het *Verwijderd* overlappende pictogram vertegenwoordigt de *verwijderd* status, waar een item is klaargezet voor verwijdering, of de *ontbrekend* status, waar een item niet aanwezig is. Een item dat ontbreekt kan zelf natuurlijk geen overlappend pictogram bezitten, maar de bovenliggende map kan worden gemarkeerd wanneer een van de onderliggende items ontbreekt.



Het *Toegevoegd* overlappende pictogram vertegenwoordigt de *toegevoegd* status wanneer een item is toegevoegd aan versiebeheer.



Het *In Subversion* overlappende pictogram wordt gebruikt om een item te vertegenwoordigen dat zich in de *normaal* status bevindt, of een item onder versiebeheer waarvan de status nog niet bekend is. Omdat TortoiseSVN een cacheproces op de achtergrond gebruikt om de status te vergaren, kan het enige seconden in beslag nemen voordat de overlappende pictogrammen worden bijgewerkt.



The *Needs Lock* overlay is used to indicate when a file has the `svn:needs-lock` property set.



Het *Vergrendeld* overlappende pictogram wordt gebruikt wanneer de lokale werkkopie een vergrendeling op het betreffende bestand heeft.



Het *Genegeerd* overlappende pictogram wordt gebruikt om een item te vertegenwoordigen dat zich in de genegeerd status bevindt door een globaal ingesteld negeren patroon, of door de `svn:ignore` eigenschap van de bovenliggende map. Dit overlappende pictogram is optioneel.



Het *Niet onder versiebeheer* overlappende pictogram wordt gebruikt om een item te vertegenwoordigen dat zich in de `niet onder versiebeheer` status bevindt. Dit is een item in een map onder versiebeheer, maar zelf niet onder versiebeheer valt. Dit overlappende pictogram is optioneel.

If an item has Subversion status `none` (the item is not within a working copy) then no overlay is shown. If you have chosen to disable the *Ignored* and *Unversioned* overlays then no overlay will be shown for those files either.

Een item kan slechts één Subversion statuswaarde hebben. Een bestand kan bijvoorbeeld lokaal zijn aangepast en op het zelfde moment zijn gemarkeerd voor verwijdering. Subversion geeft één enkele statuswaarde - in dit geval verwijderd. Deze prioriteiten worden in Subversion zelf gedefinieerd.

Als TortoiseSVN de status recursief toont (standaard instelling), toont elke map een overlappend pictogram dat zijn eigen status en dat van de onderliggende sub-mappen weergeeft. Om een enkel *samenvattend* overlappende pictogram te tonen gebruiken we bovenstaande volgorde in prioriteit om te bepalen welk overlappend pictogram moet worden gebruikt, waarbij het *Conflicterend* overlappende pictogram de hoogste prioriteit heeft.

In fact, you may find that not all of these icons are used on your system. This is because the number of overlays allowed by Windows is limited to 15. Windows uses 4 of those, and the remaining 11 can be used by other applications. If there are not enough overlay slots available, TortoiseSVN tries to be a *Good Citizen (TM)* and limits its use of overlays to give other apps a chance.

Since there are Tortoise clients available for other version control systems, we've created a shared component which is responsible for showing the overlay icons. The technical details are not important here, all you need to know is that this shared component allows all Tortoise clients to use the same overlays and therefore the limit of 11 available slots isn't used up by installing more than one Tortoise client. Of course there's one small drawback: all Tortoise clients use the same overlay icons, so you can't figure out by the overlay icons what version control system a working copy is using.

- *Normaal*, *Aangepast* en *Conflicterend* worden altijd geladen en zijn altijd zichtbaar.
- *Verwijderd* wordt indien mogelijk geladen, maar valt terug naar *Aangepast* als er niet genoeg ingangen zijn.
- *Alleen-lezen* wordt indien mogelijk geladen, maar valt terug op *Normaal* als er niet genoeg ingangen zijn.
- *Locked* is loaded if possible, but falls back to *Normal* if there are not enough slots.
- *Added* is loaded if possible, but falls back to *Modified* if there are not enough slots.

---

# Bijlage G. Taalpakketten en spellingscontrole

The standard installer has support only for English, but you can download separate language packs and spell check dictionaries separately after installation.

## G.1. Taalpakketten

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](https://tortoisesvn.net/translation_status_dev.html) [https://tortoisesvn.net/translation\_status\_dev.html]. And if there is no language pack available, why not join the team and submit your own translation ;-)

Each language pack is packaged as a .msi installer. Just run the install program and follow the instructions. After the installation finishes, the translation will be available.

The documentation has also been translated into several different languages. You can download translated manuals from the [support page](https://tortoisesvn.net/support.html) [https://tortoisesvn.net/support.html] on our website.

## G.2. Spellingscontrole

TortoiseSVN uses the Windows spell checker if it's available (Windows 8 or later). Which means that if you want the spell checker to work in a different language than the default OS language, you have to install the spell checker module in the Windows settings (Settings > Time & Language > Region & Language).

TortoiseSVN will use that spell checker if properly configured with the `tsvn:projectlanguage` project property.

In case the Windows spell checker is not available, TortoiseSVN can also use spell checker dictionaries from [OpenOffice](https://openoffice.org) [https://openoffice.org] and [Mozilla](https://mozilla.org) [https://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. After the installation finishes, the dictionary will be available too.

Or you can install the dictionaries yourself. If you have OpenOffice or Mozilla installed, you can copy those dictionaries, which are located in the installation folders for those applications. Otherwise, you need to download the required dictionary files from <http://wiki.services.openoffice.org/wiki/Dictionaries>.

Once you have got the dictionary files, you probably need to rename them so that the filenames only have the locale chars in it. Example:

- en\_US.aff
- en\_US.dic

Then just copy them into the `%APPDATA%\TortoiseSVN\dic` folder. If that folder isn't there, you have to create it first. TortoiseSVN will also search the Languages sub-folder of the TortoiseSVN installation folder (normally this will be `C:\Program Files\TortoiseSVN\Languages`); this is the place where the language packs put their files. However, the `%APPDATA%`-folder doesn't require administrator privileges and, thus, has higher priority. The next time you start TortoiseSVN, the spell checker will be available.

Als je meerdere woordenboeken installeert, gebruikt TortoiseSVN deze regels om te selecteren welke te gebruiken.

1. Controleer de `tsvn:projectlanguage` instelling. Refereer naar [Paragraaf 4.18, "Project Instellingen"](#) voor informatie over het instellen van projecteigenschappen.



2. Als er geen projecttaal is ingesteld, of als die taal niet is geïnstalleerd, probeer dan de taal die hoort bij de Windows landinstelling.
3. If the exact Windows locale doesn't work, try the "Base" language, e.g. de\_CH (Swiss-German) falls back to de\_DE (German).
4. Als bovenstaande oplossingen niet werken, dan is de standaardtaal Engels, welke is opgenomen in de standaardinstallatie.

---

# Verklarende woordenlijst

Archief	Een archief is een centrale plaats waar data opgeslagen en beheerd wordt. Een archief kan een plaats zijn waar meerdere databases of bestanden zijn opgeslagen om gedistribueerd te worden over een netwerk. Een archief kan ook een locatie zijn die direct toegankelijk is voor een gebruiker, zonder dat er gegevens over een netwerk verstuurd hoeven te worden (oftewel een locatie op een lokale harde schijf).
BASE revisie	De huidige basis revisie van een bestand of map in je <i>werkkopie</i> . Dit is de revisie van een bestand of map van de laatste ophaal actie, verversing of vastlegging. De BASE revisie is normaal gesproken niet gelijk aan de HEAD revisie (er kunnen in de tussentijd alweer wijzigingen in het archief vastgelegd zijn, waardoor de lokale bestanden achter lopen op de HEAD revisie in het archief).
Blokkade	Als je een blokkade plaatst op een object met versiebeheer, dan markeer je dit object in het archief als niet-vast-te-leggen, behalve vanuit de werkkopie waar vandaan de blokkade geplaatst is.
Conflict	Als wijzigingen in het archief worden samengevoegd met je lokale wijzigingen, dan kan het zijn dat er wijzigingen op dezelfde regels zijn doorgevoerd. In zo'n situatie kan Subversion niet automatisch beslissen welke versie gebruikt moet worden en markeert het bestand als in conflict zijnde. Je moet het conflict dan handmatig oplossen (eventueel met behulp van TortoiseMerge), voordat je wijzigingen weer kunt vastleggen.
Eigenschap	Naast het toepassen van versiebeheer voor je mappen en bestanden, geeft Subversion je ook de mogelijkheid om metadata met versiebeheer te gebruiken - aangeduid als "eigenschappen" voor elke map en bestand onder versiebeheer. Elke eigenschap heeft een naam en waarde, net zoals een register sleutel. Subversion maakt intern gebruik van enkele speciale eigenschappen, zoals <code>svn:eol-style</code> . TortoiseSVN heeft er ook enkele, zoals bijvoorbeeld <code>tsvn:logminsize</code> . Je kunt ook zelf eigenschappen toevoegen, waarvoor je zelf de naam en waarde kunt kiezen.
Exporteren	Dit commando levert een kopie op van een map onder versiebeheer, net zoals een werkkopie, maar dan zonder de lokale <code>.svn</code> mappen.
FSFS	Een Subversion eigen bestandssysteem voor de backend van archieven. Kan gebruikt worden op netwerkschijven. Standaard voor versie 1.2 en nieuwere archieven.
Geschiedenis	Toont de revisiegeschiedenis van een bestand of map. Ook bekend als "Log"
GPO	Groepsolicyobject.
HEAD revisie	De meest recente revisie van een bestand of map in het <i>archief</i> . Ook bekend als HEAD revisie.
Importeren	Het Subversion commando voor het in het archief importeren van een complete mapstructuur in één enkele revisie.
Kopie	In een Subversion archief kun je een kopie aanmaken van een enkel bestand of een hele mapstructuur. Deze kopieën worden gemaakt als "goedkope kopieën" wat eigenlijk niet meer is dan het aanmaken van een link naar het origineel, waardoor ze bijna geen extra ruimte in beslag nemen. Bij het maken van een kopie wordt de geschiedenis ook meegenomen, waardoor je de geschiedenis kunt achterhalen van voordat de kopie werd gemaakt.

Locatie wijzigen	<p>Als je archief verhuist, bijvoorbeeld doordat je het verplaatst hebt naar een andere map op je server of het domein van de server is veranderd, moet je je werkkopie de “locatie wijzigen” van je werkkopie. De URL's voor het archief van je werkkopie moeten gaan wijzen naar de nieuwe locatie.</p> <p>Noot: je moet dit commando alleen gebruiken als het hele archief verhuisd is, terwijl jouw werkkopie nog steeds wijst naar dezelfde locatie in dat archief. In elke andere situatie moet je in plaats hiervan waarschijnlijk het “Wisselen” commando gebruiken.</p>
Log	Toont de revisiegeschiedenis van een bestand of map. Ook bekend als “Geschiedenis”
Ongedaan maken	Subversion bewaart een lokale “ongewijzigde” kopie van elk bestand, zoals het was op het moment dat je je werkkopie voor het laatst ververste. Als je wijzigingen hebt aangebracht en besluit deze terug te draaien, dan kun je met het commando “Ongedaan maken” gebruiken om de ongewijzigde kopie weer terug te krijgen.
Ophalen	Een Subversion commando waarmee een lokale werkkopie wordt gemaakt in een lege (lokale) map door de bestanden met versiebeheer uit het archief op te halen.
Oplossen	Als bestanden in een werkkopie in een conflict situatie zijn achtergebleven na een samenvoegactie, dan moeten die conflicten door een mens opgelost worden door gebruik te maken van een tekstverwerker (of misschien TortoiseMerge). Dit proces wordt “Conflicten Oplossen” genoemd. Als je de conflicten hebt opgelost, dan moet je het conflicterende bestand markeren als Opgelost, waarna de wijzigingen vastgelegd kunnen worden.
Optie	Met “Verversen-naar-revisie” kun je je werkkopie de status/inhoud geven zoals de informatie beschikbaar was op een ander moment in de tijd, zo kun je met “Wisselen” je werkkopie naar een ander deel van het archief laten kijken wat inhoud betreft. Dit is vooral handig als je aan de basislijn en takken werkt, waarbij er maar enkele bestanden verschillend zijn. Je kunt dan je werkkopie wisselen tussen de twee, terwijl alleen de gewijzigde bestanden verstuurd worden.
Patch	Als er in een werkkopie alleen wijzigingen in tekstbestanden zijn doorgevoerd, dan is het mogelijk om met het Diff commando van Subversion een enkel bestand te genereren met de wijzigingen in Unified Diff formaat. Een bestand met deze indeling wordt meestal een “Patch” genoemd, kan naar iemand anders (of een maillijst) gestuurd worden en worden toegepast op een werkkopie. Iemand die geen rechten voor het vastleggen heeft, kan zo toch wijzigingen maken en deze in een patch bestand naar iemand sturen die wel rechten heeft. Je kunt de patch ook gebruiken als je je wijzigingen wilt laten nakijken door iemand anders.
Revisie	<p>Elke keer dat je een set van wijzigingen vastlegt, maak je een nieuwe “revisie” aan in het archief. Elke revisie geeft de status van de structuur van het archief weer op een zeker moment in de tijd. Als je terug wilt gaan in de tijd, kun je het archief bekijken zoals het was ten tijde van revisie N.</p> <p>Op een andere manier gezegd, een revisie kan wijzen naar een set van wijzigingen die zijn doorgevoerd toen de revisie aangemaakt is.</p>
Revisie eigenschap (revprop)	Net zoals bestanden eigenschappen kunnen hebben, kan elke revisie in het archief ook eigenschappen hebben. Enkele speciale revprops worden automatisch toegevoegd als de revisie gemaakt wordt, namelijk: <code>svn:date</code> <code>svn:author</code> <code>svn:log</code> respectievelijk de datum en tijd van de

	<p>vastlegging, degene die de vastlegging uitvoerde en het logboek bericht. Deze eigenschappen kunnen bewerkt worden, maar deze wijzigingen worden niet met versiebeheer gevolgd. Elke wijziging aan zo'n eigenschap is dus permanent en kan niet ongedaan gemaakt worden.</p>
Samenvoegen	<p>Dit is het proces waarmee je wijzigingen vanuit het archief in jouw lokale werkkopie opneemt, zonder dat de wijzigingen die jij hebt toegepast verstoord worden. Soms kunnen de wijzigingen niet helemaal automatisch samengevoegd worden, waardoor je werkkopie in een zogenoemde conflict status terecht komt.</p> <p>Samenvoegen vindt automatisch plaats als je je werkkopie ververs. Je kunt ook specifieke wijzigingen samenvoegen vanuit een andere tak met het TortoisSVN Samenvoeg commando.</p>
Schonen	<p>Om het Subversion boek aan te halen: “Rekursief opschonen van de werkkopie, blokkades verwijderen en het afronden van niet afgeronde acties. Als je ooit een <i>werkkopie geblokkeerd</i> foutmelding hebt gehad, gebruik dan dit commando om achtergebleven blokkades te verwijderen en je werkkopie weer in een bruikbare situatie te brengen.” Merk op dat in deze context <i>geblokkeerd</i> refereert naar een blokkade van het lokale bestandssysteem, niet het plaatsen van blokkades op een archief.</p>
SVN	<p>Een veelgebruikte afkorting voor Subversion</p> <p>De naam van het Subversion eigen protocol wat gebruikt wordt door “svnserve”, de archief server.</p>
Tak	<p>Een term die regelmatig gebruikt wordt bij versiebeheer systemen om aan te geven wat er gebeurt als een ontwikkeling zich op een bepaald moment opsplijst en 2 aparte paden volgt. Je kunt een vertakking maken van de basislijn voor het ontwikkelen van een nieuwe functie, zonder dat de basislijn onstabiel wordt. Of je kunt een stabiele vrijgave vertakken om alleen bug op te lossen, terwijl nieuwe ontwikkelingen plaatsvinden op de onstabiele basislijn (trunk). In Subversion worden takken gemaakt als “goedkope kopieën”.</p>
Toevoegen	<p>Een Subversion commando die gebruikt wordt om bestanden en mappen toe te voegen aan je werkkopie. De nieuwe objecten worden pas aan het archief toegevoegd als je je wijzigingen vastlegt.</p>
Vastleggen	<p>Met dit Subversion commando worden de wijzigingen in je lokale werkkopie naar het archief gestuurd, waardoor een nieuwe revisie in het archief wordt aangemaakt.</p>
Verklaren	<p>Dit commando is alleen voor tekstbestanden bedoeld en analyseert elke regel, zodat het aangegeven kan worden in welke revisie van het archief welke auteur welke regel gewijzigd heeft. Onze GUI implementatie heeft TortoiseBlame en deze toont ook de vastleg datum/tijd en het logboek bericht als je je muis boven het revisienummer houdt.</p>
Verschil	<p>Toont de “Verschillen”, ook bekend als Diff. Erg handig als je wilt zien welke wijzigingen er precies zijn doorgevoerd.</p>
Verversen	<p>Dit Subversion commando haalt de laatste wijzigingen op uit het archief naar je werkkopie, terwijl het wijzigingen van anderen samenvoegt met jouw lokale wijzigingen in je werkkopie.</p>
Verwijderen	<p>Als je een object met versiebeheer verwijderd (en deze wijziging vastlegt) zal het object niet langer aanwezig zijn in het archief vanaf deze vastgelegde revisie. Het object is natuurlijk nog wel beschikbaar in het archief voor de</p>

Werkkopie

oudere revisies, je kunt het dan dus nog steeds terugvinden. Indien nodig, kun je een verwijderd object kopiëren en weer “terughalen” met de complete geschiedenis.

Dit is je lokale “zandbak”, de omgeving waar je aan je bestanden met versiebeheer werkt en welke normaal gesproken op je lokale harde schijf is opgeslagen. Je maakt een werkkopie aan door het “Ophalen” van de bestanden uit een archief. Je wijzigingen sla je ook weer in het archief op door het “Vastleggen” van je gewijzigde bestanden.

---

# Register

## Symbolen

'werkkopie' zonder versiebeheer, 138  
(+), 49

## A

Aanmaken, 18  
    TortoiseSVN, 18  
alleen-lezen, 123  
annoteren, 129  
archief, 7, 28  
archief URL gewijzigd, 140  
archief venster, 146  
archief verkenner, 132  
archief-verkenner, 132  
ASP projecten, 212  
authenticatie, 27  
authenticatie cache, 27  
auto-props, 90  
automatisering, 215, 221, 222, 223

## B

bestanden toevoegen aan archief, 28  
bestanden vergelijken, 208  
bewerk log/auteur, 67  
blokkeren, 123  
bug tracker, 141  
bug tracking, 141  
bugtracker, 141

## C

check in, 33  
CLI, 224  
client hooks, 175  
COM, 192, 200  
COM SubWCRRev interface, 197  
command line, 215, 222, 223  
command line client, 224  
commit bericht, 206  
commit monitor, 189  
commit ongedaan maken, 207  
conflict, 11, 42  
contextsmenu, 24  
context menu ingangen, 212  
controleer op nieuwe versie, 211

## D

diff, 74, 127  
domeincontroller, 211  
Drag and drop, 26, 26

## E

explorer, xi  
exporteer veranderingen, 75

exporteren, 138  
Externe archieven, 105  
externen, 105, 208

## F

filteren, 68  
functies uitschakelen, 212

## G

geluiden, 147  
geschiedenis, 56  
globaal negeren, 149  
globbing, 82  
GPO, 211  
grafiek, 135  
groepsbeleid, 211, 212

## H

hernoem bestanden, 80  
hernoemen, 84, 132, 206  
herorganiseren, 206  
hook scripts, 21, 175  
hooks, 21

## I

IBugtraqProvider, 200  
importeren, 28  
installeer, 1  
instellingen, 147  
invoeren op zijn plaats, 30  
issue tracker, 141, 200

## K

kopieer bestanden, 80  
kopiëren, 108, 132

## L

laag prioriteit, 231  
label, 80, 108  
leeg bericht, 206  
link, 22  
Link voor Ophalen, 22  
log, 56  
log cache, 172  
logbericht, 206  
logboekboodschappen, 56

## M

maak een werkkopie, 30  
mappen vergelijken, 208  
mapstructuur conflicten, 42  
markeer vrijgave, 108  
maximaliseer, 28  
merge tools, 78  
Microsoft Word, 78  
monitoring projects, 189

msi, 211

## N

negeer, 81

Netwerk schijf, 19

Niet geversioneerde bestanden, 81

## O

ongedaan maken, 85

ophalen, 30

oplossen, 42

opschonen, 85, 86

overlays, 46, 231

## P

partial checkout, 30

patch, 127

patroonherkenning, 82

pictogrammen, 46

plugin, 200

prijzen, 129

project eigenschappen, 91

project monitor, 189

proxy server, 164

## R

rechterklik, 24

rechts slepen, 26

register, 183

remote commits, 189

repository aanmaken, 18

reservekopie, 21

revisie, 14, 135

revisie eigenschappen, 67

revisie schema, 135

revprops, 67

## S

samenvoeg conflicten, 119

samenvoegen, 112

    revisiebereik, 113

    twee takken, 116

samenvoegen traceren, 118

scheiden van archief, 209

server gebaseerde hook scripts, 21

server verplaatst, 140

server-side acties, 132

shelve, 54

snelkoppeling, 209

sparse checkout, 30

speciale bestanden, 30

spellingscontrole, 233

statistieken, 69

status, 46, 49

SUBST schijven, 162

Subversion boek, 7

Subversion eigenschappen, 87

SubWCRev, 192

SVN\_ASP\_DOT\_NET\_HACK, 212

## T

taalpakketen, 233

tak, 80, 108

terugdraaiing, 207

terugzetten, 85, 207

tijdelijke bestanden, 28

Toegang, 19

toevoegen, 79

TortoiseIDiff, 77

TortoiseSVN eigenschappen, 91

TortoiseSVN koppeling, 22

trefwoorden, 88

## U

uitbreiden van trefwoorden, 88

uitsluiten patroon, 149

UNC paden, 19

unified diff, 127

unshelve, 54

upgrade controle, 211

URL veranderd, 140

URL verwerker, 221

## V

vastleggen, 33

Veelgestelde vragen, 205

veelvoorkomende projecten, 208

vendor projecten, 208

Venster eigenschappen, 48

veranderingen, 208

vergelijken, 51, 74

vergelijken van revisies, 75

verhuizen, 140

verklaren, 129

verplaats bestanden, 80

verplaatsen, 84, 206

verplaatste server, 140

verschil gereedschappen, 78

verschil tussen beelden, 77

versie, 211

versie extractie, 192

versiebeheer, xi

versiebeheer verwijder, 209

versiebeheer verwijderen, 140, 209

versienummering in bestanden, 192

versioneer nieuwe bestanden, 79

verspreiden, 211

verstuur wijzigingen, 33

vertalingen, 233

verversen, 40, 206

verwijderen, 83

vestleggen berichten, 56

ViewVC, 146

Voeg trackinglog samen, 66

VS2003, 212

## **W**

web weergave, 146

website, 22

WebSVN, 146

weghalen, 83

werkkopie, 12

werkkopie status, 46

werkset, 51

wijziging ongedaan maken, 207

wijzigingen, 49

wijzigingen ophalen, 40

wijzigingen weergeven, 46

Windows shell, xi

wisselen, 111

woordenboek, 233