

TortoiseSVN

**Odjemalec za Subversion v
operacijskem sistemu Windows**

Verzija 1.6.16

**Stefan Küng
Lübbe Onken
Simon Large**

TortoiseSVN: Odjemalec za Subversion v operacijskem sistemu Windows: Verzija 1.6.16

od Stefan K ng, L bbe Onken, in Simon Large
Prevod: Matja   epon (matjaz.cepon@gmail.com)

Izdano 2011/01/21 21:21:17 (r20750)

Kazalo

Predgovor	xi
1. Komu je priročnik namenjen	xi
2. Vodnik po knjigi	xi
3. TortoiseSVN je brezplačen!	xii
4. Skupnost	xii
5. Zasluge	xii
6. Uporabljena terminologija	xii
1. Uvod	1
1.1. Kaj je TortoiseSVN?	1
1.2. Zgodovina TortoiseSVN	1
1.3. Značilnosti TortoiseSVN	1
1.4. Namestitev TortoiseSVN	2
1.4.1. Sistemske zahteve	2
1.4.2. Namestitev	3
1.4.3. Jezikovni paketi	3
1.4.4. Črkovalnik	3
2. Basic Version-Control Concepts	4
2.1. Skladišče	4
2.2. Modeli nadzora različic	4
2.2.1. Težave souporabe datotek	5
2.2.2. Rešitev zakleni-spremeni-odkleni	5
2.2.3. Rešitev kopiraj-spremeni-spoji	6
2.2.4. Kaj naredi Subversion?	9
2.3. Subversion v akciji	9
2.3.1. Delovne kopije	9
2.3.2. Naslovi URL skladišča	11
2.3.3. Revizije	11
2.3.4. Kako delovne kopije spremljajo skladišče	13
2.4. Povzetek	13
3. Skladišče	15
3.1. Ustvarjanje skladišča	15
3.1.1. Ustvarjanje skladišča z odjemalcem za ukazno vrstico	15
3.1.2. Ustvarjanje skladišča s programom TortoiseSVN	15
3.1.3. Krajevni dostop do skladišča	16
3.1.4. Dostop do skladišča na deljenem omrežnem pogonu	16
3.1.5. Postavitev skladišča	17
3.2. Varnostna kopija skladišča	18
3.3. Server side hook scripts	19
3.4. Povezave za prevzem	19
3.5. Accessing the Repository	20
3.6. Strežnik svnservice	20
3.6.1. Uvod	20
3.6.2. Nameščanje svnservice	20
3.6.3. Poganjanje svnservice	21
3.6.4. Osnovna avtentikacija s svnservice	22
3.6.5. Boljša varnost s SASL	23
3.6.6. Avtentikacija s svn+ssh	25
3.6.7. Avtorizacija svnservice na osnovi poti	25
3.7. Strežnik na osnovi strežnika Apache	25
3.7.1. Uvod	25
3.7.2. Namestitev strežnika Apache	26
3.7.3. Namestitev sistema Subversion	26
3.7.4. Konfiguracija	27
3.7.5. Več skladišč	29
3.7.6. Avtorizacija na osnovi poti	29

3.7.7. Avtentikacija z domeno Windows	30
3.7.8. Avtentikacija iz več virov	31
3.7.9. Zavarovanje strežnika z SSL	32
3.7.10. Uporaba certifikatov odjemalca z navideznimi gostitelji SSL	34
4. Dnevna uporaba	36
4.1. Kako začeti	36
4.1.1. Prekrivne ikone	36
4.1.2. Kontekstni meni	36
4.1.3. Povleci in spusti	38
4.1.4. Pogoste bližnjice	39
4.1.5. Avtentikacija	39
4.1.6. Povečevanje oken	40
4.2. Uvažanje podatkov v skladišče	40
4.2.1. Uvoz	40
4.2.2. Uvažanje na mestu	42
4.2.3. Posebne datoteke	42
4.3. Prezemanje delovne kopije	42
4.3.1. Globina prevzema	43
4.4. Objavljanje sprememb v skladišču	45
4.4.1. Okno objave	45
4.4.2. Seznami sprememb	47
4.4.3. Izključevanje elementov iz okna objav	47
4.4.4. Sporočila dnevniških zapisov objav	47
4.4.5. Napredek objave	49
4.5. Posodobite delovno kopijo s spremembami ostalih uporabnikov	50
4.6. Reševanje sporov	52
4.6.1. File Conflicts	52
4.6.2. Tree Conflicts	53
4.7. Pridobivanje informacije o stanju	56
4.7.1. Prekrivne ikone	56
4.7.2. Stolpci TortoiseSVN v Raziskovalcu	57
4.7.3. Krajevno in oddaljeno stanje	58
4.7.4. Pregledovanje razlik	59
4.8. Seznami sprememb	60
4.9. Pogovorno okno Dnevnik	62
4.9.1. Klicanje pogovornega okna dnevniških zapisov	63
4.9.2. Akcije dnevnika	64
4.9.3. Pridobivanje dodatnih informacij	64
4.9.4. Pridobivanje dodatnih dnevniških zapisov	68
4.9.5. Current Working Copy Revision	69
4.9.6. Zmožnosti sledenja spajanja	69
4.9.7. Spreminjanje sporočila dnevniškega zapisa in avtorja	70
4.9.8. Filtriranje dnevniških zapisov	71
4.9.9. Statistične informacije	72
4.9.10. Nepovezan način	75
4.9.11. Osveževanje pogleda	75
4.10. Pregledovanje razlik	75
4.10.1. Spremembe v datoteki	76
4.10.2. Nastavitev zaključkov vrstic in presledkov	77
4.10.3. Primerjanje map	77
4.10.4. Razlikovanje slik z uporabo programa TortoiseIDiff	79
4.10.5. Zunanja orodja za razlikovanje/spajanje	80
4.11. Dodajanje novih datotek in map	81
4.12. Kopiranje/premikanje/preimenovanje obstoječih datotek in map	81
4.13. Dodajanje datotek in map na seznam prezrtih elementov	83
4.13.1. Iskanje vzorcev v seznamu prezrtih elementov	84
4.14. Brisanje, preimenovanje in premikanje	84
4.14.1. Brisanje datotek in map	85

4.14.2. Premikanje datotek in map	86
4.14.3. Spreminjanje velikih in malih črk v imenih datotek	86
4.14.4. Obravnava težav pri imenih datotek zaradi velikih in malih črk	87
4.14.5. Popravljanje preimenovanj datotek	87
4.14.6. Brisanje datotek brez različic	87
4.15. Razveljavljanje sprememb	88
4.16. Čiščenje	89
4.17. Nastavitve projekta	89
4.17.1. Lastnosti Subversion	90
4.17.2. Projektne lastnosti TortoiseSVN	93
4.18. External Items	95
4.18.1. External Folders	95
4.18.2. External Files	98
4.19. Ustvarjanje vej/oznak	98
4.19.1. Ustvarjanje veje ali oznake	98
4.19.2. Prevzeti ali preklopiti...	100
4.20. Spajanje	101
4.20.1. Spajanje območja revizij	102
4.20.2. Vključitev veje	104
4.20.3. Spajanje dveh različnih dreves	105
4.20.4. Možnosti spajanja	106
4.20.5. Preverjanje rezultatov spajanja	107
4.20.6. Sledenje spajanja	108
4.20.7. Reševanje sporom med spajanjem	108
4.20.8. Spajanje zaključene veje	109
4.20.9. Vzdrževanje stranskih vej	109
4.21. Zaklepanje	110
4.21.1. Zaklepanje v sistemu Subversion	110
4.21.2. Pridobivanje zaklepa	111
4.21.3. Sprostitev zaklepa	111
4.21.4. Preverjanje stanja zaklepanja	112
4.21.5. Nastavite nezaklenjene datoteke samo za branje	112
4.21.6. Ukazna datoteka akcije za zaklepanje	113
4.22. Ustvarjanje in nameščanje popravkov	113
4.22.1. Ustvarjanje datoteke popravkov	113
4.22.2. Nameščanje datoteke popravkov	114
4.23. Kdo je spremenil posamezno vrstico?	115
4.23.1. Okrivi datoteke	115
4.23.2. Okrivi spremembe	117
4.24. Brskalnik po skladišču	117
4.25. Grafi revizij	119
4.25.1. Vozlišča grafa revizij	121
4.25.2. Spreminjanje pogleda	121
4.25.3. Uporaba grafa revizij	123
4.25.4. Osveževanje pogleda	124
4.25.5. Pruning Trees	124
4.26. Izvažanje delovne kopije sistema Subversion	124
4.26.1. Kako odstranim delovno kopijo iz nadzora različic	126
4.27. Premeščanje delovne kopije	126
4.28. Integracija s sistemi za sledenje zadev	127
4.28.1. Dodajanje številke zadev dnevniškimi zapisom	127
4.28.2. Pridobivanje informacij iz sledilnika zadev	130
4.29. Integracija z internetno naravnanimi pregledovalniki skladišč	131
4.30. Nastavitve TortoiseSVN	132
4.30.1. Splošne nastavitve	132
4.30.2. Revision Graph Settings	140
4.30.3. Nastavitve prekrivnih ikon	142
4.30.4. Nastavitve omrežja	145

4.30.5. Nastavitve zunanjih programov	147
4.30.6. Shranjeni podatki	151
4.30.7. Predpomnenje dnevnika	152
4.30.8. Ukazne datoteke akcij na strani odjemalca	155
4.30.9. Nastavitve TortoiseBlame	159
4.30.10. Nastavitve registra	159
4.30.11. Delovne mape Subversion	161
4.31. Zadnji korak	161
5. Program SubWCRev	162
5.1. Program SubWCRev za ukazno vrtico	162
5.2. Zamenjava ključnih besed	162
5.3. Primer uprabe ključne besede	163
5.4. Vmesnik COM	164
6. IBUGtraqProvider interface	167
6.1. The IBUGtraqProvider interface	167
6.2. The IBUGtraqProvider2 interface	168
A. Pogosto zastavljena vprašanja (FAQ)	171
B. Kako naredim...	172
B.1. Kako premaknem/kopiram večje število datotek naenkrat	172
B.2. Kako prisilim uporabnika, da vnese sporočilo dnevniškega zapisa	172
B.2.1. Ukazna datoteka akcije na strežniku	172
B.2.2. Lasnosti projekta	173
B.3. Kako posodobim izbrane datoteke iz skladišča	173
B.4. Kako prevrtim nazaj revizije v skladišču	173
B.4.1. Uporabite okno za prikaz dnevniških zapisov	173
B.4.2. Uporabite okno za spajanje	173
B.4.3. Uporabite <code>svndumpfilter</code>	174
B.5. Compare two revisions of a file or folder	174
B.6. Kako vključim skupni podprojekt	174
B.6.1. Uporabite lastnost <code>svn:externals</code>	174
B.6.2. Uporabite vgnezdjeno delovno kopijo	175
B.6.3. Uporabite relativno lokacijo	175
B.7. Kako ustvarim bližnjico do skladišča	175
B.8. Kako dodam na seznam prezrtih datoteke, ki so že pod nadzorom	176
B.9. Odstranjevanje delovne kopije iz nadzora različic	176
B.10. Kako odstranim delovno kopijo	176
C. Uporabni namigi za skrbnike sistema	177
C.1. Namestitev TortoiseSVN preko pravic skupin	177
C.2. Preusmerjanje iskanja najnovejše različice	177
C.3. Nastavljanje okoljske spremenljivke <code>SVN_ASP_DOT_NET_HACK</code>	178
C.4. Onemogočanje kontekstnega menija	178
D. Avtomatizacija TortoiseSVN	180
D.1. Ukazi TortoiseSVN	180
D.2. Ukazi TortoiseIDiff	183
E. Ustrezni ukazi v odjemalcu za ukazno vrstico	184
E.1. Konvencije in osnovna pravila	184
E.2. Ukazi TortoiseSVN	184
E.2.1. Prezemi	184
E.2.2. Posodobi	184
E.2.3. Posodobi na revizijo	185
E.2.4. Objavi	185
E.2.5. Razlikuj	185
E.2.6. Pokaži dnevnik	186
E.2.7. Preveri posodobitve	186
E.2.8. Graf revizij	186
E.2.9. Brskalnik po skladišču	186
E.2.10. Uredi spore	186
E.2.11. Rešeno	186

E.2.12. Preimenuj	187
E.2.13. Izbriši	187
E.2.14. Povrni	187
E.2.15. Čiščenje	187
E.2.16. Dobi zaklep	187
E.2.17. Odstrani zaklep	187
E.2.18. Veja/Oznaka	187
E.2.19. Preklop	188
E.2.20. Spoji	188
E.2.21. Izvozi	188
E.2.22. Premakni	188
E.2.23. Tu ustvari skladišče	188
E.2.24. Dodaj	188
E.2.25. Uvoz	189
E.2.26. Okrivi	189
E.2.27. Dodaj na seznam prezrtih	189
E.2.28. Ustvari popravek	189
E.2.29. Namesti popravek	189
F. Podrobnosti o izvedbi	190
F.1. Prekrivne ikone	190
G. Zavarovanje strežnika Svserve z SSL	192
G.1. Nameščanje strežnika na operacijskem sistemu Linux	192
G.2. Nameščanje strežnika na operacijskem sistemu Windows	192
G.3. Odjemalci SSH za uporabo s TortoiseSVN	193
G.4. Ustvarjanje certifikatov OpenSSH	193
G.4.1. Ustvarjanje ključev s ssh-keygen	193
G.4.2. Ustvarjanje ključev s programom PuTTYgen	193
G.5. Preizkušanje s programom PuTTY	193
G.6. Preizkušanje SSH s TortoiseSVN	194
G.7. Variacije konfiguracije SSH	195
Slovar	196
Stvarno kazalo	199

Seznam slik

2.1. Tipičen sistem odjemalec/strežnik	4
2.2. Težava, ki se ji je potrebno izogniti	5
2.3. Rešitev zakleni-spremeni-odkleni	6
2.4. Rešitev kopiraj-spremeni-spoji	7
2.5. ...Kopiraj-spremeni-spoji (nadaljevanje)	8
2.6. Datotečni sistem skladišča	10
2.7. Skladišče	12
3.1. Menu TortoiseSVN za datoteke brez različic	15
4.1. Raziskovalec prikaže prekrivne ikone	36
4.2. Kontekstni meni za mapo pod nadzorom različic	37
4.3. Kontekstni meni v Raziskovalcu za bližnjico v mapi pod nadzorom različic	38
4.4. Meni ob premikanju mape, ki je pod nadzorom različic	39
4.5. Okno za avtentikacijo	40
4.6. Okno za uvažanje	41
4.7. Okno za prevzem	43
4.8. Okno objave	45
4.9. Črkovalnik v oknu objave	48
4.10. Okno napredka prikazuje napredovanje objave	49
4.11. Okno napredka prikazuje končano posodobitev	50
4.12. Raziskovalec prikaže prekrivne ikone	56
4.13. Preveri posodobitve	58
4.14. Okno za objave s seznamami sprememb	61
4.15. Okno dnevnika	63
4.16. Zgornji del Dnevnika s kontekstnim menijem	64
4.17. Kontekstni meni v zgornjem delu okna v primeru dveh izbranih revizij	66
4.18. Kontekstni meni spodnjega dela Dnevnika	67
4.19. Dnevnik prikazuje sledenje spajanja revizij	70
4.20. Histogram objav glede na avtorja	72
4.21. Potični graf objav glede na avtoja	73
4.22. Objave po datumu	74
4.23. Go Offline Dialog	75
4.24. Okno za primerjanje revizij	78
4.25. Pregledovalnik razlik med slikami	79
4.26. Kontekstni menu v Raziskovalcu za datoteke brez različic	81
4.27. Meni ob premikanju mape, ki je pod nadzorom različic	82
4.28. Kontekstni menu v Raziskovalcu za datoteke brez različic	83
4.29. Kontekstni menu v Raziskovalcu za datotek pod nadzorom različic	85
4.30. Okno za povrnitev	88
4.31. Stran Lastnosti v Raziskovalcu, zavihek Subversion	90
4.32. Lastnosti v sistemu Subversion	91
4.33. Dodajanje lastnosti	92
4.34. Okno za ustvarjanje veje/oznake	99
4.35. Okno za preklon	101
4.36. Čarovnik za spajanje - Izberite obseg revizije	103
4.37. Čarovnik za spajanje - Vključitev veje	105
4.38. Čarovnik za spajanje - Spajanje dreves	106
4.39. Povratne informacije o sporih spajanja	108
4.40. Okno Vključitev veje	109
4.41. Okno zaklepov	111
4.42. Pogovorno okno Preveri posodobitve	112
4.43. Okno za ustvarjanje popravkov	114
4.44. Okno hvale/okrivi	115
4.45. TortoiseBlame	116
4.46. Brskalnik po skladišču	118
4.47. Graf revizije	120

4.48. Okno Uvoz-iz-URL	125
4.49. Okno za premeščanje	126
4.50. Primer poizvedovalnika sledilnika zadev	130
4.51. Okno za nastavitve, Splošno	132
4.52. Okno nastavitvev, Kontekstni meni	134
4.53. Okno nastavitvev, Pogovorna okna 1	135
4.54. Okno nastavitvev, Pogovorna okna 2	137
4.55. Okno nastavitvev, Barve	139
4.56. The Settings Dialog, Revision Graph Page	140
4.57. The Settings Dialog, Revision Graph Colors Page	141
4.58. Okno nastavitvev, Izbor ikon	142
4.59. Okno nastavitvev, Izbor ikon	145
4.60. Okno nastavitvev, Omrežje	146
4.61. Okno nastavitvev, ogledovalnik razlik	147
4.62. Okno nastavitvev, napredne nastavitve razlikovanja/spajanja	150
4.63. Okno nastavitvev, Shranjeni podatki	151
4.64. Okno nastavitvev, Predpomnilnik dnevnika	152
4.65. Okno nastavitvev, Statistika predpomnilnika dnevnika	154
4.66. Okno nastavitvev, Ukazne datoteke akcij	155
4.67. Okno nastavitvev, nastavitvev ukaznih datotek akcij	156
4.68. Okno za nastavitve, Okno za integracijo sledilnika zadev	158
4.69. Okno za nastavitve, TortoiseBlame	159
C.1. Okno za nadgradnjo	177

Seznam tabel

2.1. Naslovi URL za dostop do skladišča	11
3.1. Nastavitve Apache <code>httpd.conf</code>	28
5.1. Seznam stikal ukazne vrstice, ki so na voljo	162
5.2. Seznam stikal ukazne vrstice, ki so na voljo	163
5.3. Podprte COM/avtomatizacijske metode	164
C.1. Elementi menija in njihove vrednosti	178
D.1. Seznam ukazov in možnosti	180
D.2. Seznam možnosti	183

Predgovor



TortoiseSVN

- Delate v teamu?
- Se vam je že kdaj zgodilo, da ste popravljali datoteko istočasno kot nekdo drug? Ste zaradi tega izgubili narejene spremembe?
- Ste že kdaj shranili datoteko in potem želeli spremembe preklicati? Ste si kdaj želeli, da bi lahko videli, kako je datoteka izgledala pred časom?
- Ste kdaj našli hrošča v svojem projektu in ste želeli izvedeti, kdaj se je prikradel tja?

Če ste vsaj na eno vprašanje odgovorili z “da”, potem je TortoiseSVN za vas! Preberite si, kako vam lahko TortoiseSVN pomaga pri delu. Sploh ni težko.

1. Komu je priročnik namenjen

Kniga je napisana za računalniško pismene uporabnike, ki želijo uporabljati Subversion, vendar niso večši dela preko ukazne vrstice. Ker je TortoiseSVN razširitev Windowsove lupine, predpostavljamo, da uporabnik pozna Raziskovalca in ga zna uporabljati.

2. Vodnik po knjigi

V poglavju **Predgovor** podamo nekaj podatkov o projektu TortoiseSVN, o skupnosti ljudi, ki dela na tem projektu ter o licenčnih pogojih uporabe in distribucije.

V poglavju **Poglavje 1, Uvod** si pogledamo, kaj TortoiseSVN je, kaj lahko z njim naredimo, od kje prihaja in osnovna navodila za namestitev na vašem računalniku.

Poglavje **Poglavje 2, Basic Version-Control Concepts** poda nekaj osnov o sistemu za nadzor različic *Subversion*, ki je podlaga za TortoiseSVN. Ta del je izposojen iz dokumentacije Subversion in ponazori različne pristope k vodenju različic datotek in delovanje sistema Subversion.

The chapter on **Poglavje 3, Skladišče** 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. There is also a section here on how to setup a server if you need one.

Poglavje 4, Dnevna uporaba je najpomembnejše poglavje, saj so v njem razložene vse glavne zmožnosti programa TortoiseSVN in njihova uporaba. Napisano je kot vodnik. Začne se z prevzemom delovne kopije, njenim spreminjanjem, nadaljuje pa z objavo narejenih sprememb. V nadaljevanju so razložene zahtevnejše tematike.

Poglavje 5, Program SubWCRev je zunanji program, ki je vključen v paket TortoiseSVN. Z njegovo pomočjo lahko izluščimo informacije iz delovne kopije in jih zapišemo v datoteko. To je uporabno, če želimo v projekt vključiti informacijo o številki revizije.

Poglavje **Dodatek B, Kako naredim...** poda odgovore na nekatera pogosta vprašanja o nalogah, ki niso opisana v ostalih poglavjih.

Poglavje **Dodatek D, Avtomatizacija TortoiseSVN** ponazori, kako lahko kličemo pogovorna okna TortoiseSVN iz ukazne vrstice. To je primerno za pisanje ukaznih datotek, v katerih še vedno želimo uporabnikove odzive.

Poglavje **Dodatek E, Ustrezni ukazi v odjemalcu za ukazno vrstico** poda seznam ukazov TortoiseSVN in njihovih sorodnih ukazov v odjemalcu Subversion za ukazno vrstico (`svn.exe`).

3. TortoiseSVN je brezplačen!

Program TortoiseSVN je brezplačen. Za njegovo uporabo vam ni potrebno plačati. Uporabljate ga lahko, kakor želite. Razvit je pod licenco GNU General Public License (GPL).

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

4. Skupnost

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

5. Zasluge

Tim Kemp

za ustanovitev projekta TortoiseSVN

Stefan Küng

za trdo delo, s katerim je pripomogel, da je TortoiseSVN danes to, kar je

Lübbe Onken

za čudovite ikone, logotip, iskanje napak, prevode in urejanje prevodov

Simon Large

za pomoč pri dokumentaciji in iskanju napak

The Subversion Book

za izvrsten uvod v sistem Subversion in poglavje številka dve, ki smo ga uporabili v tej knjigi

The Tigris Style project

za nekatere stile, ki smo jih uporabili v tej dokumentaciji

Zunanji sodelavci

za popravke, poročila o napakah in nove ideje in za pomoč ostalim, podano preko naših dopisnih seznamov

Naši donatorji

za številne ure užitkov ob poslušanju glasbe, ki so nam jo podarili

6. Uporabljena terminologija

Da bi olajšali branje knjige, so imena vseh oken in menijev TortoiseSVN označena z drugo barvo, npr. Dnevnik.

Izbira v meniju je prikazana s puščico. TortoiseSVN → Pokaži dnevnik pomeni: iz kontekstnega menija *TortoiseSVN* izberite *Pokaži dnevnik*.

V primeru, ko se kontekstni meni pojavi znotraj enega od pogovornih oken sistema TortoiseSVN, je to prikazano takole: **Kontekstni meni** → **Shrani kot ...**

Gumbi uporabniškega vmesnika so prikazani takole: Za nadaljevanje pritisnite **V redu**.

Uporabnikove akcije so nakazane z uporabo polkrepke pisave. **ALT+A**: pritisnite in držite tipko **Alt** na tipkovnici, hkrati pa pritisnite še tipko **A**. Desni poteg: pritisnite desno tipko na miški, jo držite in *povlecite* elemente na novo lokacijo.

Odgovor sistema in vnos preko tipkovnice sta prav tako ponazorjena z drugačno pisavo.



Pomembno

Pomembne opombe so označene z ikono.



Namig

Namigi za lažje delo.



Opozorilo

Kjer morate biti previdni, kaj počnete.



Pozor

Kjer morate biti še posebej previdni, saj lahko ob neupoštevanju pride do okvare podatkov ali ostalih hudih zapletov.



Poglavje 1. Uvod

Nadzor različic je umetnost vodenja sprememb informacij. Že dolgo je orodje izjemnega pomena za programerje, ki tipično porabijo veliko časa za male spremembe na programski opremi, naslednji dan pa nekatere izmed njih razveljavijo ali preverijo. Zamislite si ekipo razvijalcev programske opreme, ki dela sočasno na istih datotekah in videli boste, zakaj je potreben dober sistem za *vodenje potencialnega nereda*.

1.1. Kaj je TortoiseSVN?

TortoiseSVN je brezplačen odprtokodni odjemalec za sistem nadzora različic *Subversion*. TortoiseSVN upravlja datoteke in mape skozi čas. Datoteke so shranjene v centralnem *skladišču*. Skladišče je podobno običajnemu strežniku datotek, s to razliko, da si zapomni vse spremembe, ki so bile kadarkoli narejene na datotekah in mapah. To omogoča, da pridobite starejšo različico datoteke in pogledate skozi zgodovino, kako so se podatki spreminjali in kdo je spremembe naredil. Prav zato marsikdo smatra *Subversion* in tudi ostale sisteme za nadzor različic kot neke vrste "časovni stroj".

Nekateri sistemi za nadzor različic so posebej prirojeni za urejanje strukture izvorne kode in imajo veliko zmožnosti, ki so specifične za razvoj programske opreme - na primer razumevanje programskih jezikov ali dodatna orodja za njihovo gradnjo. To so sistemi SCM (software configuration management). *Subversion* ni tak sistem; je splošen sistem, ki omogoča urejanje *katerokoli* zbirke datotek, vključno z datotekami izvorne kode.

1.2. Zgodovina TortoiseSVN

Leta 2002 je Tim Kemp ugotovil, da je *Subversion* zelo dober sistem za nadzor različic, a brez grafičnega uporabniškega vmesnika. Idejo za odjemalca kot lupino sistema Windows je dobil iz podobnega odjemalca za sistem CVS - *TortoiseCVS*.

Tim je preučil izvorno kodo aplikacije *TortoiseCVS* in jo uporabil kot osnovo za *TortoiseSVN*. Nato je začel s projektom, registriral domeno `tortoisesvn.org` in objavil izvorno kodo. Med tem časom je Stefan K ng iskal dober brezpla en sistem za nadzor razli ic in na el *Subversion* ter izvorno kodo za *TortoiseSVN*. Ker aplikacija  e ni bila primerna za uporabo, se je pridru il projektu in za el programirati. Kmalu je na novo napisal ve ji del obstoje e kode in za el dodajati nove ukaze in zmo nosti. Prvotna koda je kmalu povsem izginila.

Sistem *Subversion* je postajal vse bolj stabilen in je pridobival vedno ve  uporabnikov, ki so uporabljali *TortoiseSVN* kot odjemalca za *Subversion*.  tevilo uporabnikov je hitro nara alo (in raste  e danes). Takrat je L bbe Onken ponudil svojo pomo  in izdelal lepe ikone in logotip za *TortoiseSVN*. Ureja tudi spletno stran in prevode.

1.3. Zna ilnosti TortoiseSVN

Zakaj je *TortoiseSVN* tako dober odjemalec za *Subversion*? Tukaj je kratek seznam njegovih zna ilnosti.

Integracija z lupino

TortoiseSVN se integrira v lupino operacijskega sistema Windows (v Raziskovalca). To pomeni, da lahko  e naprej delate z orodji, ki jih poznate. In ni vam potrebno zamenjati aplikacije vsaki , ko  elite uporabiti funkcije sistema za nadzor razli ic!

Pri tem pa vam ni nujno uporabljati ravno Raziskovalca. Kontekstni meniji *TortoiseSVN* delujejo v velikem  tevilu upravljalnikov datotek, prav tako pa tudi v pogovornem oknu Datoteka/Odpri, ki se nahaja v vseh standardnih programih sistema Windows. Morate pa se ves as zavedati, da je *TortoiseSVN* namenoma izdelan kot nadgradnja Raziskovalca. Zato je mo no, da v drugih aplikacijah ne deluje popolnoma pravilno. Primer so te ave pri prikazu prekrivnih ikon.

Prekrivne ikone

Stanje vsake datoteke pod nadzorom različic nakazuje majhna prekrivna ikona. Tako lahko hitro vidite, kakšno je stanje vaše delovne kopije.

Preprost dostop do ukazov sistema Subversion

Vsi ukazi sistema Subversion so na razpolago v kontekstnem meniju Raziskovalca. TortoiseSVN doda tja svoj lasten pomeni.

Ker je TortoiseSVN odjemalec za Subversion, vam bomo prikazali tudi nekaj značilnosti samega sistema Subversion:

Vodenje različic map

CVS upravlja le zgodovino posameznih datotek, Subversion pa ima "virtualni" datotečni sistem pod nadzorom različic, ki upravlja spremembe na celotnem drevesu map. Pod nadzorom so datoteke *in* mape. Posledica tega je, da imamo na strani odjemalca na razpolago ukaza **premakni** in **kopiraj**, ki delujeta na datotekah in mapah.

Atomične objave

Objava zapiše vse spremembe v skladišče ali pa jih sploh ne zapiše. To omogoča razvijalcem, da sestavijo in objavijo spremembe kot logične celote.

Metapodatki pod nadzorom različic

Vsaka datoteka in mapa ima prirejeno nevidno množico "lastnosti". Lahko si izmislite kakršen koli par ključ/vrednost. Lastnosti so pod nadzorom različic, prav tako kot vsebina datoteke.

Izbira plasti omrežja

Subversion je vpeljal abstrakten koncept dostopa do skladišča, kar uporabnikom omogoča, da izdelajo nove omrežne mehanizme. Subversionov "napredni" mrežni strežnik je modul za spletni strežnik Apache, ki govori narečje protokola HTTP, imenovano WebDAV/DeltaV. To daje sistemu Subversion veliko prednosti s stališča stabilnosti in povezovanja, prinaša pa še številne dodatne zmožnosti, n. pr.: avtentikacijo, avtorizacijo, *wire compression* in brskanje po skladišču. Na razpolago pa je tudi manjši, samostojen strežnik za Subversion. Strežnik se pogovarja po prilagojenem protokolu, ki se ga preprosto preusmeri preko ssh.

Konsistentno upravljanje s podatki

Subversion zapisuje razlike med datotekami z dvojiškim algoritmom za razlikovanje, ki deluje tako na tekstovnih (uporabniku berljivih) kot na dvojiških (uporabniku neberljivih) datotekah. Datoteke obeh tipov so enako stisnjene in shranjene v skladišču, razlike pa se prenašajo v obeh smereh po mreži.

Učinkovita uporaba vej in oznak

Cena vej in oznak ni nujno sorazmerna z velikostjo projekta. Subversion ustvari veje in oznake tako, da projekt skopira z uporabo mehanizma, podobnega simbolnim povezavam. Tako ti operaciji trajata zelo kratek (konstanten) čas in zavzmeta zelo malo prostora v skladišču.

Možnost vključevanja v druge aplikacije

Sistem Subversion nima zgodovine. Razvit je kot zbirka knjižnic v jeziku C z dobro definiranim programskim vmesnikom (API). Zato je Subversion zelo primeren za vzdrževanje in uporabo s strani ostalih aplikacij in jezikov.

1.4. Namestitev TortoiseSVN

1.4.1. Sistemske zahteve

TortoiseSVN deluje na sistemih Windows 2000 SP2, Windows XP ali novejšem. Od verzije TortoiseSVN 1.2.0 ne podpiramo sistemov Windows 98, Windows ME in Windows NT4. Če želite uporabljati TortoiseSVN na teh sistemih, lahko na naši domači strani najdete starejše različice.

Če med namestitvijo TortoiseSVN ali po njej naletite na težave, se najprej obrnite na [Dodatek A, Pogosto zastavljena vprašanja \(FAQ\)](#).

1.4.2. Namestitev

TortoiseSVN ima enostaven namestitveni program. Dvokliknite na namestitveno datoteko in sledite navodilom. Namestitveni program bo poskrbel za vse potrebno.



Pomembno

Za namstitev TortoiseSVN potrebujete skrbniške pravice.

1.4.3. Jezikovni paketi

Uporabniški vmesnik TortoiseSVN je preveden v veliko jezikov, med drugim tudi v slovenščino. Jezikovne pakete najdete *na tem naslovu* [http://tortoisesvn.net/translation_status]. Če jezikovnega paketa za vaš jezik ni, se lahko pridružite naši ekipi in naredite lasten prevod ;-)

Vsak jezikovni paket je na razpolago v obliki namestitvenega programa `.exe`. Poženite ga in sledite navodilom. Ob naslednjem zagonu programa bo TortoiseSVN uporabljal izbrani jezik.

1.4.4. Črkovalnik

TortoiseSVN vsebuje črkovalnik, ki vam omogoča, da preverite sporočila v dnevniškem zapisu. To je še posebej pomembno, če na projektu uporabljate jezik, ki ni vaš materin jezik. Črkovalnik uporablja iste datoteke s slovarjem kot *OpenOffice* [<http://openoffice.org>] in *Mozilla* [<http://mozilla.org>].

Namestitveni program samodejno doda slovarja za ameriško in britansko angleščino. Če želite črkovalnik za kak dodaten jezik, je najlažja možnost, da namestite jezikovni paket programa TortoiseSVN za ta jezik. Le-ta bo namestil ustrezen slovar, poleg tega pa tudi uporabniški vmesnik v izbranem jeziku. Ko boste ponovno zagnali sistem, vam bo na voljo tudi ta slovar.

Slovarje lahko namestite tudi sami. Če imate nameščene aplikacije OpenOffice ali Mozilla, lahko prekopirate njihove slovarje, ki se nahajajo v namestitvenih mapah teh aplikacij. V nasprotnem primeru morate prenesti zahtevane slovarje z naslova <http://wiki.services.openoffice.org/wiki/Dictionaries>

Ko imate datoteke slovarjev, jih morate verjetno preimenovati, tako da vsebujejo le nize, ki označujejo jezik. Primer:

- en_US.aff
- en_US.dic

Potem jih prekopirajte v podmapo `bin` namestitvene mape TortoiseSVN. Običajno je ta mapa `C:\Program Files\TortoiseSVN\bin`. Če ne želite smetiti po podmapi `bin`, lahko datoteke črkovalnika namesitve v mapo `C:\Program Files\TortoiseSVN\Languages`. Če ta mapa še ne obstaja, jo morate naprej ustvariti. Ko boste TortoiseSVN naslednjič zagnali, bo črkovalnik na voljo.

Če namestite več slovarjev, TortoiseSVN uporabi naslednja pravila, da ugotovi, katerega naj uporabi.

1. Preveri nastavitve lastnosti `tsvn:projectlanguage`. Preberite **Razdelek 4.17**, "Nastavitve projekta" za več informacij o nastavitvi projektnih lastnosti.
2. Če projektni jezik ni nastavljen ali nastavljeni jezik ni nameščen, poskusi z jezikom sistema Windows.
3. Če jezik sistema Windows ne deluje, poizkusi z "osnovnim" jezikom. Primer: `de_CH` (švicarska nemščina) ima za osnovo `de_DE` (nemščino).
4. Če po zgornjih pravilih ne uspe najti slovarja, potem je privzeti jezik angleščina, ki se nahaja v standarnem namestitvenem paketu.

Poglavje 2. Basic Version-Control Concepts

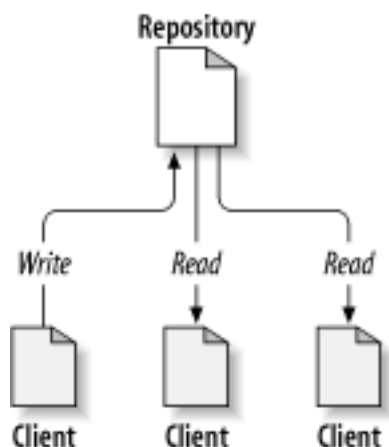
To poglavje je rahlo spremenjena verzija istega poglavja v knjigi The Subversion book. Spletna različica knjige je na voljo na tem naslovu: <http://svnbook.red-bean.com/>.

To poglavje je kratek uvod v Subversion. Če se z nadzorom različic še niste srečali, potem je to poglavje za vas. Začnemo s splošnimi principi nadzora različic, pregledamo posebnosti sistema Subversion in na enostavnih primerih pogledamo, kako se Subversion uporablja.

Čeprav primeri v tem poglavju prikazujejo, kako si uporabniki delijo programsko izvorno kodo, upoštevajte, da lahko Subversion upravlja katerekoli datoteke - ni omejen samo na pomoč računalniškim programerjem.

2.1. Skladišče

Subversion je centraliziran sistem za souporabo informacij. Njegovo bistvo je *skladišče*, centralno mesto za shranjevanje podatkov. Skladišče shranjuje informacije v obliki *drevesne strukture datotečnega sistema*, kar je tipična hierarhija datotek in map. S skladiščem se lahko poveže poljubno število *odjemalcev* in bere ali zapisuje datoteke. S pisanjem podatkov da odjemalec te podatke na voljo drugim; z branjem podatkov odjemalec dobi informacije od drugih.



Slika 2.1. Tipičen sistem odjemalec/strežnik

Zakaj je to zanimivo? Po dosedanji razlagi izgleda to kot tipičen datotečni strežnik. In v resnici skladišče je neke vrste datotečni strežnik. Subversion je poseben zaradi dejstva, da si *zapomni vsako spremembo*, ki je bila kadarkoli zapisana: vsako spremembo vsake datoteke, pa tudi spremembe strukture map, n. pr. dodajanje, brisanje in preureditev datotek in map.

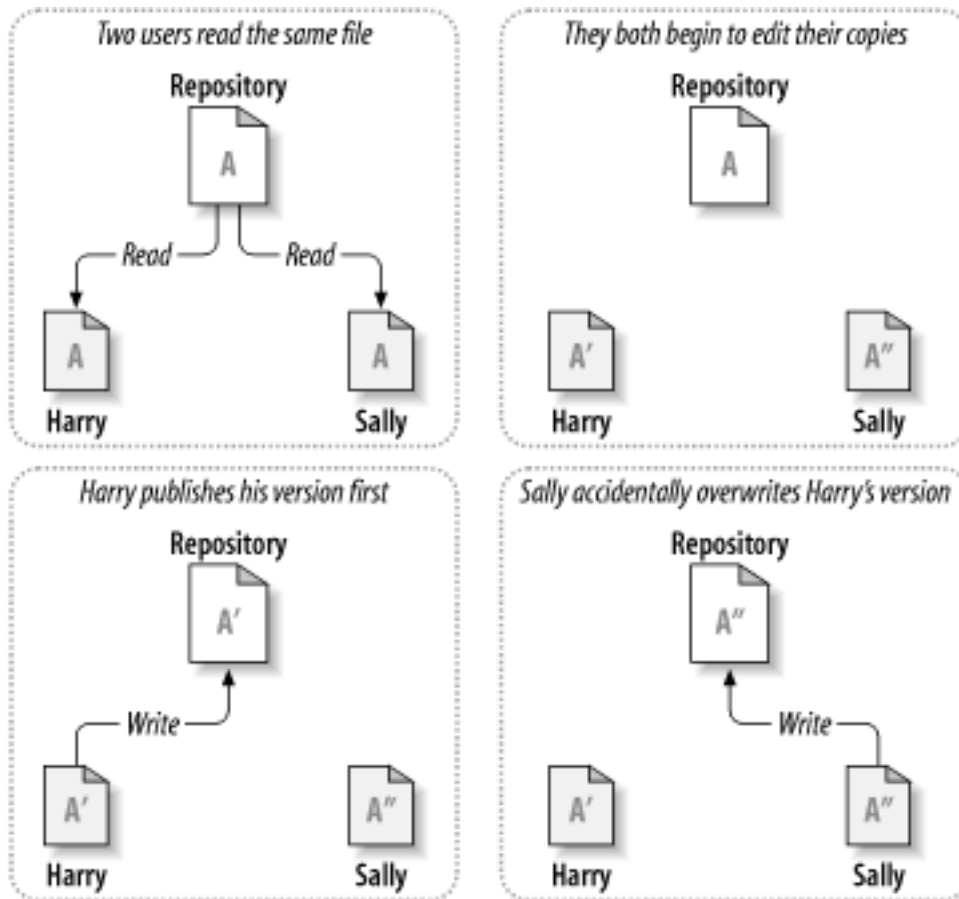
Ko odjemalec bere podatke iz skladišča, običajno gleda le zadnjo različico drevesne strukture map. Vendar ima odjemalec možnost videti tudi *predhodnja* stanja datotečnega sistema. Odjemalec lahko postavi vprašanja o zgodovini, n. pr. "kaj je ta mapa vsebovala prejšnjo sredo?" ali "kdo je zadnji avtor sprememb na tej datoteki in kakšne so te spremembe?" Takšna vprašanja so osrčje kateregakoli *sistema za nadzor različic*. Ti sistemi so nareni za hranjenje in spremljanje sprememb podatkov skozi čas.

2.2. Modeli nadzora različic

Vsi sistemi nadzora različic morajo rešiti isto osnovno težavo: kako bo sistem omogočal souporabo informacij, hkrati pa preprečil, da bi si uporabniki skakali v zelje. Vse prelahko je povoziti spremembe ostalih uporabnikov v skladišču.

2.2.1. Težave souporabe datotek

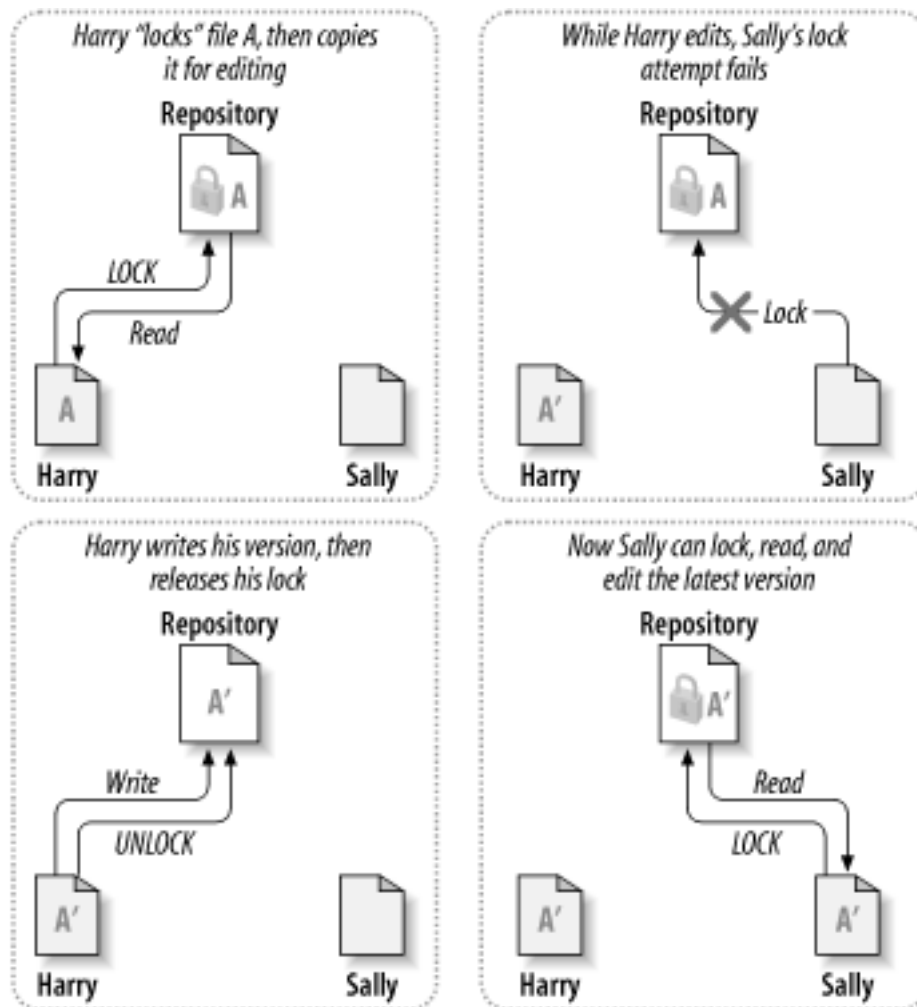
Zamislite si naslednji scenarij: imamo dva sodelavca, Harryja in Sally. Odločita se, da bosta spremenjala isto datoteko iz skladišča. Če spremembe najprej shrani Harry, potem je možno, da Sally (čez nekaj trenutkov) pomotoma prepíše te spremembe s svojo novo različico datoteke. Harryjeva datoteka sicer ne bo izgubljena, saj si sistem zapomni vsako spremembo, *ne bodo* pa te spremembe prisotne v Sallyjini datoteki, saj sploh ne ve zanje. Harryjevo delo je tako kljub vsemu izgubljeno - oziroma vsaj ni na voljo v zadnji različici - in to po vsej verjetnosti po pomoti. Takšnim situacijam se želimo izogniti.



Slika 2.2. Težava, ki se ji je potrebno izogniti

2.2.2. Rešitev zakleni-spremeni-odkleni

Veliko sistemov za nadzor različic uporablja model *zakleni-spremeni-odkleni*, kar predstavlja zelo enostavno rešitev problema. V takšnem sistemu lahko datoteko spreminja samo en uporabnik naenkrat. Pred spreminjanjem datoteke jo mora Harry *zakleniti*. Zaklepanje datoteke je podobno izposoji knjige v knjižnici; če Harry zaklene datoteko, potem Sally ne more narediti nobenih sprememb na tej datoteki. Če poskuša datoteko zakleniti, ji bo skladišče to onemogočilo. Datoteko lahko le bere in čaka, da jo bo Harry nehal urejati in sprostil zaklep. Potem ko Harry odklene datoteko, jo lahko zaklene Sally in jo začne urejati.



Slika 2.3. Rešitev zakleni-spremeni-odkleni

Model zakleni-spremeni-odkleni uporabnike omejuje in jih ovira pri delu:

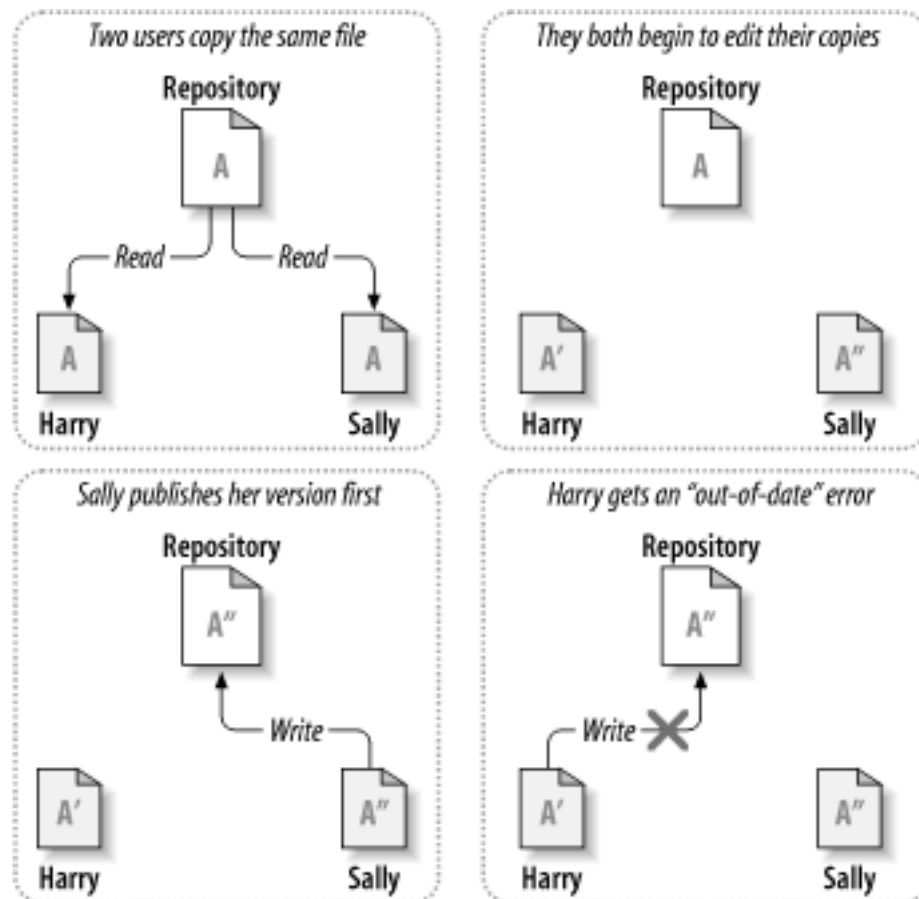
- *Zaklepanje lahko povzroča skrbniške težave.* Včasih Harry zaklene datoteko in nanjo pozabi. Medtem je Sally, ki želi spremeniti isto datoteko, prisiljena čakati. Harry gre na počitnice, Sally pa mora po pomoč k skrbniku sistema, da ji odklene datoteko. Takšna situacija povzroči precej nepotrebnih zamud in izgube časa.
- *Zaklepanje lahko povzroči nepotrebno delo eden za drugim.* Harry popravlja začetni del datoteke, Sally pa želi popravljati zadnji del iste datoteke. Te spremembe se ne prekrivajo. Oba uporabnika bi lahko datoteko spreminjala sočasno brez posledic, pod pogojem, da bi spremembe pravilno spojila skupaj. Nobene potrebe ni, da v takšni situaciji delata eden za drugim.
- *Zaklepanje ustvarja lažen občutek varnosti.* Predstavljajte si, da Harry zaklene in spreminja datoteko A, Sally pa hkrati zaklene in spreminja datoteko B. Recimo, da sta datoteki A in B odvisni ena od druge, in spremembe, narejene na posamezni datoteki, so semantično nezdružljive. Kar naenkrat datoteki A in B skupaj ne funkcionirata pravilno. Sistem zaklepanja je nemočen pri preprečevanju težave - hkrati pa ustvarja lažen občutek varnosti. Harry in Sally si mislita, da z zaklepanjem datotek začenjata varno, izolirano operacijo, kar povzroči, da se sploh ne pogovarjata o možni nezdružljivosti datotek.

2.2.3. Rešitev kopiraj-spremeni-spoji

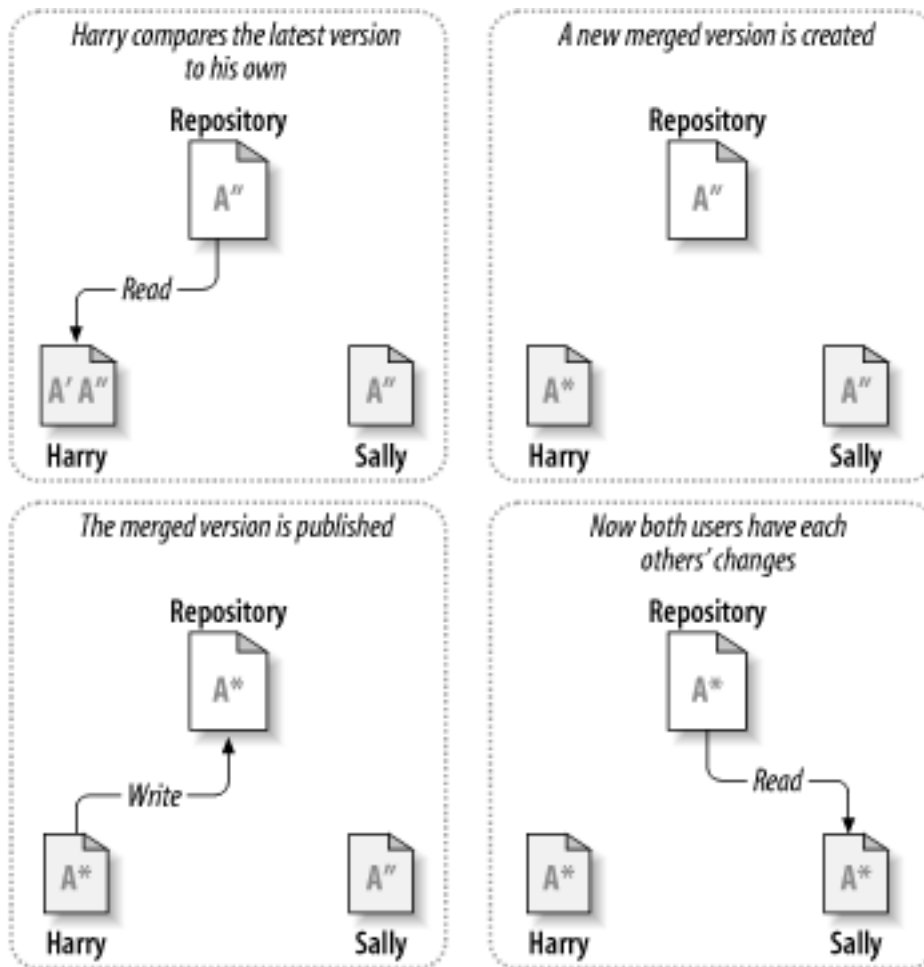
Subversion, CVS in še nekateri drugi sistemi za nadzor različic uporabljajo model *kopiraj-spremeni-spoji* kot alternativo zaklepanju datotek. Pri tem modelu ustvari vsak odjemalec osebno *delovno kopijo*

datoteke ali projekta. Uporabniki delajo sočasno in spreminjajo krajevne kopije. Nazadnje se zasebne kopije spojijo skupaj v novo, končno verzijo. Sistem za nadzor različic pogosto pomaga pri spajanju, vendar je koncu za pravilno spajanje odgovoren uporabnik.

Poglejmo primer. Harry in Sally ustvarita delovni kopiji istega projekta, prenesenega iz skladišča. Delata istočasno in naredita spremembe v isti datoteki A v svojih delovnih kopijah. Sally prva shrani spremembe v skladišče. Ko Harry kasneje poskusi shraniti svoje spremembe, ga skladišče obvesti, da je njegova datoteka A *zastarela*. Z drugimi besedami: datoteka A se je od takrat, ko jo je Harry prevzel iz skladišča, spremenila. Zato Harry zahteva od svojega odjemalca, da nove spremembe v skladišču *spoji* z datoteko A v njegovi delovni kopiji. Mogoče se spremembe ne prepletajo. Ko ima obe verziji združeni, shrani datoteko v skladišče.



Slika 2.4. Rešitev kopiraj-spremeni-spoji



Slika 2.5. ...Kopiraj-spremeni-spoji (nadaljevanje)

Kaj pa, če se Sallyjine spremembe *vseeno* prekrivajo s Harryjevimi? Nastane *sporna* situacija, ki pa običajno ne povzroča problemov. Ko Harry zahteva od svojega odjemalca, naj spoji zadnje spremembe v delovno kopijo, je njegova datoteka A označena kot sporna: Harry bo lahko videl oba nabora spornih sprememb in imel možnost izbrati pravilne. Upoštevajte, da program ne more samodejno reševati sporov; samo ljudje smo sposobni razmišljati in narediti ustrezne inteligentne odločitve. Ko Harry ročno reši prekrivajoče se spremembe (mogoče se je ob tem moral celo posvetovati s Sally!), lahko varno shrani spojeno datoteko v skladišče.

Model kopiraj-spremeni-spoji se mogoče zdi nekoliko kompliciran, vendar se v praksi izkaže, da funkcionira zelo dobro. Uporabniki lahko delajo sočasno, nikoli jim ni potrebno čakati nekoga drugega. Izkaže se, da se v primerih, ko delajo na istih datotekah, večina sprememb ne prekriva. Spori so redki. In čas, ki ga uporabniki porabijo za reševanje sporov, je mnogo krajši, kot bi bila izguba časa zaradi sistema zaklepanja.

Na koncu pridemo do enega samega ključnega dejavnika: komunikacija med uporabniki. Kjer se uporabniki malo pogovarjajo, se poveča število sintaktičnih in semantičnih sporov. Noben sistem ne more prisiliti uporabnikov, da bi idealno komunicirali med sabo in noben sistem ne more zaznati semantičnih sporov. Torej nima nobenega smisla živeti v lažnem prepričanju, da bo sistem zaklepanja preprečil spore. V praksi se izkaže, da zaklepanje zmanjša produktivnost bolj kot karkoli drugega.

Obstaja pa primer, v katerem se model zakleni-spremeni-odkleni izkaže za boljšega in sicer pri datotekah, ki se jih ne da spajati. Na primer: če vaše skladišče vsebuje grafične datoteke in dva uporabnika naredita

spremembe na isti datoteki naenkrat, se teh dveh sprememb ne da združiti. Eden od obeh uporabnikov bo spremembe izgubil.

2.2.4. Kaj naredi Subversion?

Subversion po privzetih nastavitvah uporablja model kopiraj-spremeni-spoji in v veliko primerih je to vse, kar potrebujete. Od različice 1.2 naprej pa Subversion podpira tudi zaklepanje datotek. Če torej delate na datotekah, ki se jih ne da spajati ali če ste takšen model prisiljeni uporabljati zaradi pravil, ki veljajo v vašem podjetju, vam Subversion ponuja možnosti, ki jih potrebujete.

2.3. Subversion v akciji

2.3.1. Delovne kopije

O delovnih kopijah ste že brali. Sedaj bomo prikazali, kako jih odjemalec Subversion ustvari in uporablja.

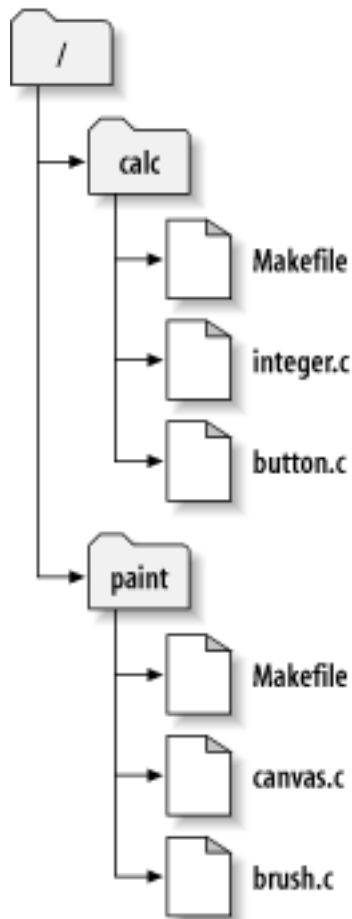
Delovna kopija sistema Subversion je običajna struktura map na vašem krajevnem sistemu, ki vsebuje datoteke. Datoteke lahko poljubno spreminjate. Če so to datoteke izvorne kode, lahko iz njih povsem običajno prevedete program. Vaša delovna kopija je vaše zasebno delovno območje: Subversion ne bo v vašo delovno kopijo nikoli vnašal sprememb, ki so jih naredili drugi uporabniki, prav tako vaših sprememb ne bo dal na voljo drugim, razen ko boste to izrecno zahtevali od njega.

Ko ste naredili določene spremembe v datotekah v delovni kopiji in jih preverili, uporabite ukaze sistema Subversion, da spremembe *objavite* (s pisanjem v skladišče), tako da so vidne ostalim sodelavcem, ki delajo na projektu. Če ostali sodelavci naredijo spremembe na datotekah, vam Subversion omogoča, da te spremembe spojite v delovno kopijo (z branjem iz skladišča).

Delovna kopija vsebuje tudi nekaj posebnih datotek, ki jih ustvari in vzdržuje Subversion za pomoč pri izvajanju ukazov. Vsaka mapa znotraj delovne kopije vsebuje podmapo `.svn`, ki ji rečemo *skrbniška mapa*. Datoteke znotraj te mape pomagajo sistemu Subversion prepoznati datoteke, ki vsebujejo neobjavljene spremembe, in datoteke, ki so jih spreminjali drugi uporabniki in so tako zastarele.

Tipično skladišče sistema Subversion pogosto vsebuje datoteke (ali izvorno kodo) za več projektov; običajno je vsak projekt podmapa v drevesni strukturi skladišča. Pri takšni ureditvi bo uporabnikova delovna kopija običajno ustrezala določenemu poddrevesu skladišča.

Recimo, da imate skladišče z dvema projektoma programske opreme.



Slika 2.6. Datotečni sistem skladišča

Z drugimi besedami, korenska mapa skladišča ima dve podmapi: `paint` in `calc`.

Če želite ustvariti delovno kopijo, morate *prevzeti* neko poddrevo skladišča. (*Prevzeti* se morda sliši, kot da vir podatkov zaklenete ali rezervirate, v resnici pa si s tem izdelate zasebno kopijo projekta).

Recimo, da naredite spremembo na datoteki `button.c`. Ker si mapa `.svn` zapomni datum spremembe datoteke in izvorno vsebino, Subversion ve, da ste datoteko spremenili. Vendar pa Subversion ne objavi sprememb, dokler mu tega ne ukažete. Temu rečemo *objava* (ali *vrnitev*) sprememb v skladišče.

Da bi omogočili dostop do svojih sprememb ostalim, uporabite ukaz **objavi**.

Sedaj so spremembe v datoteki `button.c` objavljene v skladišču; če nek drug uporabnik prevzame delovno kopijo iz mape `/calc`, bo te spremembe videl v zadnji različici datoteke.

Predpostavimo, da imate sodelavko Sally, ki je prevzela delovno kopijo `/calc` sočasno z vami. Ko objavite spremembe v datoteki `button.c`, se Sallyjina delovna kopija ne spremeni; Subversion spreminja delovne kopije le na zahtevo uporabnika.

Za posodobitev projekta Sally od sistema Subversion zahteva, da *posodobi* njeno delovno kopijo z uporabo ukaza **update**. Ta spoji vaše spremembe v njeno delovno kopijo, prav tako pa tudi vse ostale spremembe, ki so bile objavljene po njenem prevzemu.

Upoštevajte, da Sally ni potrebno povedati, katere datoteke želi posodobiti; Subversion uporabi informacije v mapi `.svn` in dodatne informacije iz skladišča, da se odloči, katere datoteke je potrebno posodobiti.

2.3.2. Naslovi URL skladišča

Do skladišč Subversion lahko dostopate na več različnih načinov - na krajevnem disku ali preko različnih omrežnih protokolov. Lokacija skladišča pa je vedno navedena z naslovom URL. Shema naslova URL pove način dostopa:

Shema	Način dostopa
file://	Neposreden dostop do skladišča na krajevnem ali omrežnem disku.
http://	Dostop do strežnika Apache s sistemom Subversion preko protokola WebDAV.
https://	Enako kot http://, vendar z enkripcijo SSL.
svn://	Dostop TCP/IP brez avtentikacije preko protokola po meri do strežnika svnservice
svn+ssh://	avtentificiran, enkriptiran dostop TCP/IP preko protokola po meri do strežnika svnservice

Tabela 2.1. Naslovi URL za dostop do skladišča

V večini primerov se v sistemu Subversion za naslove URL uporablja standardna sintaksa, kar omogoča, da se ime strežnika in številka vrat pojavljata kot del naslova URL. Metoda dostopa `file://` se običajno uporablja za lokalni dostop, možno pa jo je uporabljati tudi za poti UNC. URL je naslednje oblike: `file://gostitelj/pot/do/skladišča`. Za krajevne dostope dela `gostitelj` ne navedemo ali pa navedemo `localhost`. Iz tega razloga se krajevne poti običajno zapisujejo s tremi poševnicami: `file:///pot/do/skladišča`.

Poleg tega morajo uporabniki sheme `file://` na operacijskem sistemu Windows uporabljati neuradno "standardno" sintakso za dostopanje do skladišč, ki so na istem računalniku, vendar na drugem pogonu, kot je trenutni delovni pogon odjemalca. Naslednji dve različici sintakse poti URL sta pravilni; X je pogon, na katerem se nahaja skladišče:

```
file:///X:/pot/do/skladišča
...
file:///X|/pot/do/skladišča
...
```

Upoštevajte, da se za naslove URL uporablja poševnica, čeprav je znak za ločitev posameznih delov poti v sistemu Windows obrnjena poševnica.

Do skladišč tipa FSFS lahko varno dostopate preko omrežnih map v skupni rabi, vendar na ta način *ne morete* dostopati do skladišč tipa BDB.



Pozor

Ne ustvarjajte ali dostopajte do skladišč tipa Berkley DB v mapah skupne rabe v omrežju. Takšno skladišče *ne more* obstajati na oddaljenem datotečnem sistemu, četudi imate omrežni pogon preslikan v lokalni pogon. Če boste poskušali dostopati do skladišča tipa Berkley DB v omrežju, so rezultati nepredvidljivi - lahko se takoj pojavijo čudne napake, lahko pa traja mesece, preden boste opazili, da je vaša baza podatkov pokvarjena.

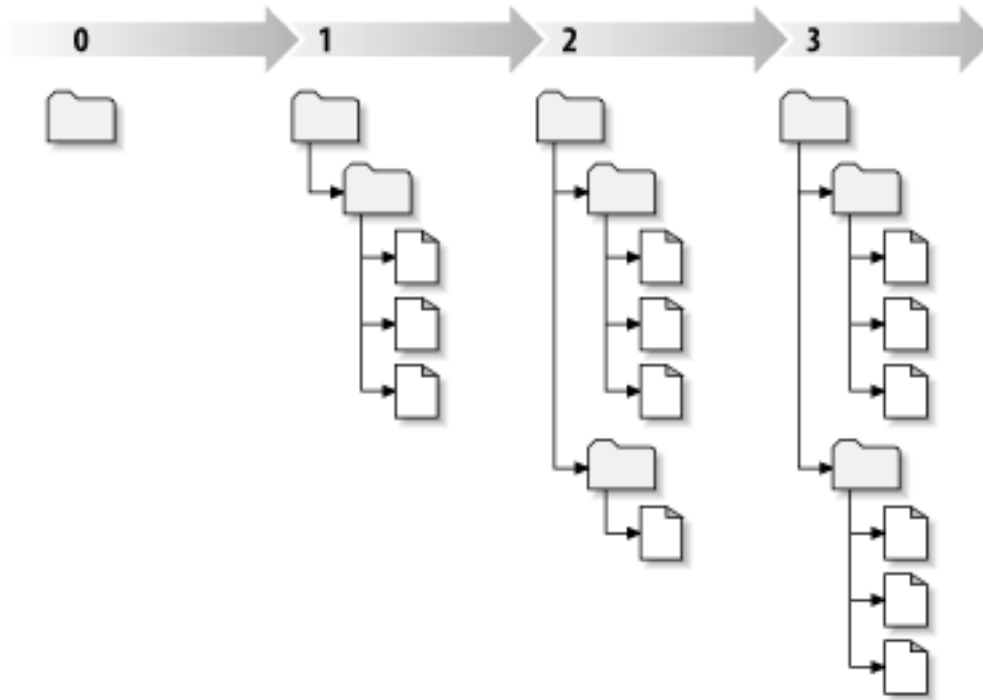
2.3.3. Revizije

Operacija **svn commit** objavi spremembe poljubnega števila datotek in map v eni atomični transakciji. V delovni kopiji lahko spreminjate vsebine datotek, ustvarjate, brišete, preimenujete in kopirate datoteke in potem objavite celoten nabor sprememb kot celoto.

V skladišču se vsaka objava opravi kot atomična transakcija: upoštevane so vse spremembe v objavi ali pa nobena. Subversion poskuša ohraniti atomičnost zaradi sesutja aplikacij, sistema, težav z omrežjem in ostalih uporabnikov posredovanj.

Vsakokrat, ko skladišče sprejme objavo, se ustvari novo stanje drevesa dototečne strukture, imenovano *revizija*. Vsaki reviziji se predpiše unikatno naravno število, za eno večje od številke prejšnje revizije. Začetna revizija novo ustvarjenega skladišča ima številko 0 in vsebuje zgolj korenko mapo.

Dober način za predstavitev skladišča je vrsta dreves. Predstavljajte si polje številke revizij, ki se začne z 0 in se razteza od leve proti desni. Iz vsake številke revizije visi drevo datotečnega sistema in vsako drevo je "posnetek" izgleda skladišča po vsaki objavi.



Slika 2.7. Skladišče

Globalne številke revizij

Za razliko od mnogih drugih sistemov za nadzor različic se revizije v sistemu Subversion nanašajo na *celotno drevo*, ne na posamezne datoteke. Vsaka številka revizije izbere celotno drevo - določeno stanje skladišča po neki objavi sprememb. Drug način gledanja je ta, da revizija N predstavlja stanje skladišča po N-ti objavi. Ko v sistemu Subversion govorimo o "reviziji 5 datoteke `foo.c`", v resnici mislimo "Datoteka `foo.c`, kot se nahaja v reviziji 5". Upoštevajte, da se v splošnem reviziji N in M neke datoteke *ne* razlikujeta.

Pomembno je poudariti, da delovne kopije niso vedno odraz samo eno revizije v skladišču; vsebujejo lahko datoteke več različnih revizij. Primer: recimo, da prevzamete delovno kopijo iz skladišča, kjer zadnja shranjena revizija nosi številko 4:

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

V tem trenutku delovna kopija ustreza točno reviziji številka 4 v skladišču. Denimo, da naredite spremembe v datoteki `button.c` in objavite spremembo. Denimo, da se med tem niso zgodile nobene druge objave. Vaša objava bo ustvarila revizijo 5 v skladišču in vaša delovna kopija bo izgledala takole:

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

Denimo, da v tem trenutku Sally objavi spremembo datoteke `integer.c` in s tem ustvari revizijo 6. Če uporabite ukaz **svn update**, ki posodobi vašo delovno kopijo, bo le-ta izgledala takole:

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

Sallyjine spremembe datoteke `integer.c` se bodo pojavile v vaši delovni kopiji in v datoteki `button.c` bodo še vedno vaše spremembe. V tem primeru je besedilo datoteke `Makefile` identično v revizijah 4, 5 in 6, vendar bo Subversion označil vašo delovno kopijo datoteke `Makefile` s številko revizije 6 in s tem povedal, je datoteka trenutna. Torej: ko boste naredili posodobitev vrhnje datoteke delovne kopije, bo le-ta ustrezala točno eni reviziji v skladišču.

2.3.4. Kako delovne kopije spremljajo skladišče

Za vsako datoteko v delovni kopiji Subversion shrani dve pomembni informaciji v skrbniško področje `.svn/`:

- na kateri reviziji je osnovana vaša delovna datoteka (rečemo ji datotečna *delovna revizija*) in
- časovni žig, ki pove, kdaj je bila krajevna kopija nazadnje posodobljena iz skladišča.

S temi informacijami in komunikacijo s skladiščem lahko Subversion ugotovi, v katerem od štirih stanj se delovna datoteka nahaja:

Nespremenjeno in trenutno

Datoteka je nespremenjena v delovni kopiji, prav tako pa od prevzema datoteke ni bila objavljena nobena sprememba v skladišču. Ukaz **objavi** ne bo storil ničesar, prav tako tudi ne ukaz **posodobi**.

Krajevno spremenjeno in trenutno

Datoteka v delovni kopiji je spremenjena, v skladišču pa ni bila objavljena nobena sprememba od osnovne revizije. Obstajajo krajevne spremembe, ki niso bile objavljene v skladišču, tako da bo ukaz **objavi** na datoteki objavil vaše spremembe, ukaz **posodobi** pa ne bo naredil ničesar.

Nespremenjeno in zastarelo

Datoteka ni bila spremenjena v delovni kopiji, se je pa spremenila v skladišču. Da bo datoteka enaka javni reviziji, jo je potrebno posodobiti. Ukaz **objavi** ne bo naredil ničesar, ukaz **posodobi** pa bo prenesel najnovejše spremembe iz skladišča v delovno kopijo.

Krajevno spremenjeno in zastarelo

Datoteka je bila spremenjena v delovni kopiji in v skladišču. Ukaz **objavi** ne bo uspel; prikazalo se bo sporočilo, da je datoteka *zastarela*. Datoteko je potrebno najprej posodobiti; ukaz **posodobi** bo poskušal javne spremembe spojiti s krajevnimi. Če Subversion ne more spojiti različic samodejno, reševanje spora prepusti uporabniku.

2.4. Povzetek

V tem poglavju smo pregledali nekaj osnovnih principov sistema Subversion:

- Predstavili smo koncept centralnega skladišča, delovne kopije na strani odjemalca in polja dreves revizij v skladišču.
- Videli smo nekaj primerov, kako lahko dva sodelavca uporabljata Subversion in model 'kopiraj-spremeni-spoj' za objavljanje sprememb in sprejemanje sprememb drug od drugega.

- Govorili smo o tem, kako Subversion sledi in upravlja informacije v delovni kopiji.

Poglavje 3. Skladišče

Ne glede na uporabljen protokol za dostop do skladišča morate vedno ustvariti vsaj eno skladišče. To lahko storite preko ukazne vrstice (Subversion) ali preko grafičnega vmesnika (TortoiseSVN).

Če še niste ustvarili skladišča Subversion, je sedaj čas, da to storite.

3.1. Ustvarjanje skladišča

You can create a repository with the FSFS backend or with the older Berkeley Database (BDB) format. The FSFS format is generally faster and easier to administer, and it works on network shares and Windows 98 without problems. The BDB format was once considered more stable simply because it has been in use for longer, but since FSFS has now been in use in the field for several years, that argument is now rather weak. Read *Choosing a Data Store* [<http://svnbook.red-bean.com/en/1.5/svn.reposadmin.planning.html#svn.reposadmin.basics.backends>] in the Subversion book for more information.

3.1.1. Ustvarjanje skladišča z odjemalcem za ukazno vrstico

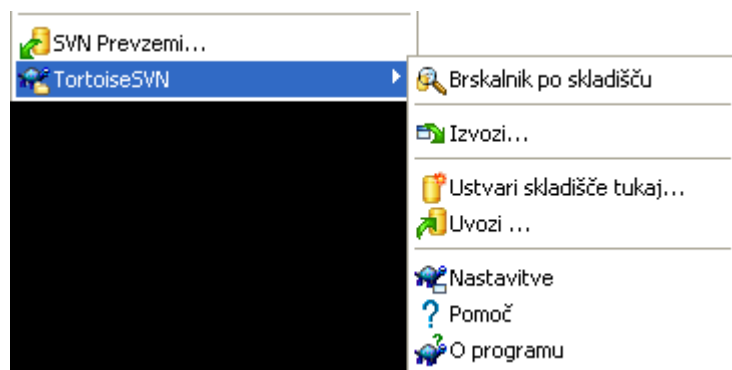
1. Ustvarite prazno mapo z imenom SVN (n.pr. D:\SVN\), ki se uporablja kot korenska mapa za vsa vaša skladišča.
2. Ustvarite še eno mapo iz imenom MojeNovoSkladišče znotraj D:\SVN\.
3. Odprite ukazni poziv (ali okno DOS), spremenite trenutno mapo v D:\SVN\ in vtipkajte

```
svnadmin create --fs-type bdb MojeNovoSkladišče  
ali
```

```
svnadmin create --fs-type fsfs MojeNovoSkladišče
```

Sedaj imate novo skladišče na lokaciji D:\SVN\MojeNovoSkladišče.

3.1.2. Ustvarjanje skladišča s programom TortoiseSVN



Slika 3.1. Menu TortoiseSVN za datoteke brez različic

1. Odprite Raziskovalca

2. Ustvarite novo mapo in jo poimenujte npr. SkladiščeSVN
3. Na novo ustvarjeno mapo kliknite z desnim gumbom in izberite TortoiseSVN → Ustvari skladišče tukaj....

Skladišče se ustvari znotraj nove mape. *Datotek v v novi mapi ne urejajte sami!!!*. Če se pojavijo kakšne napake, preverite, ali je mapa res prazna in da ni zaščitena proti pisanju.



Namig

TortoiseSVN ne ponuja več možnosti ustvarjanja skladišč BSB, lahko pa še vedno uporabite odjemalca ukazne vrstice in z njim ustvarite skladišče. Skladišča FSFS so na splošno lažja za vzdrževanje, prav tako pa nam razvijalcem programa TortoiseSVN olajšajo vzdrževanje programa (zaradi težav s kompatibilnostjo različnih revizij BDB).

Prihodnje različice TortoiseSVN ne bodo podpirale dostopa `file:///` do BDB skladišč zaradi težav s kompatibilnostjo, bodo pa vedno podpirale ta format skladišča za dostope do strežnika preko protokolov `svn://`, `http://` ali `https://`. Zato priporočamo, da za nova skladišča uporabite sistem FSFS, če želite do njih dostopati preko protokola `file://`.

Seveda priporočamo, da sploh ne uporabljate dostopa `file://`, razen za preizkušanje. Uporaba strežnika je bolj varna in zanesljiva, razen če skladišče uporablja en sam razvijalec.

3.1.3. Krajevni dostop do skladišča

Za dostop do krajevnega skladišča potrebujete le pot do skladišča. Zapomnite si, da Subversion pričakuje pot do skladišč oblike `file:///C:/SkladiščeSVN/`. Upoštevajte, da morate uporabljati poševnico in ne obrnjeno poševnico.

Za dostop do skladišča na mapi v skupni rabi v omrežju lahko uporabite preslikavo pogona ali pa pot UNC. Za poti UNC je oblika naslova URL `file://ImeStrežnika/pot/do/skladišča/`. Upoštevajte, da je potrebno na začetku uporabiti le dve poševnici.

Pred različico Subversion 1.2 so morale biti poti UNC podane v nekoliko bolj obskurni obliki `file:///\\ImeStrežnika/pot/do/skladišča`. Ta oblika je še vedno možna, vendar ni priporočljiva.



Pozor

Ne ustvarjajte skladišč tipa Berkley DB v mapah skupne rabe v omrežju, prav tako ne dostopajte do njih. Takšno skladišče *ne more* obstajati na oddaljenem datotečnem sistemu, četudi imate omrežni pogon preslikan v lokalni pogon. Če boste poskušali dostopati do skladišča tipa Berkley DB v mapi v skupni rabi, so rezultati nepredvidljivi - lahko takoj vidite čudne napake, lahko pa traja mesece, preden boste opazili, da je vaša baza podatkov pokvarjena.

3.1.4. Dostop do skladišča na deljenem omrežnem pogonu

Čeprav je teoretično možno postaviti skladišče FSFS na omrežnem deljenem pogonu in dovoliti uporabnikom dostop preko protokola `file://`, to *ni* priporočljivo. V bistvu ta način *močno* odsvetujemo in ga ne podpiramo.

Prva težava je, da uporabniku omogočite popoln dostop do skladišča, tako da ga lahko ta po pomoti izbriše ali pa ga pokvari s kakšno drugo operacijo.

Druga težava je, da vsi omrežni protokoli ne omogočajo zaklepanja datotek, ki ga Subversion za svoje delovanje potrebuje. Zato se skladišče lahko pokvari. Mogoče se to ne bo zgodilo takoj, vendar bosta slej ko prej dva uporabnika dostopala do skladišča istočasno.

Tretja težava je, da je potrebno nastaviti ustrezne pravice na datotekah. Na navadnem omrežnem pogonu sistema Windows bo to morda celo delovalo, pri uporabi strežnika SAMBA pa nastanejo težave.

Protokol `file://` je namenjen krajevnemu dostopu za enega uporabnika, predvsem za testiranje in razhroščevanje. Če želite deliti skladišče, morate *zares* razmisliti o postavitvi strežnika. To pa sploh ni tako težko, kot se sliši. Za več informacij o izbiri in navodila za postavitve preberite [Razdelek 3.5, "Accessing the Repository"](#).

3.1.5. Postavitev skladišča

Preden uvozite podatke v skladišče, premislite, kako jih boste organizirali. Če uporabite katero od priporočenih postavitvev, vam bo kasneje lažje.

Obstaja nekaj standardnih, priporočenim načinov, kako organizirati skladišče Subversion. Večina ljudi naredi mapo `trunk` za "glavno vejo" razvoja, mapo `branches` za stranske veje in mapo `tags`, ki vsebuje oznake. Če se v skladišču nahaja samo en projekt, uporabniki pogosto ustvarijo naslednjo strukturo vrhnjih map:

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

Če skladišče vsebuje več projektov, uporabniki pogosto indeksirajo njihovo postavitev glede na veje:

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

... ali po projektu:

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

Indeksiranje po projektih je smiselno, če projekti med sabo niso povezani in se prevzemajo neodvisno drug od drugega. Za povezane projekte, kjer želite narediti prevzem z eno potezo, ali pa v primeru, kjer so projekti povezani v en distribucijski paket, pa je indeksiranje po vejah pogosto boljša odločitev. V tem primeru imate le eno glavno vejo (`trunk`), relacije med podprojekti pa so lažje vidne.

Če izberete izgled `/trunk /tags /branches`, to ne pomeni, da morate prekopirati celotno glavno vejo za vsako vejo ali oznako, tako da ta struktura prinaša več prilagodljivosti.

Za nepovezane projekte boste verjetno uporabili različna skladišča. Ko objavite spremembe, se spremeni številka revizije celotnega skladišča, ne številka revizije projekta. Če imate v enem skladišču dva projekta, lahko to vodi do velikih "lukenj" pri številkah revizij. Projekta Subversion in TortoiseSVN se nahajata na istem strežniku, vendar sta postavljena v različna skladišča, kar omogoča neodvisen razvoj in reši zmedo glede številke gradenj.

Seveda vam teh postavitev ni treba upoštevati. Lahko naredite kakršnokoli variacijo, karkoli najbolj ustreza vam ali vaši ekipi sodelavcev. Postavitev, ki jo izberete, pa ni nujno trajna. Skladišče lahko kadarkoli na novo organizirate. Ker so veje in oznake navadne mape, jih lahko TortoiseSVN premakne ali preimenuje po vaših željah.

Sprememba iz enega načina postavitve v drugega se naredi z nizom premikov na strani strežnika. Če vam ni všeč, kako so stvari v skladišču organizirane, enostavno premaknite mape.

Če še niste ustvarili osnovne strukture map znotraj skladišča, je čas, da to storite sedaj. Ostajata dva načina. Če želite ustvariti le strukturo `/trunk /tags /branches`, lahko za to uporabite brskalnik po skladišču, kjer v posamičnih objavah naredite zgornje mape. Če želite izdelati globljo hierarhijo, potem je enostavneje, da strukturo map naredite na disku in jo nato v eni objavi uvozite:

1. ustvarite novo prazno mapo na trdem disku
2. ustvarite željeno strukturo map vrhnjega nivoja znotraj mape - zaenkrat ne vnašajte datotek!
3. uvozite to strukturo v skladišče - desno kliknite na mapo in izberite TortoiseSVN → Uvozi... S tem uvozite začasno mapo v korensko mapo skladišča in ustvarite osnovno postavitev.

Upoštevajte, da se ime mape, ki jo uvažate, ne pojavi v skladišču, ampak le njena vsebina. Naprimer: ustvarite naslednjo strukturo:

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

Uvozite `C:\Temp\New` v korensko mapo skladišča, ki potem izgleda takole:

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

3.2. Varnostna kopija skladišča

Neglede na tip skladišča, ki ga uporabljate, je izjemno pomembno, da redno ustvarjate varnostne kopije in da jih tudi preverite. Če je s strežnikom kaj narobe, boste mogoče lahko dobili zadnjo verzijo datotek, vendar je brez skladišča zgodovina vaših datotek za vedno izgubljena.

Najbolj enostaven način (vendar ni priporočljiv) je, da mape in datoteke skladišča prekopirate na medij, ki ga uporabljate za varnostno shranjevanje. V tem primeru noben proces ne sme dostopati do podatkov v skladišču. Dostop pomeni *kakršenkoli* dostop. V skladišče tipa BDB se podatki zapisujejo tudi, ko operacija zahteva le branje, n. pr. pridobivanje stanja. Če se med kopiranjem zgodi dostop do podatkov (spletni brskalnik je ostal odprt, WebSVN in podobno), bo varnostna kopija neuporabna.

Priporočen način je uporaba ukaza

```
svnadmin hotcopy pot/do/skladišča pot/do/varnostne/kopije --clean-logs
```

S tem naredite kopijo skladišča na varen način. Potem naredite varnostno kopije kopije. Stikalo `--clean-logs` ni potrebno, vendar odstrani vse odvečne datoteke za logiranje, ko naredite varnostno kopijo skladišča BDB, s čemer lahko prihranite nekaj prostora.

Orodje `svnadmin` se namesti samodejno ob namestitvi odjemalca Subversion za ukazno vrstico. Če nameščate odjemalca za ukazno vrstico na operacijskem sistemu Windows, je najbolje, da si prenesete namestitveni program. Stisnjen je bolj učinkovito kot različica `.zip`, tako da je prenos hitrejši, poleg

tega pa namesto vas opravi nastavitve. Najnovejša različica odjemalca Subversion za ukazno vrstico se nahaja na naslovu <http://subversion.apache.org/getting.html>.

3.3. Server side 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/1.5/svn.reposadmin.create.html#svn.reposadmin.create.hooks>] in the Subversion Book for full details about the hooks which are implemented.

Ukazne datoteke akcij izvede strežnik, ki gosti skladišče. TortoiseSVN pa vam omogoča, da nastavite ukazne datoteke akcij na strani odjemalca. Te datoteke se izvedejo krajevno ob določenih dogodkih. Za več informacij pogledajte **Razdelek 4.30.8, "Ukazne datoteke akcij na strani odjemalca"**.

Primere ukaznih datotek akcij najdete v podmapi `hooks` skladišča. Ti primeri so narejeni za sisteme Unix/Linux, tako da jih morate prilagoditi, če uporabljate strežnik s sistemom Windows. Akcija je lahko paketna datoteka ali izvršna datoteka. Spodnji primer prikazuje skripto, ki jo lahko uporabite kot akcijo `pre-revprop-change`.

```
rem Dovolj samo spremembe sporočil dnevniških zapisov.
if "%4" == "svn:log" exit 0
echo Lastnosti '%4' ne morete spremeniti >&2
exit 1
```

Upoštevajte, da se sporočila, ki jih pošljete na standardni izhod (stdout), ne prikažejo. Če želite, da se besedilo prikaže v oknu Objava zavrnjena, ga morate poslati na stderr. V sistemski skripti to dosežete z uporabo niza `>&2` na koncu sporočila.

3.4. Povezave za prevzem

Če želite ponuditi skladišče Subversion tudi ostalim uporabnikom, lahko povezavo vključite v svojo spletno stran. Da jo naredite bolj dostopno, lahko dodate *prevzemno povezavo* za uporabnike TortoiseSVN.

Ko namestite TortoiseSVN, ta registrira nov protokol `tsvn:`. Ko uporabnik klikne na takšno povezavo, se samodejno odpre okno za prevzem z že izpolnjenim naslovom URL skladišča.

Če želite dodati takšno povezavo na svojo HTML stran, ji dodajte naslednjo kodo:

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

Seveda bo vse skupaj izgledalo še bolje, če dodate primerno sliko. Uporabite lahko *logotip programa TortoiseSVN* [<http://tortoisesvn.tigris.org/images/TortoiseCheckout.png>] ali pa svojo sliko.

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

You can also make the link point to a specific revision, for example

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


3.5. Accessing the Repository

To use TortoiseSVN (or any other Subversion client), you need a place where your repositories are located. You can either store your repositories locally and access them using the `file://` protocol or you can place them on a server and access them with the `http://` or `svn://` protocols. The two server protocols can also be encrypted. You use `https://` or `svn+ssh://`, or you can use `svn://` with SASL.

If you are using a public hosting service such as *Google Code* [<http://code.google.com/hosting/>] or your server has already been setup by someone else then there is nothing else you need to do. Move along to [Poglavje 4, Dnevna uporaba](#).

If you don't have a server and you work alone, or if you are just evaluating Subversion and TortoiseSVN in isolation, then local repositories are probably your best choice. Just create a repository on your own PC as described earlier in [Poglavje 3, Skladišče](#). You can skip the rest of this chapter and go directly to [Poglavje 4, Dnevna uporaba](#) to find out how to start using it.

If you were thinking about setting up a multi-user repository on a network share, think again. Read [Razdelek 3.1.4, "Dostop do skladišča na deljenem omrežnem pogonu"](#) to find out why we think this is a bad idea. Setting up a server is not as hard as it sounds, and will give you better reliability and probably speed too.

The next sections are a step-by-step guide on how you can set up such a server on a Windows machine. Of course you can also set up a server on a Linux machine, but that is beyond the scope of this guide. 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/1.5/svn.serverconfig.html>].

3.6. Strežnik svnserve

3.6.1. Uvod

Subversion includes Svnserve - a lightweight stand-alone server which uses a custom protocol over an ordinary TCP/IP connection. It is ideal for smaller installations, or where a full blown Apache server cannot be used.

In most cases svnserve is easier to setup and runs faster than the Apache based server, although it doesn't have some of the advanced features. And now that SASL support is included it is easy to secure as well.

3.6.2. Nameščanje svnserve

1. Najnovejšo različico sistema Subversion prenesite z naslova <http://subversion.apache.org/getting.html>. Prenesete pa lahko tudi namestitveni program iz strežnika CollabNet na naslovu <http://www.collab.net/downloads/subversion>. Namestitveni program namesti svnserve kot storitev Windows, vsebuje pa tudi nekaj orodij, ki jih potrebujete za varno delovanje preko SASL.
2. Če je Subversion že nameščen in svnserve trenutno teče, ga morate pred nadaljevanjem zaustaviti.
3. Zaženite namestitveni program Subversion. Če nameščate Subversion na strežniku (priporočamo), lahko korak 4 preskočite.
4. Odprite Raziskovalca, pojdite na namestitveno mapo Subversion (običajno `C:\Program Files\Subversion`) in v mapi `bin` poiščite datoteke `svnserve.exe`, `intl3_svn.dll`, `libapr.dll`, `libapriconv.dll`, `libapriutil.dll`, `libdb*.dll`, `libeay32.dll` in `ssleay32.dll` - prekopirajte jih ali pa prekopirajte celopno mapo `bin` v mapo na vašem strežniku, n. pr. `c:\svnserve`

3.6.3. Poganjanje svnservice

Sedaj, ko je svnservice nameščen, ga morate na strežniku zagnati. Najlažji način je, da v lupini DOS izvedete naslednji ukaz ali narediti bližnjico:

```
svnservice.exe --daemon
```

svnservice bo sedaj čakal na prihajajoče zahteve na vratih 3690. Stikalo --daemon pove, naj svnservice teče kot skriti proces, dokler ga ročno ne zaustavimo.

Če še niste ustvarili skladišča, sledite navodilom, podanim pri namestitvi strežnika Apache: [Razdelek 3.7.4, "Konfiguracija"](#).

Če želite preveriti, ali svnservice teče, uporabite TortoiseSVN → Skladiščni brskalnik - videti morate skladišče.

Če predpostavimo, da se skladišče nahaja v mapi c:\sklad\TestnoSkladišče, in je vašemu strežniku ime localhost, vpišite:

```
svn://localhost/sklad/TestnoSkladišče
```

ko dobite poziv brskalnika po skladišču.

Z uporabo stikala --root lahko povečate varnost in prihranite čas pri vnašanju naslovov URL. S tem določite korensko lokacijo in omejite dostop na izbrano mapo na strežniku:

```
svnservice.exe --daemon --root pogon:\pot\do\skladišča\root
```

Če uporabimo prejšnji primer, se ukaz za zagon svnservice sedaj glasi:

```
svnservice.exe --daemon --root c:\sklad
```

V skladiščnem brskalniku TortoiseSVN pa se naslov URL skrajša na:

```
svn://localhost/TestnoSkladišče
```

Upoštevajte, da morate stikalo --root uporabiti tudi takrat, ko se skladišče nahaja na drugi particiji ali pogonu kot svnservice.

Svnservice lahko streže poljubnemu številu skladišč. Postavite jih nekam pod korensko mapo, ki ste jo pravkar definirali, in dostopajte do njih z uporabo relativnega naslova URL glede na korenskega.



Pozor

Ne ustvarjajte ali dostopajte do skladišč tipa Berkley DB v mapah skupne rabe v omrežju. Takšno skladišče *ne more* obstajati na oddaljenem datotečnem sistemu, četudi imate omrežni pogon preslikan v lokalni pogon. Če boste poskušali dostopati do skladišča tipa Berkley DB v omrežju, so rezultati nepredvidljivi - lahko se takoj pojavijo čudne napake, lahko pa traja mesece, preden boste opazili, da je vaša baza podatkov pokvarjena.

3.6.3.1. Zaženite svnservice kot storitev

Poganjanje svnservice kot nek uporabnik običajno ni najboljša rešitev. To pomeni, da mora biti uporabnik stalno prijavljen na strežniku, prav tako se mora uporabnik prijaviti po ponovnem zagonu. Boljši način

je poganjanje svnservice kot storitev (service) Windows. Od različice Subversion 1.4 se svnservice lahko namesti kot domorodna storitev (native windows service).

Za namestitev svnservice kot domorodno storitev (native windows service) izvršite naslednji ukaz v eni vrstici. S tem ustvarite storitev, ki se samodejno zažene ob zagonu sistema.

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

Če katera od poti vsebuje presledke, morate uporabiti poševnice in navednice. Primer:

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

Ko ste storitev ustvarili, jih lahko dodate tudi opis. Le ta se potem pojavi v upravljalcu storitev (Windows Service Manager).

```
sc description svnservice "Strežnik Subversion (svnservice)"
```

Pazite na nekoliko nenavadno obliko ukaza sc. Pri podajanju parametrov ključ= vrednost ne sme biti med ključem in znakom = nobenega presledka, presledek pa se mora nahajati pred vrednostjo.



Namig

Microsoft po novem priporoča, da se storitve poganjajo pod računom Lokalna storitev (Local Service) ali Omrežna storitev (Network Service). Preverite *The Services and Service Accounts Security Planning Guide* [<http://www.microsoft.com/technet/security/topics/serversecurity/serviceaccount/default.mspx>]. Za ustvarjanje storitve pod računom Lokalna storitev dodajte zgornjemu primeru naslednjo vrstico.

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

Upoštevajte, da morate dodati računu potrebne pravice za Subversion in skladišča, prav tako pa tudi za vse aplikacije, ki jih kličejo ukazne datoteke akcij. Vgrajena skupina za to se imenuje "LOCAL SERVICE".

Ko ste namestili storitev, morate v urejevalnik storitev, da jo zaženete (samo tokrat; po ponovnem zagonu strežnika se bo pognala samodejno).

Za ostale informacije kliknite na *Windows Service Support for Svnservice* [<http://svn.collab.net/repos/svn/trunk/notes/windows-service.txt>].

Če ste predhodnjo različico svnservice namestili z uporabo ovoja SVNService, sedaj pa želite uporabiti storitev Windows, morate preklicati registracijo ovoja kot storitve (ne pozabiti storitve najprej zaustaviti!). Za odstranitev registracije iz registra uporabite naslednji ukaz:

```
svnservice -remove
```

3.6.4. Osnovna avtentikacija s svnservice

Po privzetih nastavitvah svnservice omogoča anonimen dostop samo za branje. To pomeni, da lahko uporabite naslov `svn://` za prevzem in posodobitev skladišča ali skladiščni brskalnik za pregledovanje skladišča, ne morete pa objavljati narejenih sprememb.

Da bi omogočili pisanje v skladišče, morate popraviti datoteko `conf/svnserve.conf` v mapi skladišča. Ta datoteka vsebuje nastavitve `svnserve`, poleg tega pa tudi koristno dokumentacijo.

Možnost pisanja anonimnim uporabnikom lahko omogočite z enostavno nastavitvijo:

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

Vendar pa v tem primeru ne veste, kdo je spremembo vpisal, saj bo lastnost `svn:author` prazna. Prav tako ne boste mogli določati, kdo lahko spremembe v skladišče zapisuje. To je nevarno!

Rešitev problema je ustvarjanje baze gesel:

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

`datotekauporabnikov` je datoteka, ki se nahaja v isti mapi kot datoteka `svnserve.conf`. Datoteka se lahko nahaja tudi drugje (kar je uporabno, če imate več skladišč, kjer potrebujete iste pravice za dostop) in jo navedete z absolutno potjo ali potjo, relativno na mapo `conf`. Če vključite pot, mora biti navedena `/po/pravilih/sistema/unix`. Uporaba znaka `\` ali črk pogona ni dovoljena. Datoteka `datotekauporabnikov` mora imeti naslednjo strukturo:

```
[users]
uporabniško ime = geslo
...
```

Ta primer prepove dostop vsem anonimnim uporabnikom (takšnim, ki se niso avtentificirali), uporabnikom, ki so navedeni v datoteki `datotekauporabnikov`, pa dovoli branje in pisanje po skladišču.



Namig

If you maintain multiple repositories using the same password database, the use of an authentication realm will make life easier for users, as TortoiseSVN can cache your credentials so that you only have to enter them once. More information can be found in the Subversion book, specifically in the sections [Create a 'users' file and realm](http://svnbook.red-bean.com/en/1.5/svn.serverconfig.svnserve.html#svn.serverconfig.svnserve.auth.users) [http://svnbook.red-bean.com/en/1.5/svn.serverconfig.svnserve.html#svn.serverconfig.svnserve.auth.users] and [Client Credentials Caching](http://svnbook.red-bean.com/en/1.5/svn.serverconfig.netmodel.html#svn.serverconfig.netmodel.credcache) [http://svnbook.red-bean.com/en/1.5/svn.serverconfig.netmodel.html#svn.serverconfig.netmodel.credcache]

3.6.5. Boljša varnost s SASL

3.6.5.1. Kaj je SASL?

The Cyrus Simple Authentication and Security Layer je odprtokodni projekt univerze Carnegie Mellon, ki kateremukoli protokolu doda generično avtentikacijo in enkripcijo. Od različice 1.5 naprej znata to knjižnico uporabljati tako `svnserve` kot odjemalec TortoiseSVN.

For a more complete discussion of the options available, you should look at the Subversion book in the section [Using svnserve with SASL](http://svnbook.red-bean.com/en/1.5/svn.serverconfig.svnserve.html#svn.serverconfig.svnserve.sasl) [http://svnbook.red-bean.com/en/1.5/svn.serverconfig.svnserve.html#svn.serverconfig.svnserve.sasl]. If you are just looking for a simple way to set up secure authentication and encryption on a Windows server, so that your repository can be accessed safely over the big bad Internet, read on.

3.6.5.2. Avtentikacija SASL

Za aktivacijo določenih mehanizmov SASL na strežniku morate narediti tri stvari. Najprej ustvarite sekcijo `[sasl]` v datoteki `svnserve.conf`, ki se nahaja znotraj skladišča, pri tem uporabite naslednjo vrednost:

```
use-sasl = true
```

Nato ustvarite datoteko `svn.conf` na primerni lokaciji - ponavadi je to v namestitveni mapi Subversion.

Nazadnje ustvarite vnose v registru, ki SASL povejo, kje naj najde stvari. Ustvarite vnos `[HKEY_LOCAL_MACHINE\SOFTWARE\Carnegie Mellon\Project Cyrus\SASL Library]`, vanj pa dodajte dve novi vrednosti (niz/string): `SearchPath` nastavite na mapo, ki vsebuje vtičnike `sasl*.dll` (običajno je to namestitvena mapa Subversion), in `ConfFile` nastavite na mapo, ki vsebuje datoteko `svn.conf`. Če ste za namestitev uporabili namestitveni program s strežnika CollabNet, imate te vnose že v registru.

Datoteki `svn.conf` dodajte naslednje besedilo:

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

Zadnja vrstica prikazuje lokacijo avtentikacijske baze podatkov, ki se imenuje `sasldb`. Nahaja se lahko kjerkoli, vendar je najbolj primerna lokacija nadrejena mapa skladišča. Poskrbite, da ima storitev `svnserve` pravice za branje te datoteke.

Če storitev `svnserve` že teče, jo morate zaustaviti in zagnati pogovno, da prebere posodobljene namestitvene podatke.

Sedaj, ko je namestitev zaključena, lahko ustvarite uporabnike in gesla. Za to potrebujete program `saslpasswd2`. Če ste uporabili namestitveni program CollabNet, se ta program nahaja v namestitveni mapi. Uporabite naslednjo ukazno vrstico:

```
saslpasswd2 -c -f C:\TortoiseSVN\sasldb -u realm uporabniškoime
```

Stikalo `-f` pove, da podajate lokacijo baze podatkov, vrednost `realm` mora biti enaka vrednosti, ki ste jo navedli v datoteki `svnserve.conf` skladišča, uporabniškoime pa je željeno uporabniško ime uporabnika. Upoštevajte, da `realm` ne sme vsebovati presledkov.

Seznam uporabniških imen v bazi lahko izpišete s programom `sasldblistusers2`.

3.6.5.3. Enkripcija SASL

Za določanje različnih nivojev enkripcije lahko v datoteki `svnserve.conf` skladišča nastavite dve vrednosti:

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

Vrednosti `min-encryption` in `max-encryption` določata nivo enkripcije, ki jo zahteva strežnik. Če želite enkripcijo onemogočiti, nastavite obe vrednosti na 0. Če želite omogočiti preverjanje kontrolne vsote (torej preprečiti spreminjanje in zagotoviti integriteto podatkov brez enkripcije), nastavite obema spremenljivkama vrednost 1. Če želite omogočiti (vendar ne zahtevati) enkripcijo, nastavite najmanjšo

vrednost na 0, največjo pa na število bitov enkripcije. Za brezpogojno uporabo enkripcije nastavite obe spremenljivki na vrednost več kot 1. V prejšnjem primeru zahtevamo vsaj 128 bitno enkripcijo, hkrati pa največ 256 bitno.

3.6.6. Avtentikacija s svn+ssh

Drug način za avtentikacijo uporabnikov s strežnikom na osnovi svnservice je uporaba SSH (secure shell) za prenos zahtevkov s pomočjo tunnelskega načina. Ta način ni tako enostaven kot SASL, vendar je lahko uporaben v določenih primerih.

S tem pristopom se svnservice ne zaganja kot skriti proces, ampak varnostna lupina zažene svnservice za vas, poganja pa ga kot avtentificiran uporabnik SSH. Na strežniku potrebujete skriti proces za varnostno lupino (secure shell daemon).

Osnovna navodila za postavitve strežnika so podana v poglavju [Dodatek G, Zavarovanje strežnika Svnservice z SSL](#). Podrobnosti o SSH najdete v pogosto zastavljenih vprašanjih. Iščite niz "SSH".

Več informacij o svnservice lahko dobite na naslovu [Version Control with Subversion](#) [<http://svnbook.red-bean.com>].

3.6.7. Avtorizacija svnservice na osnovi poti

Od različice Subversion 1.3 najprej svnservice omogoča uporabo istega modula `mod_authz_svn` za avtorizacijo na osnovi poti, kot je na voljo v strežniku Apache. Spremeniti morate datoteko `conf/svnservice.conf` v mapi skladišča in dodati vrstico, ki pove, kje se nahaja avtorizacijska datoteka.

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

`authz` je datoteka, ki jo ustvarite za določanje pravic dostopa. Za vsako skladišče lahko ustvarite svojo datoteko, lahko pa uporabite skupno datoteko za več skladišč. Preberite [Razdelek 3.7.6, "Avtorizacija na osnovi poti"](#), kjer je opisana oblika datoteke.

3.7. Strežnik na osnovi strežnika Apache

3.7.1. Uvod

Najbolj prilagodljiv sistem za Subversion dobite pri uporabi strežnika Apache. Njegova namestitve je sicer nekoliko bolj zahtevna, vendar ima v primerjavi z ostalimi strežniki številne prednosti:

WebDAV

Strežniki Subversion, postavljeni na strežniku Apache, uporabljajo protokol WebDAV, ki ga podpirajo številni ostali programi. Ta omogoča, da naprimer priklopite skladišče kot "internetno mapo" v Windowsovem Raziskovalcu in do nje dostopate kot do katerekoli druge mape v datotečnem sistemu

Brskanje po skladišču

V spletni brskalnik lahko vpišete naslov URL skladišča in brskate po vsebini, ne da bi imeli nameščenega odjemalca za Subversion. To omogoča dostop do podatkov precej širšemu krogu uporabnikov.

Avtentikacija

Uporabite lahko prav vse avtentikacijske mehanizme, ki jih podpira Apache, vključno z SSPI in LDAP.

Varnost

Ker je strežnik Apache zelo stabilen in varen, je prav tako varno tudi vaše skladišče. Uporabite lahko tudi enkripcijo SSL.

3.7.2. Namestitev strežnika Apache

Za namestitev strežnika Apache potrebujete računalnik s sistemom Windows 2000, Windows XP+SP1, Windows 2003, Vista ali Server 2008.



Pozor

Prosimo, upoštevajte, da lahko uporaba operacijskega sistema Windows XP brez prvega paketa popravkov (service pack 1) pripelje do nepravilnih podatkov v omrežju in tako pokvari skladišče!

1. Prenesite si zadnjo različico spletnega strežnika Apache z naslova <http://httpd.apache.org/download.cgi>. Poskrbite, da prenesete različico 2.2.x - različica 1.3.x.x ne bo delovala!

Namestitveni program MSI za spletni strežnik Apache dobite, če kliknete na `other files` in greste na pot `binaries/win32`. Lahko izbere datoteko `apache-2.2.x-win32-x86-openssl-0.9.x.msi` (vključuje tudi OpenSSL).

2. Ko ste prenesli namestitveni program za strežnik Apache 2, ga poženite in vodil vas bo skozi namestitveni proces. Preverite, da ste pravilno vnesli naslov URL strežnika (če strežnik nima imena DNS ali pa ga ne poznate, vnesite kar naslov IP strežnika). Priporočamo, da namestite Apache *za vse uporabnike, na vrata 80, kot storitev* (service). Opomba: če že imate nameščen IIS ali kakšen drug program, ki posluša port 80, je lahko namestitev neuspešna. Če se to zgodi, pojdite v namestitveno mapo strežnika Apache, običajno `c:\Program Files\Apache Group\Apache2\conf` in poiščite datoteko `httpd.conf`. Spremenite jo tako, da vrednost `vrat` v izrazu `Listen 80` zamenjate z vrednostjo katerihkoli prostih vrat, na primer `Listen 81`. Potem namestitev ponovite - tokrat bi morala biti uspešna.
3. Sedaj preverite, če spletni strežnik Apache deluje pravilno. Usmerite svoj spletni brskalnik na stran `http://localhost/` - pojaviti se mora prednastavljena spletna stran.



Opozorilo

Če se odločite namestiti strežnik Apache kot storitev (service), upoštevajte, da se bo po privzetih nastavitvah pognal s krajevnim sistemskim računom (local system account). Bolj varen način je, da za poganjanje strežnika Apache ustvarite poseben račun.

Zagotovite, da ima račun, pod katerim teče strežnik Apache, v mapi, kjer se nahajajo skladišča, eksplicitno določene vse pravice (desni klik na mapo | lastnosti | varnost). V nasprotnem primeru uporabniki ne bodo mogli objavljati svojih sprememb.

Tudi če Apache teče pod sistemskim računom, morate pravico vseeno določiti (v tem primeru za račun SYSTEM).

Če Apache tega dovoljenja nima, bodo uporabniki dobili opozorilo "Dostop zavržen", ki je v dnevniku napak strežnika Apache predstavljen s številko 500.

3.7.3. Namestitev sistema Subversion

1. Prenesite zadnjo različico Win32 dvojiških datotek Subversion za Apache. Prepričajte se, da ste prenesli pravilno različico, drugače boste ob ponovnem zagonu dobili čudna sporočila o napakah. Če uporabljate Apache 2.2.x, uporabite povezavo <http://subversion.tigris.org/servlets/ProjectDocumentList?folderID=8100>.
2. Poženite namestitveni program za Subversion in sledite navodilom. Če namestitveni program ugotovi, da imate nameščen strežnik Apache, ste že skoraj pri koncu. Če strežnika ne najde, potem vas čaka še nekaj korakov.
- 3.

Z Raziskovalcem pojdite v namestitveno mapo Subversion (običajno `c:\program files\Subversion`) in poiščite datoteki `/httpd/mod_dav_svn.so` in `mod_authz_svn.so`. Prekopirajte ti datoteki v podmapo `modules` namestitvene mape strežnika Apache (običajno `c:\program files\apache group\apache2\modules`).

4. Prekopirajte datoteki `/bin/libdb*.dll` in `/bin/intl3_svn.dll` iz namestitvene mape Subversion v podmapo `bin` strežnika Apache.
5. Popravite konfiguracijsko datoteko strežnika Apache (ponavadi `C:\Program Files\Apache Group\Apache2\conf\httpd.conf`) z urejevalnikom besedil (n. pr. z Beležko) in naredite naslednje spremembe:

Odstranite komentar (oznako '#') izpred naslednjih vrstic:

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

Dodajte naslednji vrstici na konec sekcije `LoadModule`.

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

3.7.4. Konfiguracija

Apache in Subversion sta nameščena, vendar Apache še ne zna delati z odjemalci Subversion, kakršen je TortoiseSVN. Da poveste strežniku Apache, kateri naslovi URL bodo uporabljeni kot skladišča Subversion, morate popraviti njegovo konfiguracijsko datoteko (običajno se nahaja na `c:\program files\apache group\apache2\conf\httpd.conf`) z urejevalnikom besedila (n. pr. z Beležko):

1. Na koncu datoteke `Config` dodajte naslednje vrstice:

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

S tem nastavite strežnik Apache tako, da so vsa skladišča sistema Subversion znotraj mape `D:\SVN`. Zunanjemu svetu so skladišča na voljo na naslovu `http://MojStrežnik/svn/`. Dostop je omejen na uporabniška imena/gesla, ki so navedena v datoteki `passwd`.

2. Za ustvarjanje datoteke `passwd` ponovno odprite ukazno lupino (okno DOS), pojdite v mapo strežnika Apache2 (ponavadi `c:\program files\apache group\apache2`) in z naslednjim ukazom ustvarite datoteko:

```
bin\htpasswd -c passwd <username>
```

S tem ustvarite datoteko z imenom `passwd`, ki se uporablja za avtentikacijo. Dodatne uporabnike lahko vnesete z


```
bin\htpasswd geslo <uporabniško ime>
```

3. Ponovno zaženite storitev (service) Apache.
4. Usmerite spletni brskalnik na `http://MojStrežnik/svn/MojeNovoSkladišče` (kjer je `MojeNovoSkladišče` ime skladišča Subversion, ki je ste ga predhodno ustvarili). Če je šlo vse po načrtih, potem vas mora pričakati vnosno polje za uporabniško ime in geslo. Po vpisu lahko vidite vsebino skladišča.

Kratka obrazložitev pravkar vnesenega:

Nastavitev	Razlaga
<Location /svn>	pomeni, da so skladišča sistema Subversion na voljo na naslovu URL <code>http://MojStrežnik/svn/</code>
DAV svn	pove strežniku Apache, kateri modul bo odgovoren za strežbo naslova URL - v tem primeru modul Subversion.
SVNListParentPath on	V različicah sistema Subversion 1.3 ali novejših ta ukaz omogoča izpis seznama vse razpoložljivih skladišč v mapi <code>SVNParentPath</code> .
SVNParentPath D:\SVN	pove sistemu Subversion, naj skladišča išče v mapi <code>D:\SVN</code>
SVNIndexXSLT svnindex.xml"	Nastavitev za lepši izpis pri brskanju po skladiščih.
AuthType Basic	se uporablja za nastavitev osnovne avtentikacije, n. pr. uporabniško ime/geslo
AuthName "Subversion repositories"	se uporablja kot informacija ob prikazu pojavnega okna za avtentikacijo. Uporabniku pove, za kaj se avtentikacija uporablja.
AuthUserFile passwd	določi datoteko z gesli, ki naj se uporabi za avtentikacijo
AuthzSVNAccessFile	lokacija datoteke dostopov za poti v skladiščih Subversion
Require valid-user	določi, da lahko do naslova URL dostopajo le uporabniki, ki so navedli pravilno kombinacijo uporabniškega imena in gesla

Tabela 3.1. Nastavitve Apache `httpd.conf`

Vendar je to le primer. Obstaja še zelo veliko stvari, ki jih lahko naredite s spletnim strežnikom Apache.

- Če želite, da imajo pravico branja v vašem skladišču vsi uporabniki, pravico pisanja pa samo nekateri, spremenite vrstico

```
Require valid-user
```

```
v
```

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

- Z uporabno datoteko `passwd` lahko omejite oziroma dovolite dostop do vseh skladišč naenkrat. Če želite imeti več nadzora nad tem, kateri uporabnik ima pravice v določeni mapi skladišča, odkomentirajte vrstico

```
#AuthzSVNAccessFile svnaccessfile
```

in ustvarite datoteko dostopa. Apache poskrbi, da lahko samo veljavni uporabniki dostopajo do lokacije /svn, in potem posreduje uporabniško ime modula AuthzSVNAccessFile, tako da lahko le-ta bolj podrobno ugotovi pravice glede na pravila, ki so zapisana v datoteki dostopa. Poti so zapisane v obliki skladišče:pot ali preprosto pot. Če ne določite skladišča, pravilo velja za vsa skladišča v mapi SVNParentPath. Oblika avtorizacijske datoteke, ki jo uporablja mod_authz_svn, je opisana v [Razdelek 3.7.6, "Avtorizacija na osnovi poti"](#)

- Da bi polepšali izpis pri brskanju po skladišču, odkomentirajte naslednjo vrstico

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

in dodajte datoteke svnindex.xsl, svnindex.css in menucheckout.ico v korensko mapo dokumentov na strežniku (ponavadi C:/Program Files/Apache Group/Apache2/htdocs). Mapa se nastavi z ukazom DocumentRoot v konfiguracijski datoteki strežnika Apache.

You can get those three files directly from our source repository at <http://tortoissvn.googlecode.com/svn/trunk/contrib/svnindex>. ([Razdelek 3, "TortoiseSVN je brezplačen!"](#) explains how to access the TortoiseSVN source repository).

Datoteka XSL skladišča TortoiseSVN ima koristen dodatek: če uporabljate za brskanje po skladišču spletni brskalnik, se ob vsaki mapi v skladišču izriše ikona na desni strani imena. Če kliknete nanjo, se prikaže pogovorno okno za prevzem za ta naslov URL.

3.7.5. Več skladišč

Če ste pri nameščanju nastavili ukaz SVNParentPath, potem vam ni treba spreminjati konfiguracijske datoteke Apache vsakič, ko dodate novo skladišče. Enostavno ustvarite novo skladišče na isti lokaciji, na kateri ste ustvarili prvo skladišče! V našem podjetju imamo neposreden dostop do te lokacije na strežniku preko SMB (običajen datotečni dostop sistema Windows). Tako enostavno na tej lokaciji ustvarimo novo mapo, uporabimo ukaz TortoiseSVN → Tukaj ustvari skladišče... in nov projekt ima svoj dom...

Če uporabljate Subversion 1.3 ali novejšo različico, lahko uporabite ukaz SVNListParentPath on. S tem dovolite strežniku Apache, da prikaže seznam vseh projektov, če v brskalnik vpišete korensko pot namesto določenega skladišča.

3.7.6. Avtorizacija na osnovi poti

Modul mod_authz_svn omogoča podrobno nastavitvev pravic dostopa na osnovi uporabniških imen in poti skladišč. Možnost je na voljo pri uporabi strežnika Apache, od različice Subversion 1.3 naprej pa tudi pri uporabi svnsrve.

Primer datoteke izgleda takole:

```
[groups]
skrbniki = john, kate
razvijalci1 = john, rachel, sally
razvijalci2 = kate, peter, mark
dokumentaristi = bob, jane, mike
tecajniki = zak
# Privzeto pravilo za dostope za VSA skladišča
# Vsi lahko berejo, skrbniki imajo pravice za pisanje, # Dan German nima nobenih p
[/]
* = r
@skrbniki = rw
```

```

dangerman =
# Dovolji razvijalcem popoln dostop do projektnih skladišč
[proj1:/]
@razvijalci1 = rw
[proj2:/]
@razvijalci2 = rw
[bigproj:/]
@razvijalci1 = rw
@razvijalci2 = rw
trevor = rw
# Dodeli dokumentaristom pravice pisanja v vseh doc mapah
[/trunk/doc]
@dokumentaristi = rw
# Dodeli tečajnikom pravice pisanja le v skladišču TrainingRepos
[TrainingRepos:/]
@tečajniki = rw

```

Upoštevajte, da je preverjanje vsake poti lahko zelo potratna operacija, predvsem v primeru dnevnika. Strežnik preverja vsako spremenjeno pot v vsaki reviziji in preveri berljivost, kar je lahko zelo zamudno pri revizijah, kjer je spremenjeno veliko število datotek.

Avtentikacija in avtorizacija sta dva ločena procesa. Če želi uporabnik dobiti dostop do poti v skladišču, mora zadostiti *obema* pogojema: običajni zahtevi po avtentikaciji in avtorizacijski zahtevi za dostop do datoteke.

3.7.7. Avtentikacija z domeno Windows

Kot se verjetno opazili, morate narediti vnos uporabniškega imena in gesla v datoteko `passwd` za vsakega uporabnika posebej. In če želite, da vaši uporabniki (iz varnostnih razlogov) periodično spreminjajo geslo, morate te spremembe vnašati ročno.

Vendar obstaja rešitev problema - v primeru, če dostopate do skladišča iz krajevnega omrežja z domenskim kontrolerjem sistema Windows: `mod_auth_sspi!`

Originalni modul SSPI, vključno z izvorno kodo, je včasih ponujajo podjetje Syneapps. Vendar je razvoj že nekaj časa ustavljen. A ne obupajte, skupnost nadaljuje s projektom in ga izboljšuje. Nova lokacija projekta je [SourceForge](http://sourceforge.net/projects/mod-auth-sspi/) [http://sourceforge.net/projects/mod-auth-sspi/].

- Prenesite modul, ki ustreza različni vašega strežnika Apache, prekopirajte datoteko `mod_auth_sspi.so` v podmapo `modules` namestitvene mape strežnika Apache.
- Popravite konfiguracijsko datoteko strežnika Apache: dodajte vrstico

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

v sekcijo `LoadModule`. To vrstico morate vstaviti *pred* vrstico

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

- Če želite, da Subversion uporablja ta tip avtentikacije, morate spremeniti vrstico

```
AuthType Basic
```

v

```
AuthType SSPI
```

```
in dodati
```

```
SSPIAuth On
SSPIAuthoritative On
SSPIDomain <domaincontroller>
SSPIOmitDomain on
SSPIUsernameCase lower
SSPIPerRequestAuth on
SSPIOfferBasic On
```

v blok `<Location /svn>`. Če nimate domenskega kontrolerja, pustite ime domenske kontrole na `<domaincontroller>`.

Če za avtentikacijo uporabljate SSPI, potem ne potrebujete več vrstice `AuthUserFile` za datoteko gesel. Apache avtentificira vaše uporabniško ime in geslo v domeni Windows. Morali pa boste posodobiti seznam uporabnikov v datoteki `svnaccessfile`, da bo le-ta vsebovala tudi uporabnike oblike `DOMENA\uporabniško ime`.



Pomembno

Avtentikacija SSPI je na voljo le v varne (SSL) povezave (https). Če za dostop do strežnika uporabljate navadno (http) povezavo, stvar ne bo delovala.

Za vklop varne povezave (SSL) na strežniku preberite [Razdelek 3.7.9, "Zavarovanje strežnika z SSL"](#)



Namig

Datoteke `AuthzSVNAccessFile` sistema Subversion so občutljive na velike in male črke pri navajanju uporabniških imen (beseda `UPorabnik` je torej drugačna kot `uporabnik`).

V Microsoftovem svetu domene Windows in uporabniška imena ne ločijo velikih in malih črk. Kljub vsemu nekateri skrbniki sistema radi uporabljajo kamelji stil za uporabniške račune (n.pr. `JUporabnik`).

Ta razlika pa vam lahko ponagaja pri uporabi avtentikacije SSPI. Sistem Subversion dobi domeno Windows in uporabniško ime v natančno takšni obliki, kot ju je uporabnik vpisal. Internet Explorer pogosto pošlje uporabniško ime strežniku Apache v obliki, ki je bila uporabljena pri ustvarjanju računa.

Rezultat tega je, da boste mogoče potrebovali vsaj dva vnosa v `AuthzSVNAccessFile` za vsakega uporabnika -- vnos z malimi črkami in vnos v enakem formatu, kot ga Internet Explorer pošlje strežniku Apache. Prav tako boste morali naučiti uporabnike, da vnašajo avtentikacijske podatke z malimi črkami, ko dostopajo do skladišč preko programa TortoiseSVN.

Zapis dostopov in napak strežnika Apache sta vaša najboljša prijatelja, ko je potrebno reševati takšne težave. Pomagata vam ugotoviti natančen niz, ki ga dobi modul `AuthzSVNAccessFile`. Mogoče boste morali poskušati ugotoviti pravilno obliko niza v datoteki `svnaccessfile` (n. pr. `DOMENA\uporabnik` oz. `DOMEMA//uporabnik`), preden bo sistem res deloval.

3.7.8. Avtentikacija iz več virov

Določiti je mogoče več virov za avtentikacijo skladišč Subversion. Da bi to delovalo, morate vsako avtentikacijo narediti neavtoritativno. S tem bo Apache pri več virih preveril, ali je sta uporabniško ime in geslo pravilna.

Pogosta je situacija, ko imamo avtentikacijo tako preko domene Windows kot preko datoteke `passwd`. Na ta način lahko omogočite dostop tudi uporabnikom, ki nimajo uporabniškega imena in gesla za domeno Windows.

- Da bi omogočili avtentikacijo preko domene Windows in datoteke `passwd`, dodajte naslednji vrstici znotraj bloka `<Location>` v konfiguracijski datoteki strežnika Apache:

```
AuthBasicAuthoritative Off
SSPIAuthoritative Off
```

Sledi celoten primer konfiguracije strežnika Apache za kombinacijo avtentikacije preko domene Windows in datoteke `passwd`:

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

  AuthName "Skladišča Subversion"
  AuthzSVNAccessFile svnaccessfile.txt

# Logiranje preko domen NT.
  AuthType SSPI
  SSPIAuth On
  SSPIAuthoritative Off
  SSPIDomain <domaincontroller>
  SSPIOfferBasic On

# Logiranje preko datoteke passwd.
  AuthType Basic
  AuthBasicAuthoritative Off
  AuthUserFile passwd

  Require valid-user
</Location>
```

3.7.9. Zavarovanje strežnika z SSL

Čeprav ima Apache 2.2.x podporo za OpenSSL, le-ta ob privzetih nastavitvah ni vključena. Vključiti jo morate ročno.

1. V konfiguracijski datoteki strežnika Apache odstranite komentar izpred naslednjih vrstic:

```
#LoadModule ssl_module modules/mod_ssl.so
```

in na koncu

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

nato spremenite vrstico (v eni vrstici)

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

v

```
SSLMutex default
```

2. Nato morate ustvariti certifikat SSL. Odprite ukazni poziv (okno DOS) in pojdite v namestitveno mapo strežnika Apache (n. pr. C:\program files\apache group\apache2) in vpišite naslednji ukaz:

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

Vpisati boste morali šifrirno frazo. Ne uporabljajte enostavnih besed, ampak vpišite cel stavek, naprimer del pesmi. Daljša je fraza, boljša je. Vnesti morate še naslov URL vašega strežnika. Na ostala vprašanja vam ni nujno odgovoriti, vendar vam priporočamo, da vseeno odgovorite.

Običajno se datoteka `privkey.pem` ustvari samodejno, če pa se ne, morate vtipkati naslednji ukaz, da jo ustvarite:

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

Nato vpišite naslednje ukaze

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

in (v eni vrstici)

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

in nato (v eni vrstici)

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

S tem ustvarite certifikat, ki bo potekel čez 4000 dni. Nato vpišite (v eni vrstici) še:

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

Ti ukazi ustvarijo nekaj datotek v konfiguracijski mapi (`conf`) strežnika Apache (`server.der.crt`, `server.csr`, `server.key`, `.rnd`, `privkey.pem`, `server.cert`).

3. Ponovno zaženite storitev Apache.
4. Usmerite vaš spletni brskalnik na `https://servername/svn/project ...`



SSL in Internet Explorer

Če dostop do strežnika zavarujete s protokolom SSL in uporabite avtentikacijo z domeno Windows, boste ugotovili, da brskanje po skladiščih z Internet Explorerjem ne deluje več. Ne skrbite - samo Internet Explorer ni sposoben izvesti avtentikacije. Ostali brskalniki tega

problema nimajo, TortoiseSVN in ostali odjemalci za Subversion še vedno lahko izvajajo avtentikacijo.

Če še vedno želite uporabljati Internet Explorer za brskanje po skladišču, lahko:

- v konfiguracijski datoteki Apache definirate ločen ukaz `<Location /path>` in dodate vrstico `SSPIBasicPreferred On`. To omogoči avtentikacijo preko Internet Explorerja, vendar pa ostali brskalniki in odjemalci za Subversion ne bodo mogli izvajati avtentikacije na tej lokaciji.
- Ponudite tudi brskanje preko neenkriptirane avtentikacije (brez SSL). Začuda IE nima nobenih težav z avtentikacijo, če povezava ni zaščitena z SSL.
- Pri "standardni" namestitvi SSL se v Apachejevem virtualnem gostitelju pogosto nahaja naslednji stavek:

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

Obstajajo (so obstajali?) dobri razlogi za takšno konfiguracijo, poglejte na http://www.modssl.org/docs/2.8/ssl_faq.html#ToC49 Toda če želite uporabiti avtentikacijo NTLM, morate uporabiti opcijo `keepalive`. Če odkomentirate celoten blok `SetEnvIf`, boste lahko uporabljali avtentikacijo IE z avtentikacijo sistema Windows preko protokola SSL in strežnika Apache na sistemu WIN32 z vključenim modulom `mod_auth_sspi`.



Vsiljevanje dostopa SSL

Če ste nastavili protokol SSL, da bi bili dostopi do skladišč bolj varni, onemogočite navaden - ne-SSL (http) dostop in dovolite le dostop https. To naredite tako, da dodate nov ukaz v Subversion blok `<Location>`: `SSLRequireSSL`.

Primer bloka `<Location>` izgleda takole:

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

3.7.10. Uporaba certifikatov odjemalca z navideznimi gostitelji SSL

Prispevek je na dopisni seznam poslal Nigel Green. Hvala!

V določenih primerih morate na enem strežniku ustvariti dva navidezna gostitelja SSL: prvega za javni dostop preko interneta brez zahteve po certifikatu, drugega za varen dostop s certifikatom, ki poganja strežnik Subversion.

Če odseku *per-server* v konfiguraciji strežnika Apache (n. pr. izven blokov `VirtualHost` in `Directory`) dodate ukaz `SSLVerifyClient Optional`, strežnik prisilite, da ob prvem kontaktu

v načinu SSL zahteva certifikat. Zaradi napake v modulu `mod_ssl` je pomembno certifikat zatevati na tem mestu, kajti stvar ob ponovnem vzpostavljanju poveze SSL ne deluje.

Zagato rešimo tako, da dodamo naslednji ukaz navideznemu gostitelju, ki ga želimo uporabljati za Subversion:

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

S tem dostop do mape dovolimo le uporabnikom, ki so uporabili veljaven certifikat.

Za zaključek pogledjmo pomembne dele konfiguracijske datoteke strežnika Apache:

```
SSLVerifyClient Optional

### Navidezni gostitelj za JAVEN dostop
### (ne zahteva certifikata)

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

### Navidezni gostitelj za SUBVERSION
### (zahteva certifikat)
<VirtualHost 127.0.0.1:443>
  <Directory "korenska mapa za subversion">
    SSLRequire %{SSL_CLIENT_VERIFY} eq "SUCCESS"
  </Directory>

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

Poglavje 4. Dnevna uporaba

Ta dokument opisuje dnevno uporabo odjemalca TortoiseSVN. To *ni* uvod v sisteme za nadzor različic, prav tako pa *ni* uvod v Subversion (SVN). To poglavje uporabite, ko veste, kaj želite narediti, vendar se ne spomnite natančno, kako se to naredi.

Če potrebujete uvod v nadzor različic s sistemom Subversion, vam priporočamo odlično knjigo: *Version Control with Subversion* [<http://svnbook.red-bean.com/>].

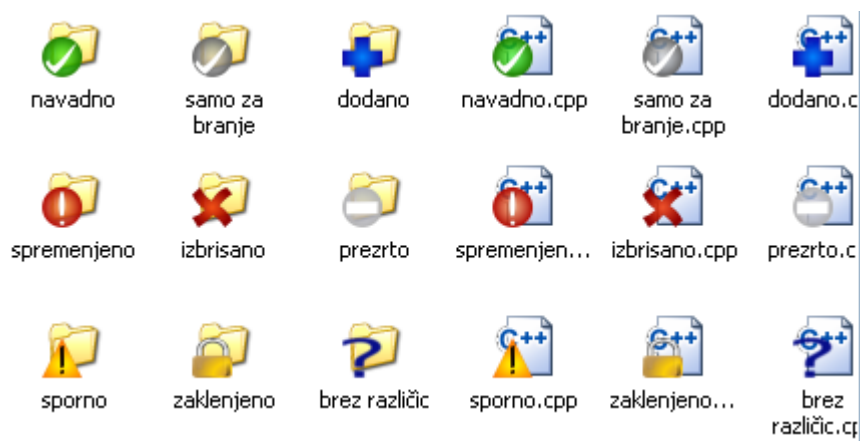
Tako kot programa TortoiseSVN in Subversion je tudi ta dokument "v delu". Če najdete kakšne napake, jih sporočite na dopisni seznam, da bomo lahko dokumentacijo dopolnili. Nekateri posnetki zaslonov v Dnevni uporabi so lahko drugačni, kot so trenutno v aplikaciji. Prosimo, oprostite. Aplikacijo TortoiseSVN razvijamo v svojem prostem času.

Da bi se čimveč naučili iz poglavja Dnevna uporaba:

- morate imeti TortoiseSVN že nameščen,
- morate poznati sisteme za nadzor različic,
- morate poznati osnove sistema Subversion,
- imeti morate nameščen strežnik in/ali dostop do skladišča Subversion.

4.1. Kako začeti

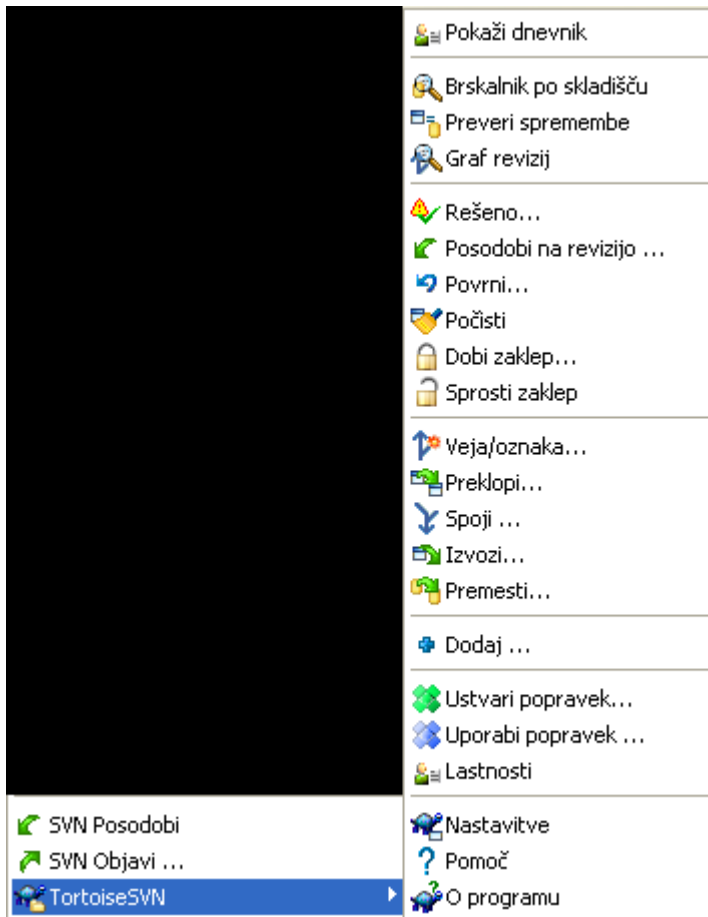
4.1.1. Prekrivne ikone



Slika 4.1. Raziskovalec prikaže prekrivne ikone

Ena izmed vizuelno najbolj opaznih zmožnosti TortoiseSVN so prekrivne ikone, ki se pojavijo na datotekah v delovni kopiji. Z njihovo pomočjo že na prvi pogled vidite, katere datoteke so spremenjene. Preberite [Razdelek 4.7.1, "Prekrivne ikone"](#) za več informacij o tem, kaj različne ikone predstavljajo.

4.1.2. Kontekstni meni



Slika 4.2. Kontekstni meni za mapo pod nadzorom različic

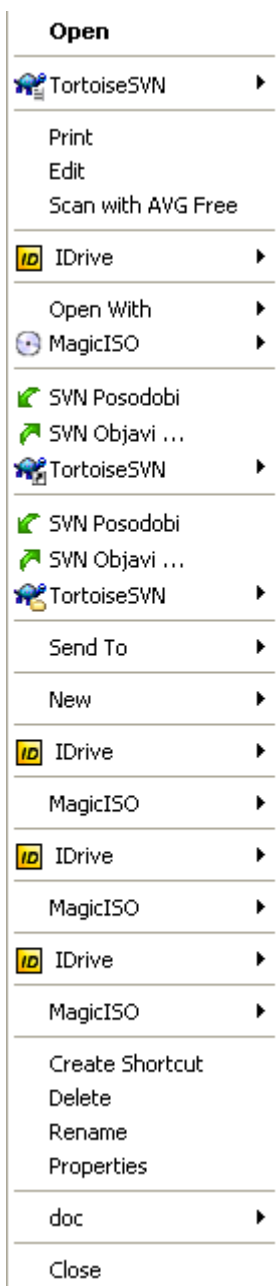
Vsi ukazi TortoiseSVN se kličejo iz kontekstnega menija Raziskovalca. Večina je vidna, ko desno kliknete na datoteko ali mapo. Ukazi, ki so na voljo, so odvisni od tega, ali je datoteka ali mapa oziroma njihova nadrejena mapa pod nadzorom različic ali ne. Meni TortoiseSVN lahko vidite tudi kot del glavnega menija Datoteka v Raziskovalcu.



Namig

Nekateri zelo redko uporabljeni ukazi so na voljo v razširjenem kontekstnem meniju, ki ga prikažete tako, da držite pritisnjeno tipko **Shift** in desno kliknete.

V nekaterih primerih lahko vidite več elementov TortoiseSVN. To ni napaka!

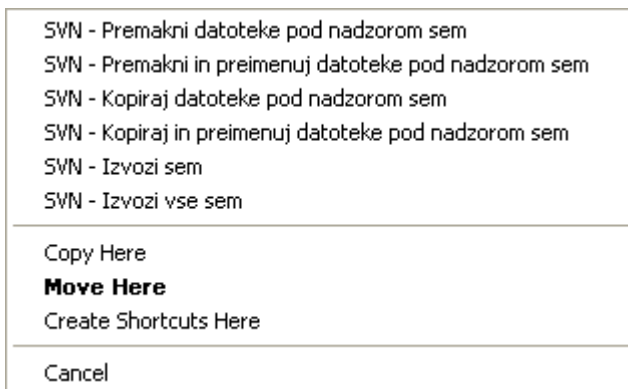


Slika 4.3. Kontekstni meni v Raziskovalcu za bližnjico v mapi pod nadzorom različic

Ta primer opisuje bližnjico za datoteke brez različic. Znotraj mape pod nadzorom in v meniju Raziskovalca so *trije* elementi sistema TortoiseSVN. Eden je za mapo, eden za bližnjico samo in tretji za objekt, na katerega bližnjica kaže. Da bi lažje ločevali med njimi, imajo ikone v spodnjem desnem kotu indikator, ki pove, ali se element menija nanaša na datoteko, mapo, bližnico ali na več elementov.

Če uporabljate operacijski sistem Windows 2000, boste opazili, da so v kontekstnem meniju vnosi prikazani le tekstovno, brez ikon, ki jih vidite na zgornji sliki. Zavedamo se, da je prikaz ikon v prejšnjih različicah deloval. Microsoft je spremenil način dela z ikonami v operacijskem sistemu Windows Vista, zato smo morali uporabiti drugačen način prikaza ikon, ki pa na žalost ne deluje v operacijskem sistemu Windows 2000.

4.1.3. Povleci in spusti



Slika 4.4. Meni ob premikanju mape, ki je pod nadzorom različic

Drugi ukazi so na voljo kot meni, ko z desnim gumbom povlečete datoteke ali mape na novo lokacijo znotraj delovne kopije ali ko z desnim gumbom povlečete datoteko ali mapo brez različic v mapo, ki je pod nadzorom.

4.1.4. Pogoste bližnjice

Nekatere pogoste operacije imajo znane bližnjice sistema Windows, vendar se ne pojavljajo na gumbih ali v menijih. Če vam ne uspe ugotoviti, kako se naredi kaj preprostega, n. pr. kako se osveži pogled, pogledajte sem.

F1
Pomoč, seveda.

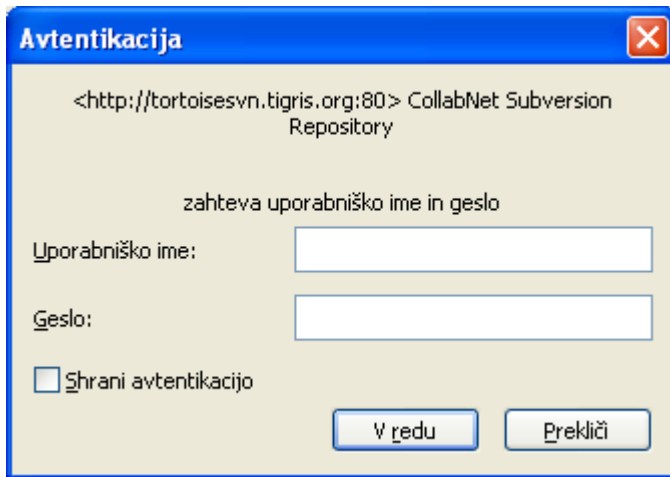
F5
Osveži trenutni pogled. To je verjetno najbolj uporaben ukaz, ki ga lahko izvršite z eno tipko. Na primer: v Raziskovalcu boste s tem osvežili prekrivne ikone v delovni kopiji. V oknu za objave se bo s tem ponovno pregledala delovna kopija, v okno pa se bodo dodale najdene spremembe. V oknu dnevniških zapisov bo to povzročilo ponovno kontaktiranje skladišča in preverjanje, če obstajajo novejši zapisi.

Ctrl-A
Izberi vse. To lahko uporabite, če se pojavi sporočilo o napaki in ga želite prekopirati in prilepiti v elektronsko pošto. Uporabite Ctrl-A, če želite izbrati sporočilo o napaki in potem...

Ctrl-C
... Kopirajte izbrano besedilo

4.1.5. Avtentikacija

Če je skladišče, do katerega želite dostopati, zaščiteno z geslom, se prikaže avtentikacijsko pogovorno okno.



Slika 4.5. Okno za avtentikacijo

Vpišite svoje uporabniško ime in geslo. Potrditveno polje pove programu TortoiseSVN, naj avtentikacijske podatke shrani v privzeto mapo: %APPDATA%\Subversion\auth v tri podmape:

- `svn.simple` vsebuje podatke o osnovni avtentikaciji (uporabniško ime/geslo).
- `svn.ssl.server` vsebuje certificate strežnika SSL
- `svn.username` vsebuje avtentikacijske podatke za avtentikacijo samo z uporabniškim imenom (brez gesla).

Če želite izbrisati avtentikacijski predpomnilnik za vse strežnike, lahko to storite v nastavitvah TortoiseSVN - stran **Splošno**. Ta gumb bo izbrisal vse avtentikacijske podatke iz map `auth`, prav tako pa tudi vse podatke iz registra (iz prejšnjih različic TortoiseSVN). Preberite [Razdelek 4.30.6, "Shranjeni podatki"](#).

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 [windows-help-central.com](http://www.windows-help-central.com/windows-shutdown-script.html) [http://www.windows-help-central.com/windows-shutdown-script.html].

Za več informacij, kako namestiti strežnik za avtentikacijo in nadzor dostopov, si oglejte [Razdelek 3.5, "Accessing the Repository"](#)

4.1.6. Povečevanje oken

Kar nekaj pogovornih oken TortoiseSVN prikaže veliko informacij. Pogosto je uporabno, če povečate le širino ali le višino, namesto da povečate okno na celo stran. Z lažje prilagajanje je tu gumb **Razpni**. Uporabite srednji miškin gumb za podaljšanje navpične stranice na višino zaslona in desni miških gumb za podaljšanje vodoravne stranice na širino zaslona.

4.2. Uvažanje podatkov v skladišče

4.2.1. Uvoz

Če uvažate v obstoječo skladišče, ki že vsebuje projekte, potem je struktura map že določena. Če uvažate podatke v novo skladišče, potem je koristno vložiti nekaj dodatnega časa in razmisliti o strukturi. Preberite [Razdelek 3.1.5, "Postavitev skladišča"](#) za dodatne napotke.

To podpoglavje opisuje ukaz Uvozi, katerega namen je uvažanje strukture map v skladišče v eni potezi. Čeprav opravi svoje delo, pa ima nekaj pomanjkljivosti:

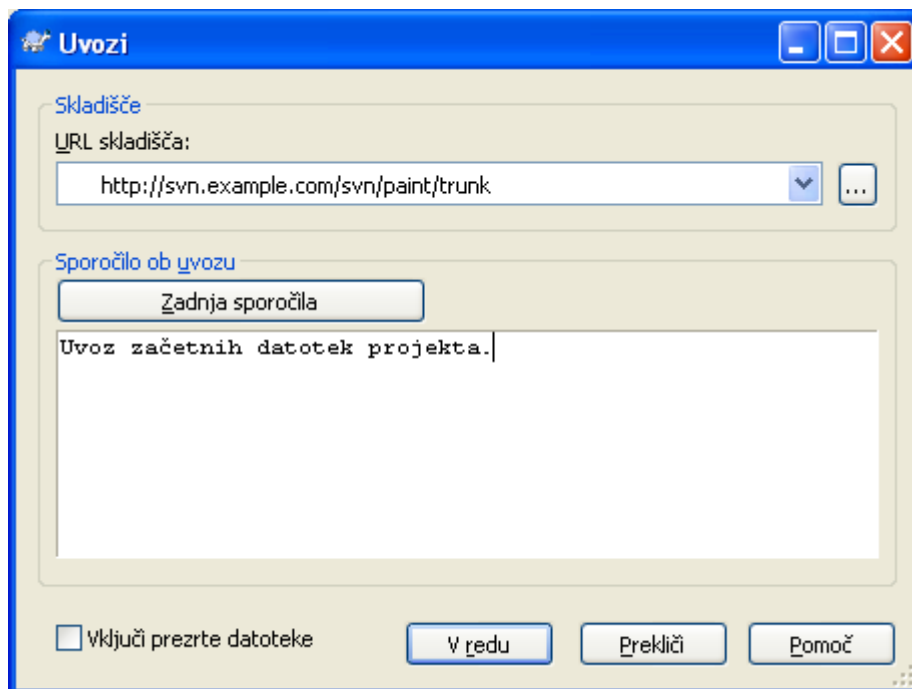
- Datotek in map, ki jih želite vključiti, ne morete izbrati, lahko pa si pomagata z nastavitvami splošnega vzorca prezrtih elementov.
- Uvožena mapa ne postane delovna kopija. Narediti morate prevzem datotek iz strežnika.
- Povsem preprosto je uvoziti v napačno mapo znotraj hierarhije skladišča.

Zato priporočamo, da tega ukaza ne uporabljate, ampak uporabite postopek v dveh korakih, opisan v [Razdelek 4.2.2, "Uvažanje na mestu"](#). Ker pa ste že ravno tukaj, si vseeno pogledjmo, kako uvažanje deluje...

Pred projekt uvozite v skladišče, morate:

1. Odstraniti vse datoteke, ki niso potrebne za gradnjo projekta (začasne datoteke, datoteke, ki jih generira prevajalnik, n. pr. *.obj, prevedene dvojiške datoteke...).
2. Organizirajte datoteke v mape in podmape. Čeprav je možno preimenovanje/premikanje tudi kasneje, je zelo priporočljivo, da postavite strukturo projekta pred uvažanjem.

Sedaj izberite vrhno datoteko vašega projekta v Raziskovalcu in desno kliknite. S tem odprete kontekstni meni. Izberite ukaz TortoiseSVN → Uvozi..., ki prikaže naslednje okno:



Slika 4.6. Okno za uvažanje

V tem pogovornem oknu vpišete naslov URL skladišča, kamor želite uvoziti svoj projekt. Zavedati se morate, da krajevna mapa, ki jo uvažate, ne postane del skladišča; v skladišče se uvozi le njena vsebina. Primer: če imate naslednjo strukturo map:

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

in uvozite C:\Projekti\Widget v `http://mojadomena.com/svn/trunk`, boste morda presenečeni ugotovili, da se vse vaše podmape nahajajo v podmapi `trunk` in ne v podmapi `Widget`. Podmapo morate navesti kot del naslova URL: `http://mojadomena.com/svn/trunk/Widget-X`. Uvažanje samodejno ustvari podmape znotraj skladišča, če le-te še ne obstajajo.

Sporočilo uvoza se uporabi kot sporočilo dnevniškega zapisa.

Po privzetih nastavitvah se datoteke in mape, ki ustrezajo splošnemu vzorcu prezrtih elementov, *ne* uvozijo. Če želite to pri posameznem uvozu spremeniti, potrdite potrditveno polje **Vključi prezrte datoteke**. Preberite **Razdelek 4.30.1, "Splošne nastavitve"** za več informacij o nastavljanju splošnih vzorcev prezrtih elementov.

Ko pritisnete gumb **V redu**, TortoiseSVN uvozi celotno strukturo map, vključno z vsemi datotekami, v skladišče. Projekt je sedaj shranjen v skladišču pod nadzorom različic. Upoštevajte, da mapa, ki ste jo uvozili, *NI* pod nadzorom različic! Da bi dobili *delovno kopijo* datotek pod nadzorom, morate narediti prevzem verzije, ki ste jo pravkar uvozili. Ali pa berite naprej, da izveste, kako uvoziti mapo na mestu.

4.2.2. Uvažanje na mestu

Ob predpostavki, da skladišče že imate in želite dodati vanj novo drevesno strukturo, naredite naslednje:

1. Uporabite brskalnik po skladišču, da ustvarite novo projektno mapo neposredno v skladišču.
2. Prezmite novo mapo na mapi, ki jo želite uvoziti. Dobili boste obvestilo, da krajevna mapa ni prazna. Sedaj imate vrhno mapo pod nadzorom različic, medtem ko njena vsebina ni pod nadzorom.
3. Uporabite TortoiseSVN → **Dodaj...** na mapi pod nadzorom. S tem lahko dodate delno ali celotno vsebino, dodate ali odstranite datoteke, nastavite lastnosti `svn:ignore` na mapah in naredite vse ostale spremembe, ki so potrebne.
4. Objavite vrhno mapo in dobili ste novo drevo pod nadzorom in krajevno delovno kopijo, ustvarjeno iz obstoječe mape.

4.2.3. Posebne datoteke

Včasih imate v sistemu različic datoteko, ki vsebuje podatke, specifične za uporabnika. To pomeni, da imate datoteko, ki jo mora vsak uporabnik urediti, da ustreza njegovi/njeni namestitvi. Imeti takšno datoteko pod nadzorom je težko, ker bi vsak uporabnik ob vsaki objavi objavil svoje spremembe te datoteke.

V takšnih primerih priporočamo uporabo *predlog*. Ustvarite datoteko, ki vsebuje vse podatke, ki jih razvijalci potrebujejo, jo dodajte v sistem nadzora različic in pustite, da jo razvijalci prevzamejo. Nato si vsak razvijalec *naredi kopijo* te datoteke in jo preimenuje. Po tem spreminjanje kopije ni več problematično.

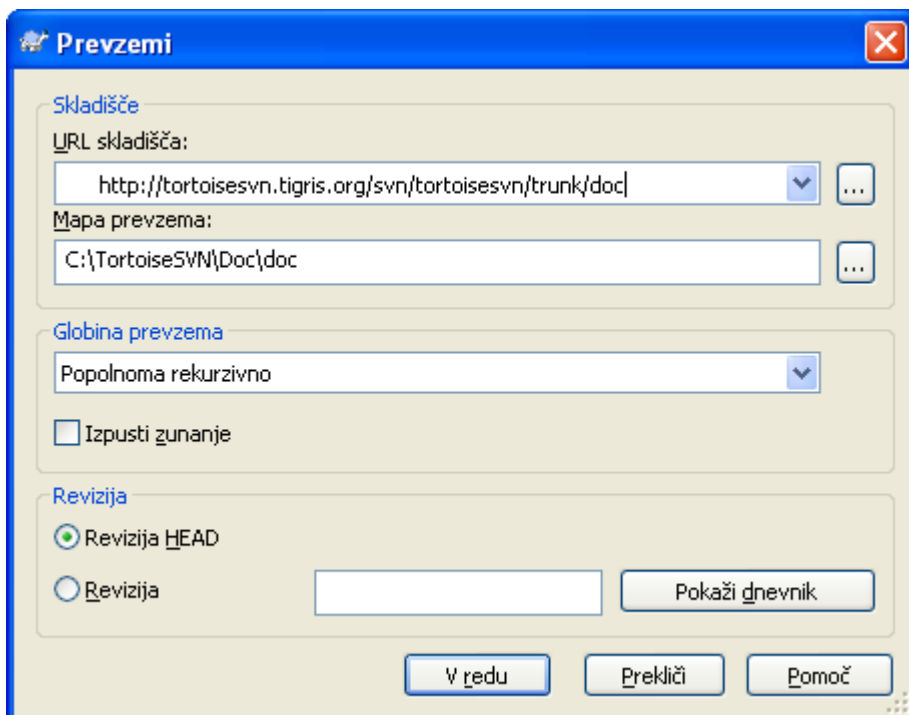
Za primer si pogledajte ukazno datoteko za gradnjo TortoiseSVN. Ta kliče datoteko `TortoiseVars.bat`, ki ne obstaja v skladišču. Obstaja le datoteka `TortoiseVars.tmpl`. `TortoiseVars.tmpl` je predloga datoteke. Vsak uporabnik si izdela svojo kopijo in jo preimenuje v `TortoiseVars.bat`. V datoteko smo dodali komentarje, da uporabniki vidijo, katere vrstice morajo spremeniti v skladu s svojo namestitvijo, da bo stvar delovala.

Da ne bi motili uporabnikov, smo dodali datoteko `TortoiseVars.bat` na listo prezrtih elementov v nadrejeni mapi, to pomeni, da smo nastavili lastnost `svn:ignore`, ki vsebuje to datoteko. Tako se ta datoteka ne bo pojavila v oknu za objave kot datoteka brez različic.

4.3. Prevzemanje delovne kopije

Da bi ustvarili delovno kopijo, morate narediti *prevzem* iz skladišča.

V Raziskovalcu izberite mapo, kjer želite imeti delovno kopijo. Z desnim klikom prikličite kontekstni meni in izberite ukaz TortoiseSVN → Prezemi..., ki ponudi naslednje pogovorno okno:



Slika 4.7. Okno za prevzem

Če vnesete ime mape, ki še ne obstaja, se le-ta pred prevzemom samodejno ustvari.

4.3.1. Globina prevzema

Lahko določite tudi *globino* prevzema. Ta označuje, kako globoko v mape deluje prevzem. Če želite prevzeti le nekatere mape velikega drevesa, prevzemite le vrhno mapo, nato pa rekurzivno posodobite posamezne izbrane mape.

Popolnoma rekurzivno

Prevzemi celotno drevo, vključno z mapami in podmapami.

Takojšnji nasledniki, vključno z mapami

Prevzemi določeno mapo vključno z datotekami in podmapami, vendar brez vsebine podmap.

Samo podrejene datoteke

Prevzemi izbrano mapo, vključno z vsemi datotekami, vendar brez podmap.

Samo ta element

Prevzemi le mapo. Ne prenašaj vsebine (datotok in podmap).

Delovna kopija

Ohrani globino, določeno v delovni kopiji. Ta možnost se ne uporablja v oknu za prevzem, ampak je privzeta nastavitve v vseh ostalih oknih, kjer je mogoče nastaviti globino.

Exclude

Used to reduce working copy depth after a folder has already been populated. This option is only available in the Update to revision dialog.

Če prevzamete delno delovno kopijo (za globino prevzema izberete karkoli razen popolnoma rekurzivno), lahko dodatne podmape prevzamete z uporabno brskalnika po skladišču ([Razdelek 4.24, "Brskalnik po skladišču"](#)) ali pogovornega okna "preveri spremembe" ([Razdelek 4.7.3, "Krajevno in oddaljeno stanje"](#)).

V brskalniku po skladišču desno kliknite na prevzeto mapo in izberite TortoiseSVN → Brskalnik po skladišču. Poiščite podmapo, ki jo želite dodati v delovno kopijo, nato pa uporabite Kontekstni meni → Posodobi element na revizijo. Ta meni je na voljo le če izberete element, ki se še ne nahaja v vaši delovni kopiji, obstaja pa njegova nadrejena mapa.

V pogovornem oknu za preverjanje sprememb najprej kliknite gumb Preveri skladišče. Pogovorno okno prikaže vse datoteke in mape, ki se nahajajo v skladišču, pa niso prevezete, kot dodane oddaljeno. Desno kliknite na mape, ki jih želite dodati v delovno kopijo, nato uporabite Kontekstni meni → Posodobi.

Ta zmožnost je uporabna, kadar želite prevzeti dele velikega drevesa, hkrati pa ohraniti udobje posodabljanja ene same delovne kopije. Denimo, da imate veliki drevo s podmapami Projekt01 do Projekt99, prevzeti pa želite le mape Projekt03, Projekt25 and Projekt76/Podprojekt. Naredite naslednje korake:

1. Prezemite nadrejeno mapo z globino "Samo ta element". S tem ste ustvarili prazno vrhno mapo.
2. Za prikaz vsebine skladišča izberite novo mapo in uporabite TortoiseSVN → Brskalnik po skladišču.
3. Desno kliknite na mapo Projekt03 in izberite Kontekstni meni → Posodobi element na revizijo.... Uporabite prevzete nastavitve in kliknite na gumb V redu. S tem ste prevzeli vsebino mape.

Isti postopek ponovite za mapo Projekt25.

4. Pojdite na Projekt76/Podprojekt in ponovite postopek. Mapa Projekt76 nima vsebine - izjema je le mapa Podprojekt, katere vsebina je prenešana. Subversion je ustvaril vmesne mapi brez prenašanja vsebine.



Changing working copy depth

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....



Uporaba starejšega strežnika

Strežniki pred različico 1.5 ne poznajo koncepta globine delovne kopije, zato teh zahtev ne znajo vedno učinkovito izvrševati. Ukaz bo še vedno deloval, vendar bo starejši strežnik verjetno poslal vse podatke, ki jih bo odjemalec nato sfiltriral in odstranil nepotrebno vsebino. To pa pomeni veliko mrežnega prometa. Če je le mogoče, nadgradite strežnik na različico 1.5.

Če projekt vsebuje reference na zunanje projekte, ki jih *ne* želite prevzeti istočasno, uporabite potrditveno polje Izpusti zunanje.



Pomembno

Če je možnost Izpusti zunanje potrjena ali če želite povečati globino, morate narediti posodobitev delovne kopije z uporabo ukaza TortoiseSVN → Posodobi na revizijo... namesto ukaza TortoiseSVN → Posodobi. Standardna posodobitev vključuje vse zunanje projekte in obdrži obstoječo globino.

Priporočamo, da prevzamete samo trunk del strukture map ali manj. Če uporabite vrhno mapo, se vam lahko zgodi, da boste imeli povsem poln trdi disk, saj boste prevzeli prav vsako kopijo drevesne strukture, vključno z vsemi vejami in oznakami vašega projekta!



Izvažanje

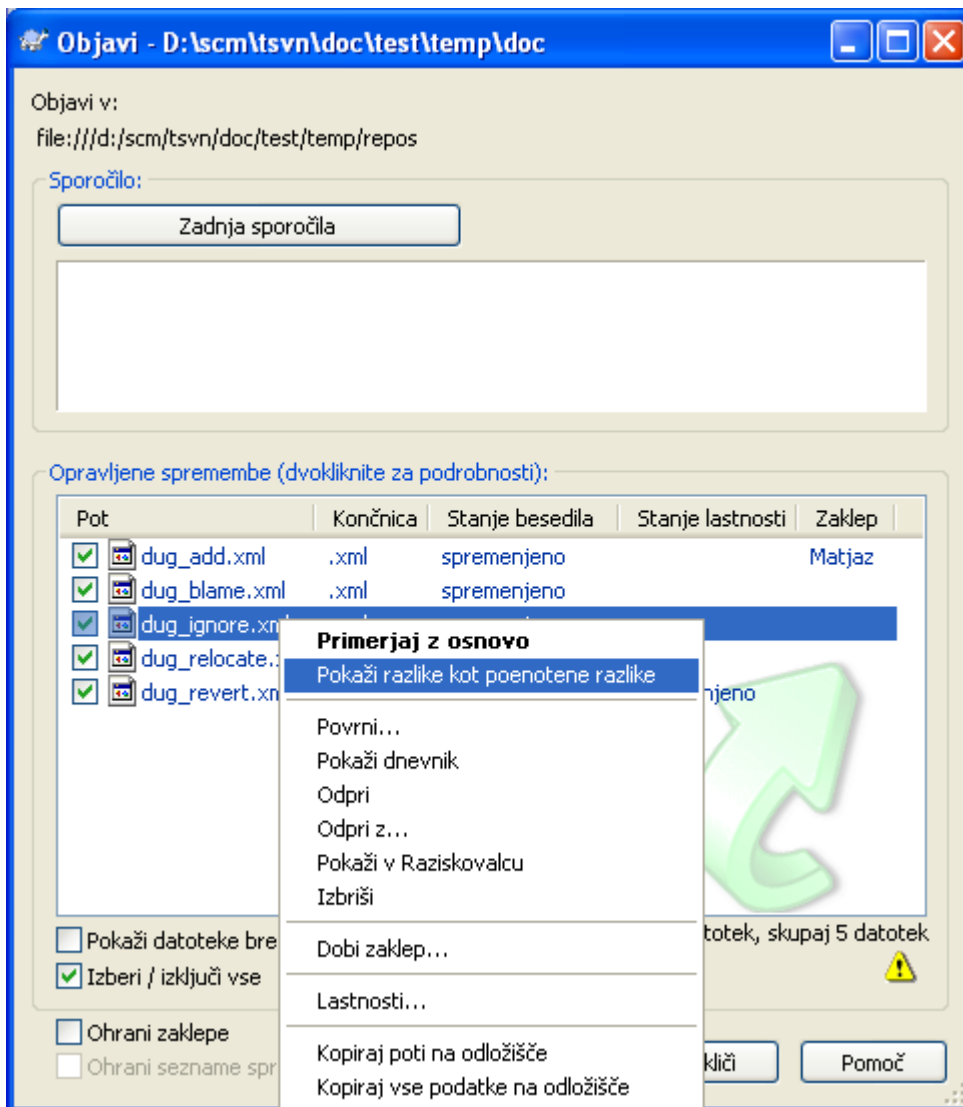
Včasih potrebujete krajevno kopijo brez map `.svn`, n. pr. če želite zapakirati svojo izvorno kodo. Več o tem si preberite v [Razdelek 4.26, "Izvažanje delovne kopije sistema Subversion"](#).

4.4. Objavljanje sprememb v skladišču

Pošiljanje sprememb, ki ste jih naredili na delovni kopiji, se imenuje *objava* sprememb. Preden pa spremembe objavite, se morate prepričati, da je vaša delovna kopija posodobljena. Uporabite lahko ukaz TortoiseSVN → Posodobi. Lahko pa uporabite ukaz TortoiseSVN → Preveri spremembe, da vidite, katere datoteke so bile spremenjene v delovni kopiji ali v skladišču.

4.4.1. Okno objave

Če je delovna kopija sodobna in brez sporov, ste pripravljeni, da spremembe objavite. Izberite datoteke in/ali mape, ki jih želite objaviti, in izberite TortoiseSVN → Objavi...



Slika 4.8. Okno objave

Okno za objave pokaže vsako spremenjeno datoteko, vključno z dodanimi, izbranimi in datotekami brez različic. Če ne želite, da se spremembe določene datoteke objavijo, izklopite potrditveno polje pred datoteko. Če želite vključiti datoteko brez različic, potrdite potrditveno polje - s tem jo dodate v objavo.

Elementi, ki so bili preklopljeni na drugo pot v skladišču, so označeni z znakom (s). Lahko da ste kakšen element preklopili, ko ste delali na neki veji in ga pozabili preklopiti nazaj. To je opozorilo!



Naj objavim datoteke ali mape?

Ko objavljate datoteke, pogovorno okno prikaže le datoteke, ki ste jih izbrali. Ko objavljate mapo, pogovorno okno izbere spremenjene datoteke samodejno. Če ste pozabili na novo datoteko, jo okno pri objavljanju mape vseeno najde. Objava mape *ne* pomeni, da se vse datoteke označijo kot spremenjene, ampak vam le olajša delo, saj ga opravi namesto vas.

Če ste spreminjali datoteke, ki so vključene iz drugega skladišča z uporabo lastnosti `svn:externals`, teh sprememb ne morete objaviti v isti atomični objavi. Če se to zgodi, se pod seznamom datotek pojavi opozorilni simbol, namig pa vam pojasni, da morajo biti zunanje datoteke objavljene posebej.



Veliko datotek brez različic v oknu objave

Če se vam zdi, da okno za objavo prikaže preveliko število datotek brez različic (generirane datoteke razvojnega okolja, varnostne datoteke), obstaja več načinov, da situacijo popravite. Lahko:

- dodate datoteko (ali končnico) na seznam izločenih datotek, kar storite v nastavitvah. To vpliva na vse delovne kopije, ki jih imate.
- dodate datoteko na seznam prezrtih (`svn:ignore`) z uporabo ukaza `TortoiseSVN → Dodaj na seznam prezrtih`. Ta nastavek vpliva le na mapo, v kateri ste nastavili lastnost `svn:ignore`. Z uporabo okna Lastnosti lahko lastnost mape spremenite.

Za več informacij preberite [Razdelek 4.13, "Dodajanje datotek in map na seznam prezrtih elementov"](#).

Dvoklik na spremenjeno datoteko v oknu objav požene zunanje orodje za razlikovanje in vam prikaže spremembe. Kontekstni meni vam ponuja še več možnosti, kar je razvidno iz posnetka zaslona. Poleg tega lahko iz okna potegnete in prenesete datoteke v ostale aplikacije, npr. v urejevalnik besedil ali razvojno okolje (IDE).

Izbiro elementov naredite s potrditvijo polja pred posameznim elementom. Za mape lahko uporabite označevanje s tipko **Shift**-izbira, s čemer naredite akcijo rekurzivno.

Stolpci v spodnjem delu okna so povsem prilagodljivi. Z desnim klikom na glavo kateregakoli stolpca priključete kontekstni meni, ki vam omogoča izbiro stolpcev, ki jih želite prikazati. Prav tako lahko spremenite širino stolpcev. To storite tako, da rob stolpca premaknete, ko je miška na meji med dvema stolpcema. Prilagoditve se ohranijo, tako da boste naslednjik videli enako postavitev.

Če imate nastavljene privzete nastavitve in naredite objavo sprememb, se vsi zaklepi v vaši lasti sprostijo takoj po uspešno izvedeni objavi. Če želite zaklepe ohraniti, potrdite polje **Ohrani zaklepe**. Privzeto stanje tega polja je definirano v konfiguracijski datoteki sistema Subversion z možnostjo `no_unlock`. Za več informacij o urejanju konfiguracijske datoteke Subversion si preberite [Razdelek 4.30.1, "Splošne nastavitve"](#).



Povleci in spusti

Datoteke lahko povlečete in spustite v okno za objavo od drugod pod pogojem, da ste delovne kopije prevzeli iz istega skladišča. Primer: imate ogromno delovno kopijo in

uporabljate več Raziskovalcev, ki ima nastavljene različne trenutne mape znotraj delovne kopije. Če se želite izogniti objavljanju vrhnje mape (kar bi pomenilo dolgo iskanje spremenjenih datotek), lahko okno za objave odprete v neki mapi, nato pa iz različnih Raziskovalcev povlečete v okno datoteke, ki jih želite istočasno objaviti.

Datoteke brez različic lahko povlečete v pogovorno okno za objavo in bodo samodejno dodane v sistem Subversion.



Popravljanje zunanjih preimenovanj

Včasih datoteke preimenujete zunaj sistema Subversion, zato se v seznamu sprememb pojavita manjkajoča datoteka in datoteka brez različic. Da bi preprečili izgubo zgodovine, morate sistemu Subversion povedati, da med datotekama obstaja povezava. Enostavno izberite tako staro ime (manjkajoče) kot novo ime (brez različic) in uporabite **Kontekstni meni** → **Popravi premik** in s tem povežete datoteki v preimenovanje.

4.4.2. Sezname sprememb

Okno za objave omogoča uporabo seznamov sprememb, kar pomaga pri združevanju sprememb v logične skupine. Za več informacij o tej zmožnosti preberite [Razdelek 4.8, "Sezname sprememb"](#).

4.4.3. Izključevanje elementov iz okna objav

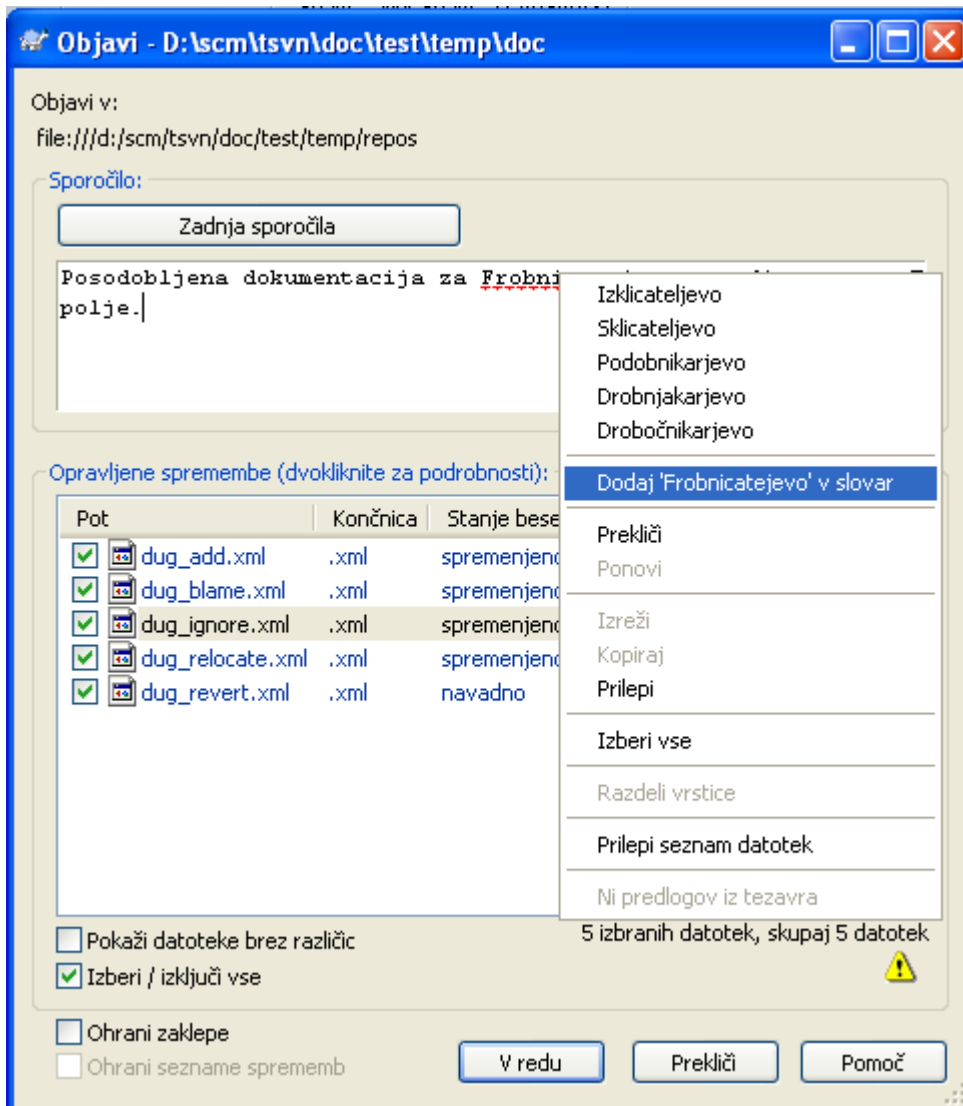
Včasih se v skladišču nahajajo datoteke, ki se pogosto spreminjajo, vendar jih ne želite vedno znova objavljati. Včasih to pomeni napako v procesu gradnje projekta - zakaj so te datoteke sploh pod nadzorom različic? Mogoče bi morali uporabljati predloge? Včasih pa je takšna situacija neizogibna. Primer: razvojno okolje spremeni časovni žig datoteke pri vsaki gradnji. Projektna datoteka mora biti pod nadzorom, saj vsebuje nastavitve za gradnjo, vendar ni potrebe, da jo objavljamo ob vsaki spremembi časovnega žiga.

Da bi olajšali delo v tašnih primerih, smo izdelali poseben seznam sprememb, ki se imenuje `ignore-on-commit`. Datoteke na tem seznamu so samodejno izključene iz izbora v oknu objav. Spremembe lahko še vedno objavite, vendar morate ročno izbrati takšno datoteko.

4.4.4. Sporočila dnevniških zapisov objav

Vedno vnesite sporočilo dnevniškega zapisa, ki opisuje spremembe, ki jih objavljate. S tem pri pregledovanju dnevniških zapisov vidite, kaj se je spremenilo in kdaj. Sporočilo je lahko poljubne dolžine. Veliko projektov natančno določa, kaj je potrebno vpisati, v katerem jeziku, včasih pa je določena zelo natančna oblika zapisa.

Sporočilom dnevniških zapisov lahko dodate enostavno oblikovanje, podobno tistemu, ki ga uporabljate pri elektronski pošti. Če želite besedilu dodati oblikovanje, uporabite `*besedilo*` za polkrepko pisavo, `_besedilo_` za podčrtano pisavo in `^besedilo^` za poševno pisavo.



Slika 4.9. Črkovalnik v oknu objave

TortoiseSVN vsebuje črkovalnik, ki vam pomaga pri pravilnem pisanju sporočil dnevniških zapisov. Označil bo vse nepravilno črkovane besede. Uporabite kontekstni meni za dostop do predlaganih popravkov. Seveda ne pozna prav *vseh* tehničnih terminov, tako bodo pravilno črkovane besede včasih označene kot napačne. Ne skrbite. Z uporabo kontekstnega menija jih lahko dodate v osebni slovar.

Okno za vnos sporočila dnevniškega zapisa pozna možnost samozaključevanja imen datotek in funkcij. Ta uporablja regularne izraze, da izlušči imena razredov in funkcij iz besedilnih datotek, ki jih objavljate, prav tako pa tudi sama imena datotek. Če beseda, ki jo tipkate, ustreza besedi v seznamu (potem, ko ste vpisali vsaj tri znake ali pritisnili tipko **Ctrl+Space**), se pojavi spustni meni, ki vam omogoča, da izberete cel izraz. Regularni izrazi, ki so na voljo, se nahajajo v podmapi `bin` namestitvene mape sistema TortoiseSVN. Definirate pa lahko tudi svoje regularne izraze in jih shranite v datoteko `%APPDATA%\TortoiseSVN\autolist.txt`. Seveda se pri nadgrnji sistema TortoiseSVN ta zasebni seznam ohrani. Če regularnih izrazov ne poznate, si lahko ogledate dokumentacijo na spletu na naslovih http://en.wikipedia.org/wiki/Regular_expression in <http://www.regular-expressions.info/>.

Predhodno vnesena sporočila dnevniških zapisov lahko ponovno uporabite. Enostavno kliknite na **Zadnja sporočila** in prikaže se vam nekaj zadnjih sporočil, ki ste jih vnesli v tej delovni kopiji. Število shranjenih sporočil lahko nastavite v oknu za nastavitve TortoiseSVN.

Shranjena sporočila lahko izbrišete na strani **Shranjeni podatki** v oknu za nastavitve TortoiseSVN, posamezna sporočila pa lahko izbrišete iz okna **Zadnja sporočila**, če držite pritisnjeno tipko **Delete**.

If you want to include the checked paths in your log message, you can use the command **Context Menu** → **Paste filename list** in the edit control.

Another way to insert the paths into the log message is to simply drag the files from the file list onto the edit control.



Posebne lastnosti map

Obstaja več posebnih lastnosti za mape, ki vam omogočajo več nadzora nad oblikovanjem sporočil dnevniških zapisov in jezikom, ki se uporabi za preverjanje črkovanja. Več o tem pove [Razdelek 4.17, "Nastavitve projekta"](#).

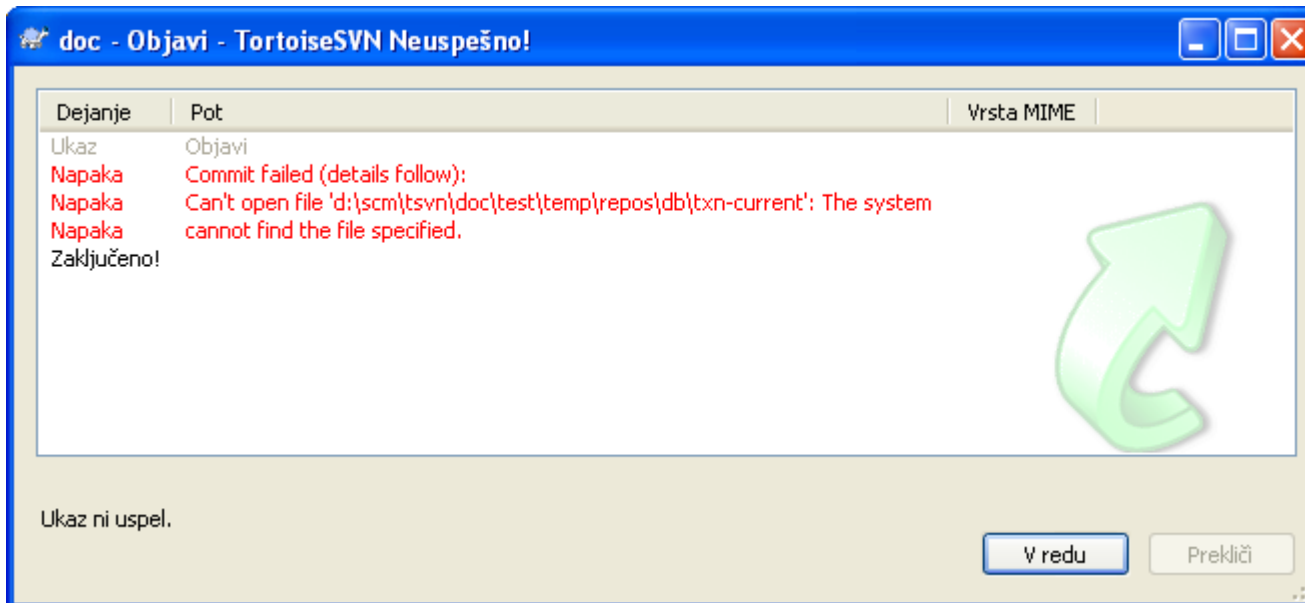


Integracija s sistemi za sledenje hroščev

Če uporabljate sistem za sledenje zadev, lahko v vnosno polje ID hrošča / št. zadeve: vpišete eno ali več številke zadev. Če vpišete več številke, naj bodo ločene z vejico. Druga možnost je, da uporabljate podporo sistem za sledenje zadev na osnovi regularnih izrazov. V tem primeru lahko številko zadeve navedete nekje znotraj sporočila dnevniškega zapisa. Več o tem vam pove [Razdelek 4.28, "Integracija s sistemi za sledenje zadev"](#).

4.4.5. Napredek objave

Po pritisku na gumb **V redu** se pojavi okno napredka objave.



Slika 4.10. Okno napredka prikazuje napredovanje objave

Okno napredka uporablja barvno kodiranje za označevanje različnih operacij objave.

Modra

Objavljanje spremembe.

Škrlatna

Objavljanje novega elementa.

Temno rdeča

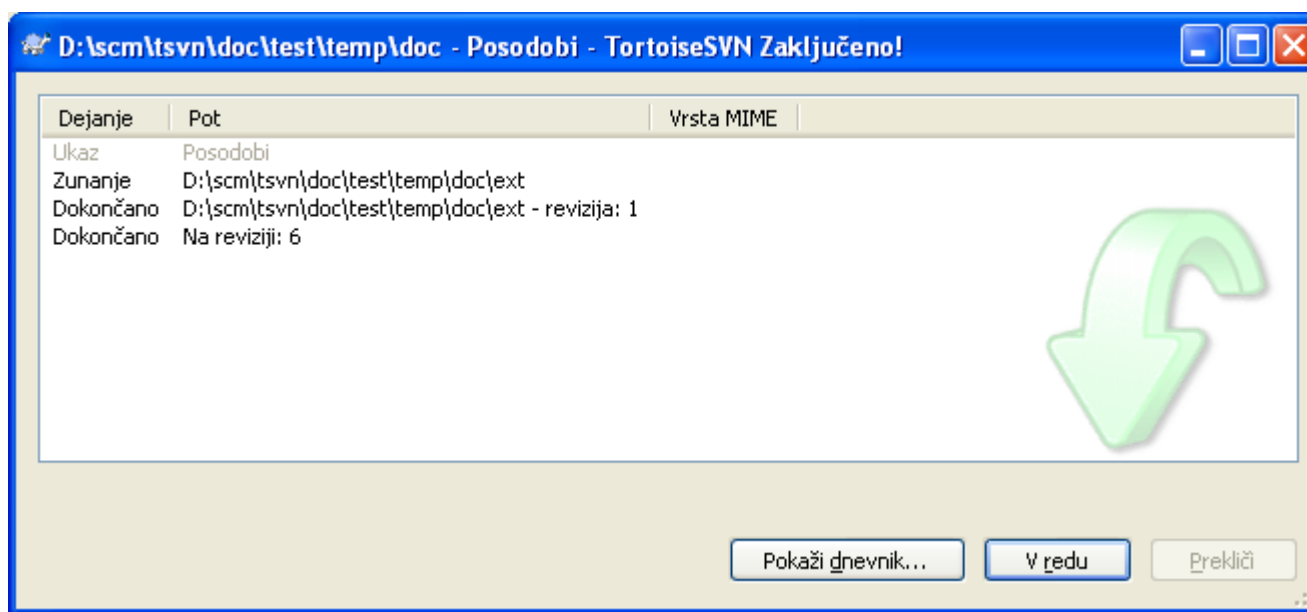
Objavljanje brisanja ali zamenjave.

Črna

Ostali elementi.

To je privzeta barvna shema, vendar lahko barve poljubno nastavite v oknu za nastavitve. Za več informacij preberite [Razdelek 4.30.1.4, "Nastavitev barv TortoiseSVN"](#).

4.5. Posodobite delovno kopijo s spremembami ostalih uporabnikov



Slika 4.11. Okno napredka prikazuje končano posodobitev

Ob določenih časovnih razmakih morate poskrbeti, da se spremembe, ki jih na projektu naredijo drugi uporabniki, vnesejo v vašo delovno kopijo. Procesu prenosa sprememb iz strežnika v delovno kopijo rečemo *posodabljanje*. Posodobite lahko posamezne datoteke, skupino izbranih datotek ali celotno hierarhijo map. Ko želite izvesti posodobitev, izberite datoteke in/ali mape in iz kontekstnega menija zaženite TortoiseSVN → Posodobi. Pojavi se okno, ki prikazuje napredek posodobitve. Spremembe, ki so jih naredili ostali uporabniki, se spojijo v vašo delovno kopijo, pri tem pa se vaše spremembe ohranijo. V skladišču se s posodobitvijo delovne kopije *ne* zgodi nobena sprememba.

Okno napredka uporablja barvno označevanje različnih akcij posodobitev

Škratna

Nov element dodan v delovno kopijo.

Temno rdeča

Odvečen element brisan iz delovne kopije ali manjkajoč element zamenjan v delovni kopiji.

Zelena

Spremembe iz skladišča so bile uspešno spojene s krajevnimi spremembami.

Svetlo rdeča

Spremembe iz skladišča, spojene s krajevnimi spremembami, so povzročile spore, ki jih morate rešiti.

Črna

Nespremenjen element v delovni kopiji, posodobljen iz nove verzije v skladišču.

To je privzeta barvna shema, vendar lahko barve poljubno nastavite v oknu za nastavitve. Za več informacij preberite [Razdelek 4.30.1.4, "Nastavitev barv TortoiseSVN"](#).

Če med posodabljanjem nastanejo *spori* (to se lahko zgodi, če je nek drug uporabnik spremenil iste vrstice v datoteki kot vi, vendar so spremembe različne od vaših), okno pokaže spore v rdeči barvi. Z dvoklikom na te vrstice zaženete zunanje orodje za spajanje, s katerim lahko spore rešite.

Ko je posodobitev končana, dialog napredka pod seznamom datotek pokaže povzetek - število posodobljenih, dodanih, odstranjenih, spornih... elementov. Povzetek lahko skopirate na odložišče s kombinacijo tipk **Ctrl+C**.

Ukaz Posodobi ne omogoča nobenih nastavitvev, ampak le posodobi vašo delovno kopijo na zadnjo (HEAD) revizijo v skladišču. To je tudi najbolj pogosta uporaba posodobitve. Če želite imeti več nadzora nad procesom posodabljanja, uporabite ukaz TortoiseSVN → Posodobi na revizijo.... Ta vam omogoča, da delovno kopijo posodobite na neko določeno revizijo, ne samo na zadnjo. Denimo, da ste naredili delovno kopijo iz revizije 100, sedaj pa želite pogledati stanje, kot je bilo v reviziji 50 - preprosto posodobite vašo delovno kopijo na revizijo 50. V istem pogovornem oknu lahko izberete tudi *globino*, do katere se izvede posodobitev na trenutni mapi. Za definicijo uporabljenih izrazov pogledajte v [Razdelek 4.3.1, "Globina prevzema"](#). Privzeta globina je *delovna kopija*, ki ohrani obstoječo nastavitvev globine. Izberete lahko tudi, ali želite pri posodabljanju prezreti zunanje projekte (projekte, ki imajo nastavljeno lastnost `svn:externals`).



Opozorilo

Če datoteko ali mapo posodobite na neko določeno revizijo, na teh elementih ne smete delati sprememb. Če boste spremembe poskusili objaviti, boste dobili obvestilo, da so datoteke oziroma mape "zastarele"! Če želite razveljaviti spremembe na datoteki in začeti znova iz zgodnejše revizije, se lahko zavrtite nazaj na prejšnjo revizijo iz dnevnika. Preberite [Razdelek B.4, "Kako prevrtim nazaj revizije v skladišču"](#) za nadaljnje informacije in druge metode.

Update to Revision can occasionally be useful to see what your project looked like at some earlier point in its history. But in general, updating individual files to an earlier revision is not a good idea as it leaves your working copy in an inconsistent state. If the file you are updating has changed name, you may even find that the file just disappears from your working copy because no file of that name existed in the earlier revision. You should also note that the item will show a normal green overlay, so it is indistinguishable from files which are up-to-date.

If you simply want a local copy of an old version of a file it is better to use the Context Menu → Save revision to... command from the log dialog for that file.



Več datotek/map

Če v Raziskovalcu izberete več datotek in map in potem uporabite ukaz Posodobi, bodo vse datoteke/mape posodobljene ena za drugo. TortoiseSVN poskrbi, da se vse datoteke/mape, ki se nahajajo v istem skladišču, posodobijo na natančno isto revizijo, tudi če se med procesom posodabljanja v skladišču zgodi nova objava!



Krajevna datoteka že obstaja

Včasih posodabljanje ni uspešno in dobite obvestilo, da v delovni kopiji že obstaja datoteka z enakim imenom. To se zgodi takrat, ko želi Subversion prevzeti iz skladišča novo dodano datoteko in ugotovi, da v delovni kopiji že obstaja datoteka z enakim imenom. Subversion nikoli ne prepíše datoteke, ki ni pod nadzorom - lahko vsebuje nekaj, na čemer delate in ima slučajno povsem enako ime, kot ga je nek drug uporabnik uporabil za novo dodano datoteko.

Če se pojavi takšna napaka, jo lahko rešite tako, da preimenujete krajevno datoteko brez različic. Po posodobitvi lahko preverite, če je preimenovana datoteka še potrebna.

Če se sporočila o napakah ponavljajo, raje uporabite TortoiseSVN → Preveri spremembe, da dobite seznam težavnih datotek. Na ta način se lotite reševanja vseh naenkrat.

4.6. Reševanje sporov

Once in a while, you will get a *conflict* when you update/merge your files from the repository or when you switch your working copy to a different URL. There are two kinds of conflicts:

file conflicts

A file conflict occurs if two (or more) developers have changed the same few lines of a file.

tree conflicts

A tree conflict occurs when a developer moved/renamed/deleted a file or folder, which another developer either also has moved/renamed/deleted or just modified.

4.6.1. File Conflicts

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

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

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

imedatoteke.kon.mine

To je vaša datoteka v delovni kopiji, preden ste jo posodobili - ne vsebuje oznak sporov. Datoteka vsebuje zgolj vaše zadnje spremembe.

imedatoteke.kon.rSTARAREVIZIJA

To je datoteka prevzete revizije (BASE), preden ste posodobili delovno kopijo. Torej datoteka, ki ste jo prevzeli, preden ste naredili zadnje spremembe.

imedatoteke.kon.rNOVAREVIZIJA

To je datoteka, ki jo je odjemalec Subversion pri posodabljanju delovne kopije prejel od strežnika. Ta datoteka ustreza reviziji HEAD v skladišču.

Uporabite lahko zunanje orodje za spajanje / urejanje sporov z ukazom TortoiseSVN → Uredi spore ali pa uporabite katerikoli urejevalnik besedil in ročno rešite spor. Odločiti se morate, kako mora koda izgledati, narediti ustrezne spremembe in shraniti datoteko.

Nato uporabite ukaz TortoiseSVN → Rešen in objavite svoje spremembe v skladišču. Upoštevajte, da ukaz Reši ne reši sporov, ampak le odstrani datoteki imedatoteke.kon.mine in imedatoteke.kon.r*, da lahko spremembe objavite.

Če se pojavijo spori pri dvojiških datotekah, Subversion ne poskuša sam spajati datotek. Krajevna datoteka ostane nespremenjena (natančno takšna kot ob vaši zadnji spremembi), pojavijo pa se datoteke imedatoteke.kon.r*. Če želite preklicati svoje spremembe in uporabiti različico datoteke iz

skladišča, uporabite ukaz Povrni. Če želite obdržati vašo različico in povziti različico v skladišču, uporabite ukaz Rešen, nato pa objavite spremembe.

Ukaz Rešen lahko uporabite tudi za več datotek, če kliknete na nadrejeno mapo in izberete TortoiseSVN → Rešen... Pokaže se okno s seznamom spornih datotek v mapi, v katerem izberete datoteke, ki jih želite označiti kot rešene.

4.6.2. Tree Conflicts

A tree conflict occurs when a developer moved/renamed/deleted a file or folder, which another developer either also has moved/renamed/deleted or just modified. There are many different situations that can result in a tree conflict, and all of them require different steps to resolve the conflict.

When a file is deleted locally in Subversion, the file is also deleted from the local file system, so even if it is part of a tree conflict it cannot show a conflicted overlay and you cannot right click on it to resolve the conflict. Use the **Check for Modifications** dialog instead to access the **Edit conflicts** option.

TortoiseSVN can help find the right place to merge changes, but there may be additional work required to sort out the conflicts. Remember that after an update the working BASE will always contain the revision of each item as it was in the repository at the time of update. If you revert a change after updating it goes back to the repository state, not to the way it was when you started making your own local changes.

4.6.2.1. Local delete, incoming edit upon update

1. Developer A modifies `Foo.c` and commits it to the repository
2. Developer B has simultaneously moved `Foo.c` to `Bar.c` in his working copy, or simply deleted `Foo.c` or its parent folder.

An update of developer B's working copy results in a tree conflict:

- `Foo.c` has been deleted from working copy, but is marked with a tree conflict.
- If the conflict results from a rename rather than a delete then `Bar.c` is marked as added, but does not contain developer A's modifications.

Developer B now has to choose whether to keep Developer A's changes. In the case of a file rename, he can merge the changes to `Foo.c` into the renamed file `Bar.c`. For simple file or directory deletions he can choose to keep the item with Developer A's changes and discard the deletion. Or, by marking the conflict as resolved without doing anything he effectively discards Developer A's changes.

The conflict edit dialog offers to merge changes if it can find the original file of the renamed `Bar.c`. Depending on where the update was invoked, it may not be possible to find the source file.

4.6.2.2. Local edit, incoming delete upon update

1. Developer A moves `Foo.c` to `Bar.c` and commits it to the repository.
2. Developer B modifies `Foo.c` in his working copy.

Or in the case of a folder move ...

1. Developer A moves parent folder `FooFolder` to `BarFolder` and commits it to the repository.
2. Developer B modifies `Foo.c` in his working copy.

An update of developer B's working copy results in a tree conflict. For a simple file conflict:

- `Bar.c` is added to the working copy as a normal file.

- `Foo.c` is marked as added (with history) and has a tree conflict.

For a folder conflict:

- `BarFolder` is added to the working copy as a normal folder.
- `FooFolder` is marked as added (with history) and has a tree conflict.

`Foo.c` is marked as modified.

Developer B now has to decide whether to go with developer A's reorganisation and merge her changes into the corresponding file in the new structure, or simply revert A's changes and keep the local file.

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. 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. In this case use the **Remove** button in the conflict editor dialog to clean up and mark the conflict as resolved.

If Developer B decides that A's changes were wrong then she must choose the **Keep** 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.

4.6.2.3. Local delete, incoming delete upon update

1. Developer A moves `Foo.c` to `Bar.c` and commits it to the repository
2. Developer B moves `Foo.c` to `Bix.c`

An update of developer B's working copy results in a tree conflict:

- `Bix.c` is marked as added with history.
- `Bar.c` is added to the working copy with status 'normal'.
- `Foo.c` is marked as deleted and has a tree 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.

Then developer B has to decide which new filename of `Foo.c` to keep - the one done by developer A or the rename done by himself.

After developer B has manually resolved the conflict, the tree conflict has to be marked as resolved with the button in the conflict editor dialog.

4.6.2.4. Local missing, incoming edit upon merge

1. Developer A working on trunk modifies `Foo.c` and commits it to the repository
2. Developer B working on a branch moves `Foo.c` to `Bar.c` and commits it to the repository

A merge of developer A's trunk changes to developer B's branch working copy results in a tree conflict:

- `Bar.c` is already in the working copy with status 'normal'.
- `Foo.c` is marked as missing with a tree conflict.

To resolve this conflict, Developer B has to mark the file as resolved in the conflict editor dialog, which will remove it from the conflict list. She then has to decide whether to copy the missing file `Foo.c` from

the repository to the working copy, whether to merge Developer A's changes to `Foo.c` into the renamed `Bar.c` or whether to ignore the changes by marking the conflict as resolved and doing nothing else.

Note that if you copy the missing file from the repository and then mark as resolved, your copy will be removed again. You have to resolve the conflict first.

4.6.2.5. Local edit, incoming delete upon merge

1. Developer A working on trunk moves `Foo.c` to `Bar.c` and commits it to the repository
2. Developer B working on a branch modifies `Foo.c` and commits it to the repository.

There is an equivalent case for folder moves, but it is not yet detected in Subversion 1.6 ...

1. Developer A working on trunk moves parent folder `FooFolder` to `BarFolder` and commits it to the repository.
2. Developer B working on a branch modifies `Foo.c` in her working copy.

A merge of developer A's trunk changes to developer B's branch working copy results in a tree conflict:

- `Bar.c` is marked as added.
- `Foo.c` is marked as modified with a tree conflict.

Developer B now has to decide whether to go with developer A's reorganisation and merge her changes into the corresponding file in the new structure, or simply revert A's changes and keep the local file.

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. In this case use the **Remove** button in the conflict editor dialog to clean up and mark the conflict as resolved.

If Developer B decides that A's changes were wrong then she must choose the **Keep** 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.

4.6.2.6. Local delete, incoming delete upon merge

1. Developer A working on trunk moves `Foo.c` to `Bar.c` and commits it to the repository
2. Developer B working on a branch moves `Foo.c` to `Bix.c` and commits it to the repository

A merge of developer A's trunk changes to developer B's branch working copy results in a tree conflict:

- `Bix.c` is marked with normal (unmodified) status.
- `Bar.c` is marked as added with history.
- `Foo.c` is marked as missing and has a tree 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. 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.

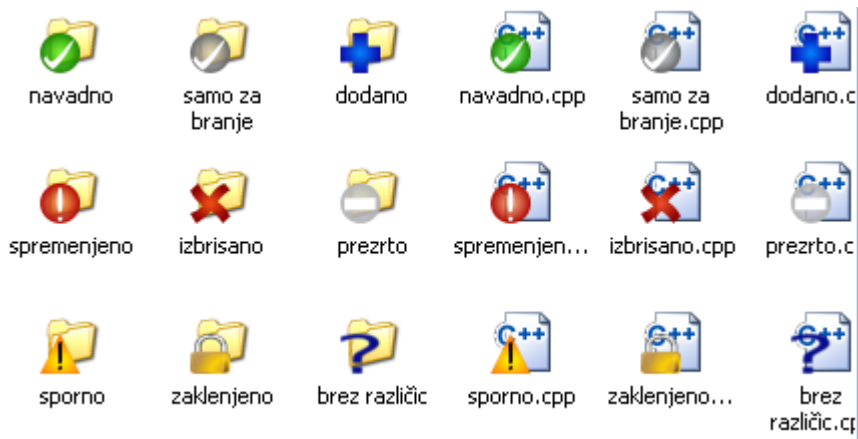
Then developer B has to decide which new filename of `Foo.c` to keep - the one done by developer A or the rename done by himself.

After developer B has manually resolved the conflict, the tree conflict has to be marked as resolved with the button in the conflict editor dialog.

4.7. Pridobivanje informacije o stanju

Ko delate na delovni kopiji, pogosto želite vedeti, katere datoteke ste spremenili/dodali/odstranili ali preimenovali oziroma celo, katere datoteke so spremenili in objavili ostali uporabniki.

4.7.1. Prekrivne ikone



Slika 4.12. Raziskovalec prikaže prekrivne ikone

Ko prevzamete delovno kopijo iz skladišča Subversion, imajo datoteke v Raziskovalcu spremenjene ikone. To je razlog za veliko popularnost programa TortoiseSVN. TortoiseSVN doda datotekam prekrivno ikono, ki se prekriva z originalno ikono. Prekrivna ikona je odvisna od statusa datoteke.



Sveže prevzeta delovna kopija ima zeleno prekrivno ikono. To pomeni, da je status datoteke/mape *običajno*.



Ko začnete datoteko urejati, se njen status spremeni v *spremenjeno*, prekrivna ikona pa se spremeni v rdeč klikaj. Tako hitro vidite, katere datoteke so se od zadnje posodobitve spremenile in jih je potrebno objaviti.



Če se pri posodabljanju pojavi *spor*, se ikona spremeni v rumen klikaj.



Če ste datoteki nastavili lastnost `svn:needs-lock`, Subversion napravi to datoteko samo za branje, dokler za datoteko ne pridobite zaklepa. Datoteke, ki so namenjene samo za branje, imajo takšno ikono, da lahko hitro vidite, da morate pred urejanjem datoteke zanjo pridobiti zaklep.



Če ste lastnik zaklepa datoteke, ki ima stanje *običajno*, vas ta ikona opominja, da morate zaklep sprostiti, če ga ne uporabljate, da lahko tudi drugi uporabniki objavijo spremembe na tej datoteki.



Ta ikona pove, da so nekatere datoteke ali mape znotraj trenutne mape označene za *brisanje* iz nadzora različic ali pa da se datoteka, ki je pod nadzorom, ne nahaja v mapi.



Znak plus pomeni, da je datoteka ali mapa označena za *dodajanje* v nadzor različic.



Znak pomeni, da je datoteka ali mapa v nadzoru različic *prezrta*. Prikaz te prekrivne ikone je možno vključiti ali izključiti.



Ta ikona označuje datoteke in mape, ki niso pod nadzorom, prav tako pa niso prezrte. Uporabo te ikone lahko izključite.

Lahko celo ugotovite, da omenjene ikone niso vse na voljo na vašem sistemu. Razlog je v omejitvi števila prekrivnih ikon sistema Windows in če uporabljate tudi starejšo različico programa TortoiseCVS, potem ni več dovolj prostih prekrivnih ikon. TortoiseSVN poskuša biti "pošten državljan (TM)" in ne uporablja veliko prekrivnih ikon, da ostane prostor tudi za ostale aplikacije.

Na trgu je več odjemalcev tipa Tortoise (TortoiseCVS, TortoiseHG, ...), tako da je omejitev števila prekrivnih resna težava. Zato je projekt TortoiseSVN predstavil skupni nabor ikon, zbranih v knjižnici DLL, ki jo lahko uporabljajo vsi odjemalci tipa Tortoise. Preverite, ali vaš odjemalec že uporablja to knjižnico :-)

Za opis, kakšnemu statusu v sistemu Subversion ustrezajo prekrivne ikone in ostale tehnične podrobnosti preberite [Razdelek F.1, "Prekrivne ikone"](#).

4.7.2. Stolpci TortoiseSVN v Raziskovalcu

Iste informacije, ki jih dajejo prekrivne ikone (in še veliko več), je možno prikazati z dodatnimi stolpci v podrobnem pogledu Raziskovalca.

Enostavno desnokliknite na eno od glav stolpcev in iz kontekstnega menija izberite Več.... Pojavi se pogovorno okno, kjer lahko določite stolpce in njihov vrstni red, ki se prikaže v pogledu "Detajli". Poiščite stolce, katerih imena se začnejo s SVN. Označite tiste, ki jih želite prikazati in zaprite okno s klikom na gumb **V redu**. Stolpci se dodajo na desni strani. Z metodo povleci in spusti lahko spremenite vrstni red, prav tako pa jim lahko spremenite širino, tako da ustreza vašim potrebam.



Pomembno

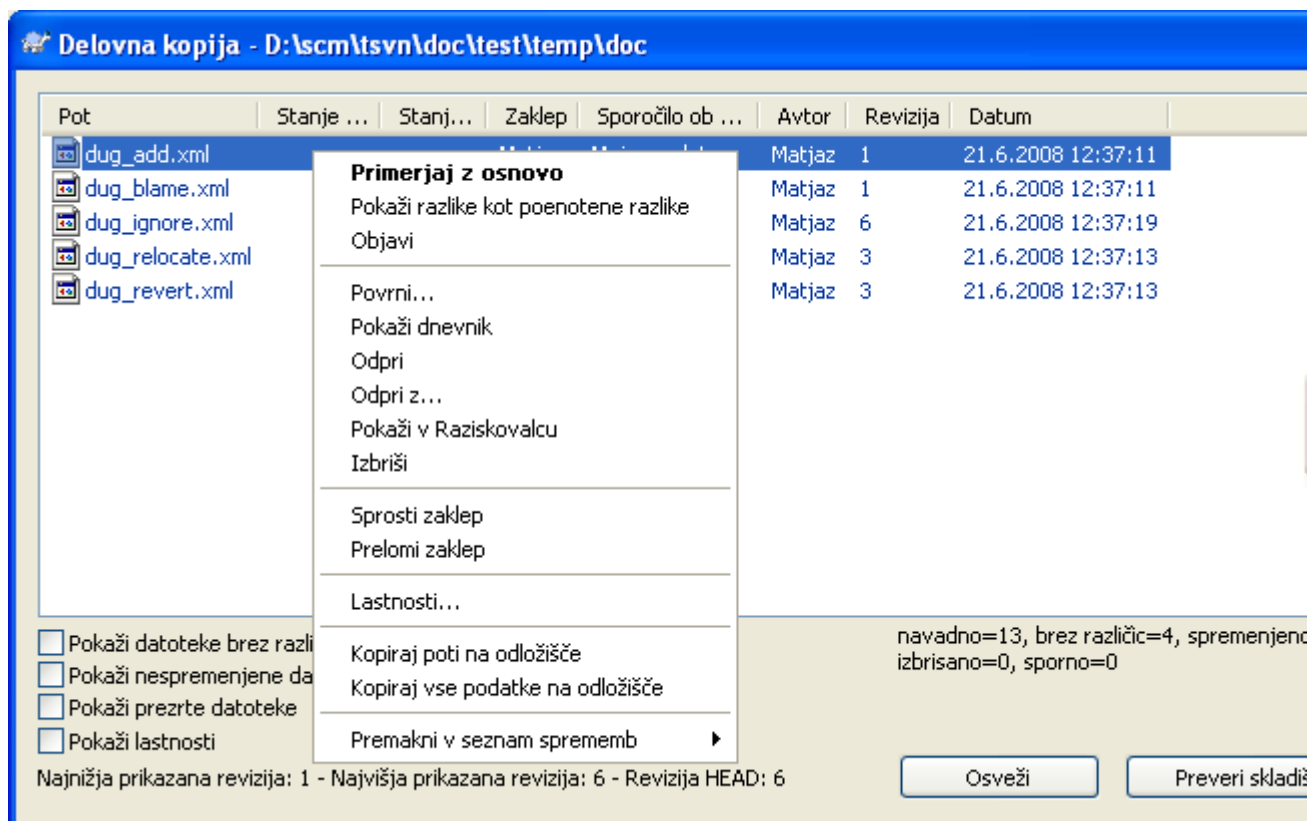
Dodatni stolci v Raziskovalcu na operacijskem sistemu Vista niso na voljo. Microsoft se je odločil, da dodatnih stolcev ne dopušča več za vse datoteke, ampak le za določene vrste datotek.



Namig

Če želite, da se trenutni raspored prikaže v vseh delovnih kopijah, lahko ta pogled nastavite kot privzet.

4.7.3. Krajevno in oddaljeno stanje



Slika 4.13. Preveri posodobitve

Pogosto je zelo koristno, če veste, katere datoteke ste spremenili in katere datoteke so spremenili vaši sodelavci. Takrat pride prav ukaz TortoiseSVN → Preveri posodobitve.... To pogovorno okno prikaže vse datoteke, ki so se kakorkoli spremenile, prav tako pa tudi vse datoteke brez različic.

If you click on the Check Repository then you can also look for changes in the repository. That way you can check before an update if there's a possible conflict. You can also update selected files from the repository without updating the whole folder. By default, the Check Repository button only fetches the remote status with the checkout depth of the working copy. If you want to see all files and folders in the repository, even those you have not checked out, then you have to hold down the **Shift** key when you click on the Check Repository button.

Pogovorno okno uporablja barvno kodiranje, da poudari stanje.

Modra

Krajevno spremenjeni elementi.

Škratna

Dodani elementi. Elementi, ki so bili dodani z zgodovino, imajo pred v stolpcu Stanje besedila znak +, namig pa pokaže, od kje je bil element skopiran.

Temno rdeča

Izbrisani ali manjkajoči elementi.

Zelena

Elementi so spremenjeni krajevno in v skladišču. Ob posodobitvi bodo spremembe spojene. Situacija *lahko* povzroči spore pri posodobitvi.

Svetlo rdeča

Elementi, ki so bili krajevno spremenjeni in izbrisani v skladišču ali spremenjeni v skladišču in izbrisani krajevno. Ob posodobitvi se spori *bodo* pojavili.

Črna

Nespremenjeni elementi in elementi brez različic.

To je privzeta barvna shema, vendar lahko barve poljubno nastavite v oknu za nastavitve. Za več informacij preberite [Razdelek 4.30.1.4, "Nastavitev barv TortoiseSVN"](#).

Elementi, ki so bili preklopljeni na drugo pot v skladišču, so označeni z znakom (s). Lahko da ste kakšen element preklopili, ko ste delali na neki veji in ga pozabili preklopiti nazaj. To je opozorilo!

Iz kontekstnega menija lahko prikažete spremembe. Z uporabo **Kontekstni meni** → **Primerjaj z osnovo** preverite spremembe, ki ste jih naredili *vi*. Z ukazom **Kontekstni meni** → **Pokaži razlike kot poenotene razlike** preverite spremembe, ki so jih v skladišču naredili drugi uporabniki.

Posameznim datotekam lahko povrnete spremembe. Če ste pomotoma izbrisali datoteko, bo le-ta označena kot *manjkajoča*, zato lahko uporabite ukaz **Povrni**, da jo pridobite nazaj.

Datoteke brez različic in prezrte datoteke lahko pošljete v Koš z ukazom **Kontekstni meni** → **Zbriši**. Če želite datoteke zbrisati trajno (brez pošiljanja v Koš), ob kliku na **Zbriši** držite pritisnjeno tipko **Shift**.

Če želite podrobno pregledati datoteko, jo lahko od tu povlečete v drugo aplikacijo, npr. urejevalnik besedil ali razvojno okolje (IDE).

Stolpci so prilagodljivi. Če desno kliknete na glavo stolpca, se pojavi kontekstni meni, ki vam omogoča izbiro prikazanih stolpcev. Spremenite lahko tudi širino stolpcev z uporabo potega ročaja, ki se pojavi, ko greste z miško čez mejo stolpca. Spremembe se shranijo, tako da boste prilagojen pogled videli tudi naslednjič.

Če delate na več nepovezanih nalogah hkrati, lahko sestavljate datoteke skupaj v sezname sprememb. Za več informacij preberite [Razdelek 4.4.2, "Seznami sprememb"](#).

Na dnu okna vidite povzetek območja revizij, ki jih uporabljate v delovni kopiji. To so *objavljene* revizije, ne *posodobljene* revizije; predstavljajo območje revizij, v katerih so bile datoteke nazadnje objavljene in ne revizije, pri katerih je bila narejena posodobitev delovne kopije. Območje revizij se nanaša le na prikazane elemente, ne na celovno delovno kopijo. Če želite, da se povzetek nanaša na celotno delovno kopijo, potrdite polje **Prikaži nespremenjene datoteke**.

**Namig**

Če želite videti ploski pogled delovne kopije (vse datoteke in mape na vseh nivojih hierarhije), to najlažje dosežete s pogovornim oknom **Preveri spremembe**. Edino, kar morate storiti, je, da potrdite polje **Prikaži nespremenjene datoteke**.

**Popravljanje zunanjih preimenovanj**

Včasih datoteke preimenujete zunaj sistema Subversion, zato se v seznamu sprememb pojavita manjkajoča datoteka in datoteka brez različic. Da bi preprečili izgubo zgodovine, morate sistemu Subversion povedati, da med datotekama obstaja povezava. Enostavno izberite tako staro ime (manjkajoče) kot novo ime (brez različic) in uporabite **Kontekstni meni** → **Popravi premik** in s tem povežete datoteki v preimenovanje.

4.7.4. Pregledovanje razlik

Pogosto želite pogledati, kaj ste v datoteki spremenili. To storite z izbiro željene datoteke in uporabo ukaza Razlikuj iz kontekstnega menija. S tem poženete zunanji pregledovalnik razlik, ki bo primerjal trenutno datoteko z osnovno datoteko (revizija BASE), ki se je shranila, ko ste naredili zadnji prevzem ali posodobitev.



Namig

Tudi ko ne delate na delovni kopiji ali ko imate več verzij iste datoteke v mapi, lahko prikažete razlike:

V Raziskovalcu izberite datoteki, ki ju želite primerjati (n. pr. z uporabo tipke **Ctrl** in miške) in iz kontekstnega menija TortoiseSVN izberite Razlikuj. Datoteka, na katero ste kliknili nazadnje (datoteka, ki ima fokus oziroma datoteka s pikčasto obrobo) se bo obravnavala kot novejša.

4.8. Sezname sprememb

V idealnem svetu delate naenkrat na eni zadevi in vaša delovna kopija vsebuje le en skupek logičnih sprememb. V redu, nazaj v realnost. Večkrat se zgodi, da morate delati na več nepovezanih nalogah naenkrat in ko pogledate v okno za objave, vidite vse spremembe skupaj. Zmožnost *seznam sprememb* vam pomaga združiti datoteke v skupine. Tako lažje vidite, kaj delate. Seveda je to možno le, če se spremembe ne prekrivajo. Če dve različni nalogi zahtevata spremembo iste datoteke, sprememb ni mogoče ločiti.



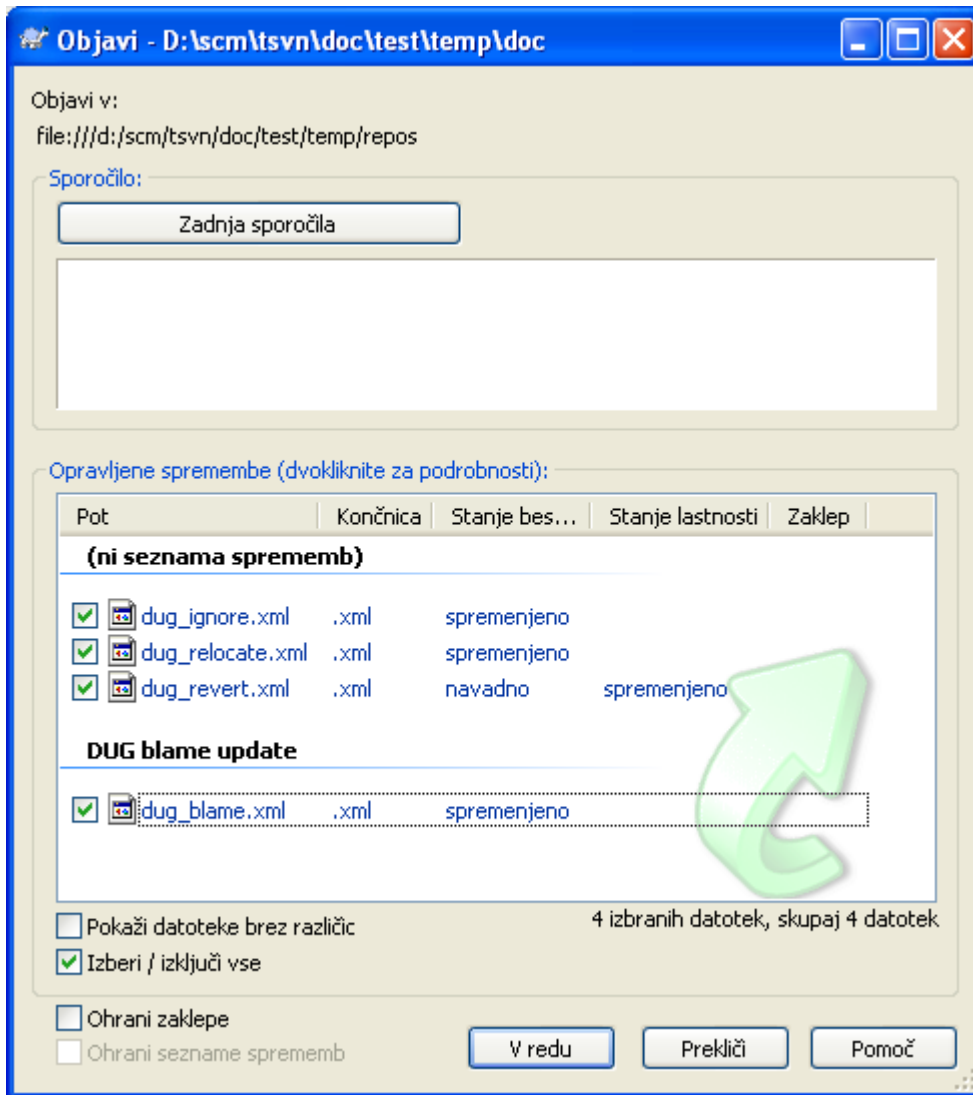
Pomembno

Možnost uporabe seznamov sprememb v programu TortoiseSVN je na voljo le v operacijskem sistemu Windows XP ali novejšem, ker uporablja možnosti lupine, ki v sistemu Windows 2000 ne obstajajo. Oprostite, ampak sistem Windows 2000 je že precej star, zato vas prosimo, da se ne pritožujete.

Sezname sprememb se uporabljajo na več mestih, najbolj pa so pomembni pri preverjanju sprememb in pri objavi le-teh. Začnimo z oknom za preverjanje sprememb, ki po delu na več zmožnostih in več spremenjenih datotekah pokaže dolg seznam datotek. Sedaj želite spremenjene datoteke urediti in jih glede nato, katerim novostim pripadajo, postaviti v ustrezne skupine.

Izberite eno ali več datotek in jih premaknite v seznam sprememb z uporabo ukaza **Kontekstni meni** → **Premakni v seznam sprememb**. Na začetku ni nobenega seznama sprememb, zato ga boste morali ob prvi uporabi ukaza ustvariti. Poimenujte ga smiselno glede na uporabo in kliknite gumb **V redu**. Okno za objavo se spremeni in prikazuje skupine elementov.

Ko ste ustvarili seznam sprememb, lahko vanj dodate elemente. Vzamete jih lahko iz drugega seznama sprememb ali pa kar iz Raziskovalca. Dodajanje iz Raziskovalca je uporabno, saj lahko dodajate datoteke preden le-te sploh spremenite. To lahko naredite tudi neposredno iz okna za preverjanje sprememb, vendar le, če imate vključeno možnost prikaza datotek brez različic.



Slika 4.14. Okno za objave s seznamami sprememb

V pogovornem oknu za objave vidite iste datoteke, združene v seznane sprememb. Poleg vizualnega pregleda lahko skupine uporabljate tudi za hitro izbiro datotek za objavo s izbirom glave skupine.

V sistemu Windows XP je ob desnem kliku na glavo skupine na voljo kontekstni meni, ki omogoča potrditev ali izklop vseh elementov skupine. V sistemu Windows Vista ta meni ni obvezen. Kliknite na glavo skupine. S tem izberete vse elemente. Nato potrdite en element - s tem potrdite vse izbrane elemente.

TortoiseSVN si rezervira en seznam sprememb za lastno uporabo in sicer `ignore-on-commit`. Seznam se uporablja za datoteke, ki jih skoraj nikoli ne objavljate, čeprav imajo krajevne spremembe. Ta zmožnost je opisana v poglavju [Razdelek 4.4.3, "Izključevanje elementov iz okna objav"](#).

Ob objavi datotek iz seznama sprememb običajno želite, da se te odstranijo iz seznama, tako da je to privzeta možnost pri vsaki objavi. Če želite, da datoteke ostanejo del seznama sprememb, potrdite polje Ohrani sezname sprememb na dnu pogovornega okna.



Namig

Seznami sprememb so povsem stvar odjemalca. Ustvarjanje in odstranjevanje seznamov sprememb ne vpliva na skladišče, prav tako pa tudi ne na delovne kopije drugih uporabnikov. Seznami sprememb so priročna možnost organiziranja spremenjenih datotek.

4.9. Pogovorno okno Dnevnik

Za vsako objavljeno spremembo vnesite sporočilo dnevniškega zapisa. Tako lahko kasneje ugotovite, katere spremembe ste kdaj naredili in imate podroben pregled celotnega razvojnega procesa.

Pogovorno okno dnevnika pridobi dnevniške zapise in jih prikaže. Zaslona je razdeljen na tri dele.

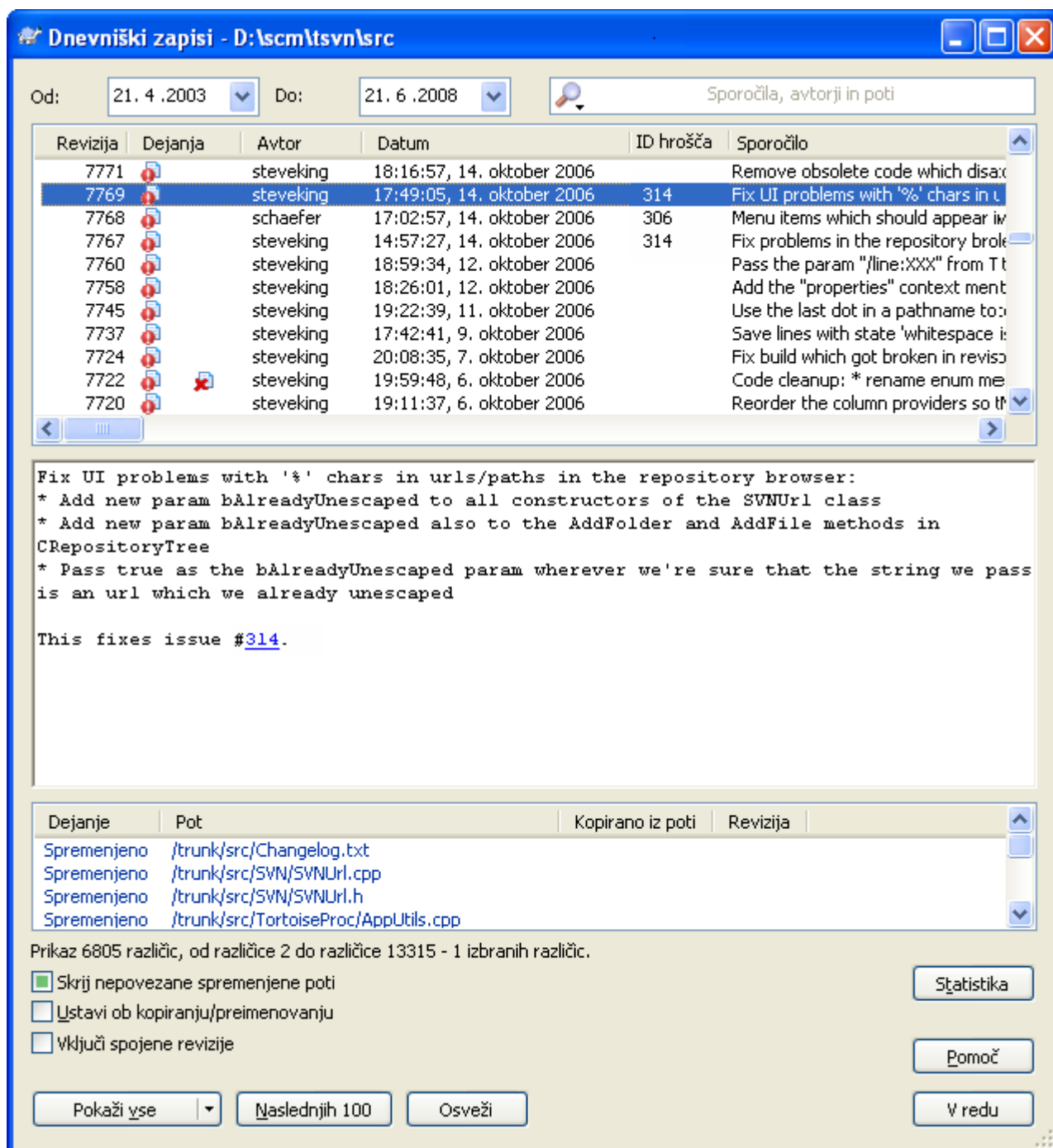
- Zgornji del prikazuje seznam revizij, kjer je bila datoteka/mapa objavljena. Povzetek vsebuje čas in datum, avtorja objavljene spremembe in začetni del sporočila dnevnika.

Vrstice, obarvane z modro barvo, pomenijo, da je bil nek element prekopiran na to vejo razvoja (mogoče iz druge veje).

- Srednji del prikazuje sporočilo dnevniškega zapisa za izbrano revizijo.
- Spodnji del prikazuje seznam datotek in map, ki so bile spremenjene v izbrani reviziji.

Pogovorno okno pa naredi še veliko več - ponudi vam kontekstni meni, ki prinaša še več informacij o zgodovini projekta.

4.9.1. Klicanje pogovornega okna dnevniških zapisov



Slika 4.15. Okno dnevnika

Obstaja veliko mest, od koder lahko lahko pokličete okno dnevnika:

- Iz kontekstnega podmenija TortoiseSVN
- Iz strani lastnosti
- Iz pogovornega okna napredka po posodobitvi delovne kopije. V tem primeru Dnevnik prikaže le tiste revizije, ki so se spremenile od zadnje posodobitve.

If the repository is unavailable you will see the Want to go offline? dialog, described in [Razdelek 4.9.10, "Nepovezan način"](#).

4.9.2. Akcije dnevnika

Zgornji del okna vsebuje stolpec **Dejanja** z ikonami, ki povzemajo, kaj se je v določeni reviziji zgodilo. Obstajajo štiri različne ikone, vsaka je prikazana v svojem stolpcu.



Če je v reviziji prišlo do spremembe datoteke ali mape, se v prvem stolpcu prikaže ikona *spremenjeno*.



Če je bila v reviziji dodana datoteka ali mapa, se v drugem stolpcu prikaže ikona *dodano*.

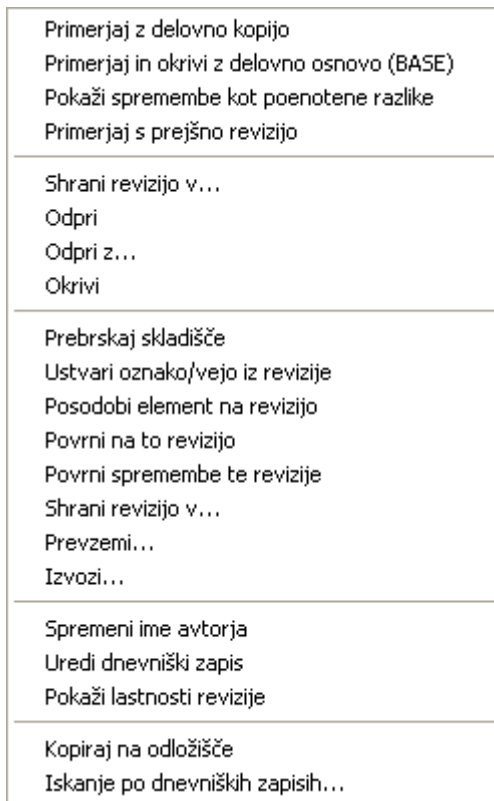


Če je bila v reviziji izbrisana datoteka ali mapa, se v tretjem stolpcu prikaže ikona *zbrisano*.



Če je revizija zamenjala datoteko ali mapo, se v četrtem stolcu prikaže ikona *zamenjano*.

4.9.3. Pridobivanje dodatnih informacij



Slika 4.16. Zgornji del Dnevnika s kontekstnim menijem

Zgornja polovica Dnevnika ponuja kontekstni meni, ki vam omogoča dostop do veliko informacij. Nekateri vnosi v meniju so na voljo le pri datotekah, drugi le pri mapah.

Compare with working copy

Primerjate izbrane revizije z delovno kopijo. Privzeto orodje za razlikovanje je TortoiseMerge, ki je del paketa TortoiseSVN. Če ste zahtevali dnevnik za mapo, se bo prikazal seznam spremenjenih datotek, vi pa boste lahko pregledali narejene spremembe za vsako datoteko posebej.

Compare and blame with working BASE

Okrivite izbrano revizijo in datoteko v delovni osnovi z vizualno primerjavo razlik. Za podrobnejše informacije preberite **Razdelek 4.23.2, "Okrivi spremembe"**. (velja le za datoteke).

Show changes as unified diff

Pogledate spremembe v izbranih revizijah v obliki poenotene različice (oblika popravkov GNU). Prikaže le razlike in nekaj bližnjih vrstic. Ta oblika je težje berljiva kot vizualna primerjava, vendar prikaže vse spremembe v datoteki v zgoščeni obliki.

Compare with previous revision

Compare the selected revision with the previous revision. This works in a similar manner to comparing with your working copy. For folders this option will first show the changed files dialog allowing you to select files to compare.

Compare and blame with previous revision

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).

Save revision to...

Shranite izbrano revizijo datoteke, tako da imate starejšo različico te datoteke. (velja le za datoteke).

Open / Open with...

Odprete izbrano datoteko s privzetim pregledovalnikom ali programom po lastni izbiri. (velja le za datoteke).

Blame...

Okrivite datoteko do izbrane revizije (velja le za datoteke).

Browse repository

Odprete brskalnik po skladišču za ogled izbrane datoteke ali mape, kakršna je bila ob izbrani reviziji.

Create branch/tag from revision

Create a branch or tag from a selected revision. This is useful e.g. if you forgot to create a tag and already committed some changes which weren't supposed to get into that release.

Update item to revision

Update your working copy to the selected revision. Useful if you want to have your working copy reflect a time in the past, or if there have been further commits to the repository and you want to update your working copy one step at a time. It is best to update a whole directory in your working copy, not just one file, otherwise your working copy could be inconsistent.

If you want to undo an earlier change permanently, use **Revert to this revision** instead.

Revert to this revision

Revert to an earlier revision. If you have made several changes, and then decide that you really want to go back to how things were in revision N, this is the command you need. The changes are undone in your working copy so this operation does *not* affect the repository until you commit the changes. Note that this will undo *all* changes made after the selected revision, replacing the file/folder with the earlier version.

If your working copy is in an unmodified state, after you perform this action your working copy will show as modified. If you already have local changes, this command will merge the *undo* changes into your working copy.

What is happening internally is that Subversion performs a reverse merge of all the changes made after the selected revision, undoing the effect of those previous commits.

If after performing this action you decide that you want to *undo the undo* and get your working copy back to its previous unmodified state, you should use **TortoiseSVN** → **Revert** from within Windows Explorer, which will discard the local modifications made by this reverse merge action.

If you simply want to see what a file or folder looked like at an earlier revision, use **Update to revision** or **Save revision as...** instead.

Revert changes from this revision

Undo changes from which were made in the selected revision. The changes are undone in your working copy so this operation does *not* affect the repository at all! Note that this will undo the changes made in that revision only; it does not replace your working copy with the entire file at the earlier revision. This is very useful for undoing an earlier change when other unrelated changes have been made since.

If your working copy is in an unmodified state, after you perform this action your working copy will show as modified. If you already have local changes, this command will merge the *undo* changes into your working copy.

What is happening internally is that Subversion performs a reverse merge of that one revision, undoing its effect from a previous commit.

You can *undo the undo* as described above in **Revert to this revision**.

Merge revision to...

Merge the selected revision(s) into a different working copy. A folder selection dialog allows you to choose the working copy to merge into, but after that there is no confirmation dialog, nor any opportunity to try a test merge. It is a good idea to merge into an unmodified working copy so that you can revert the changes if it doesn't work out! This is a useful feature if you want to merge selected revisions from one branch to another.

Checkout...

Naredite svež prevzem izbrane mape in izbrane revizije. S tem se prikaže okno, v katerem potrdite naslov URL in revizijo in izberete lokacijo za prevzem.

Export...

Izvozite izbrane datoteke/mape v izbrani reviziji. Pokaže se pogovorno okno z naslovom URL in revizijo, kjer vnesete še lokacijo izvoza.

Edit author / log message

Uredite sporočilo dnevniškega zapisa in avtorja v predhodni objavi. Za dodatne informacije preberite [Razdelek 4.9.7](#), “[Spreminjanje sporočila dnevniškega zapisa in avtorja](#)”.

Show revision properties

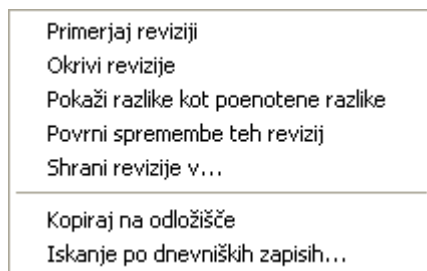
View and edit any revision property, not just log message and author. Refer to [Razdelek 4.9.7](#), “[Spreminjanje sporočila dnevniškega zapisa in avtorja](#)”.

Copy to clipboard

Prekopirate podrobnosti dnevnika izbranih revizij na odložišče. S tem prekopirate številko revizije, avtorja, datum, sporočilo in seznam spremenjenih elementov za vsako revizijo.

Search log messages...

Iščete po sporočilih dnevniških zapisov. Iskanje poteka po dnevniških zapisih, ki jih vnesete, in po povzetkih akcij, ki jih ustvari Subversion in so prikazani v spodnjem delu okna. Iskanje ni občutljivo na velike in male črke.



Slika 4.17. Kontekstni meni v zgornjem delu okna v primeru dveh izbranih revizij

Če izberete dve reviziji naenkrat (z uporabo tipke **Ctrl**), se kontekstni meni spremeni in ponudi manj možnosti. Lahko:

Compare revisions

Primerjate izbrani reviziji z grafičnim orodjem za razlikovanje. Privzeto orodje je TortoiseMerge, ki se namesti hkrati z orodjem TortoiseSVN.

Če izberete to možnost za mapo, se pojavi novo okno, ki vsebuje seznam datotek, za katere lahko pogledate spremembe. Več o primerjanju revizij si lahko preberete v [Razdelek 4.10.3, "Primerjanje map"](#).

Blame revisions

Okrivite dve reviziji in primerjate rezultate z grafičnim orodjem za razlikovanje. Za več informacij preberite [Razdelek 4.23.2, "Okrivi spremembe"](#).

Show differences as unified diff

Pogledate razlike med izbranimi revizijama v obliki poenotene razlike. Deluje za datoteke in mape.

Copy to clipboard

Skopirate sporočila dnevniških zapisov na odložišče, kot je opisano zgoraj.

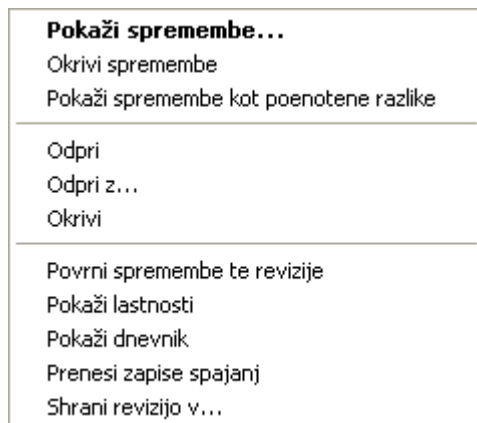
Search log messages...

Iščete med sporočili dnevniških zapisov, kot je opisano zgoraj.

Če izberete več revizij (z uporabo tipk **Ctrl** ali **Shift**), se v kontekstnem meniju pojavi možnost, ki vam omogoča povrnitev vseh sprememb, ki so bile narejene v teh revizijah. To je najlažji način, kako povrniti spremembe iz večih revizij v eni potezi.

Izberete pa lahko tudi spajanje izbranih revizij v drugo delovno kopijo, kot je opisano zgoraj.

If all selected revisions have the same author, you can edit the author of all those revisions in one go.

**Slika 4.18. Kontekstni meni spodnjega dela Dnevnika**

Spodnji del dnevnika prav tako vsebuje kontekstni meni, s katerim lahko

Show changes

Prikažete spremembe, ki so bile narejene v izbrani reviziji izbrane datoteke. Ta možnost je na razpologo le za *spremenjene* datoteke.

Blame changes

Okrivite izbrano revizijo in predhodnjo revizijo izbrane datoteke in prikažete razlike z grafičnim orodjem za razlikovanje. Za več informacij preberite [Razdelek 4.23.2, "Okrivi spremembe"](#).

Show as unified diff

Show file changes in unified diff format. This context menu is only available for files shown as *modified*.

Open / Open with...

Odprete izbrano datoteko s privzetim pregledovalnikom ali programom po lastni izbiri.

Blame...

Opens the Blame dialog, allowing you to blame up to the selected revision.

Revert changes from this revision

Povrnte spremembe, narejene na izbrani datoteki izbrane revizije.

Show properties

Pogledate lastnosti Subversion za izbran element.

Show log

Prikažete dnevnik za izbrano datoteko.

Get merge logs

Show the revision log for the selected single file, including merged changes. Find out more in [Razdelek 4.9.6, "Zmožnosti sledenja spajanja"](#).

Save revision to...

Shranite izbrano revizijo datoteke, tako da imate starejšo različico te datoteke.



Namig

Mogoče ste opazili, da uporabljamo izraza razlike in spremembe. Kakšna je razlika?

Številke revizij v sistemu Subversion pomenijo dve stvari. Revizija predstavlja stanje skladišča v nekem trenutku, predstavlja pa tudi seznam sprememb, ki so ustvarile to revizijo. N. pr. "Narejeno v r1234" pomeni, da spremembe, narejene v reviziji 1234, ustvarijo funkcionalnost X. Da ponazorimo, kateri pomen imamo v mislih, uporabljamo dva izraza.

Če izberete revizijo N in M, vam kontekstni meni ponuja možnost prikaza *razlik* med tema dvema revizijama. Ekvivalenten ukaz pri delu z odjemalcem za ukazno vrstico je `diff -r M:N`.

Če izberete samo revizijo N, vam kontekstni meni ponuja prikaz *sprememb*, opravljenih v tej reviziji. Ekvivalentna ukaza pri uporabi odjemalca za ukazno vrstico sta `diff -r N-1:N` in `diff -c N`.

Spodnja polovica prikazuje vse spremenjene datoteke v vseh izbranih revizijah. Kontekstni meni ponuja prikaz *sprememb*.

4.9.4. Pridobivanje dodatnih dnevniških zapisov

Okno Dnevnik ne pokaže vedno vseh sprememb. Razlogi so naslednji:

- V večjih skladiščih se lahko nahaja stotine ali tisoče sprememb. Pridobivanje informacij o vseh spremembah lahko traja dolgo časa. Običajno pa vas zanimajo le zadnje spremembe. Privzeta nastavitve števila dnevniških zapisov za prikaz je 100, vendar lahko to vrednost spremenite v TortoiseSVN → Nastavitve ([Razdelek 4.30.1.2, "Pogovorna okna 1"](#)),
- Če je potrjeno polje **Ustavi ob kopiranju/preimenovanju**, Dnevnik prikaže le revizije do trenutka, ko je bila izbrana datoteka ali mapa prekopirana iz neke druge lokacije v skladišču. To je uporabno, kadar pregledujete veje (ali oznake), saj se prikažejo le spremembe, ki so bile narejene na tej veji.

Običajno boste to možnost pustili izklopljeno. TortoiseSVN si zapomni stanje potrjenih polj, tako da bo upošteval vaše nastavitve.

Kadar je okno **Prikaži dnevnik** prikazano iz okna za spajanje, je to polje vedno potrjeno. Vzrok je v tem, da spajanje večinoma deluje na spremembah na vejah, zato uporaba map pred korenisko mapo v tem primeru nima nobenega smisla.

Upoštevajte, da Subversion izvaja preimenovanje s parom kopirane in izbrisane datoteke, zato bo preimenovanje datoteke ali mape povzročilo, da se v tem primeru dnevniški zapis ustavi, če je ta možnost vklopljena.

Če želite videti več dnevniških zapisov, kliknite na gumb **Naslednjih 100** in prikazalo se bo naslednjih 100 zapisov. Postopek lahko ponovite poljubno mnogokrat.

Zraven tega gumba se nahaja večfunkcijski gumb, ki si zapomni zadnjo akcijo, za katero je bil uporabljen. Kliknite na puščico, če želite videti še ostale akcije, ki so vam na razpolago.

Uporabite gumb **Pokaži obseg...**, če želite videti le določen obseg revizij. V okno boste vnesli začetno in končno revizijo.

Uporabite gumb **Prikaži vse**, če želite videti vse dnevniške zapise od končne revizije HEAD do revizije 1.

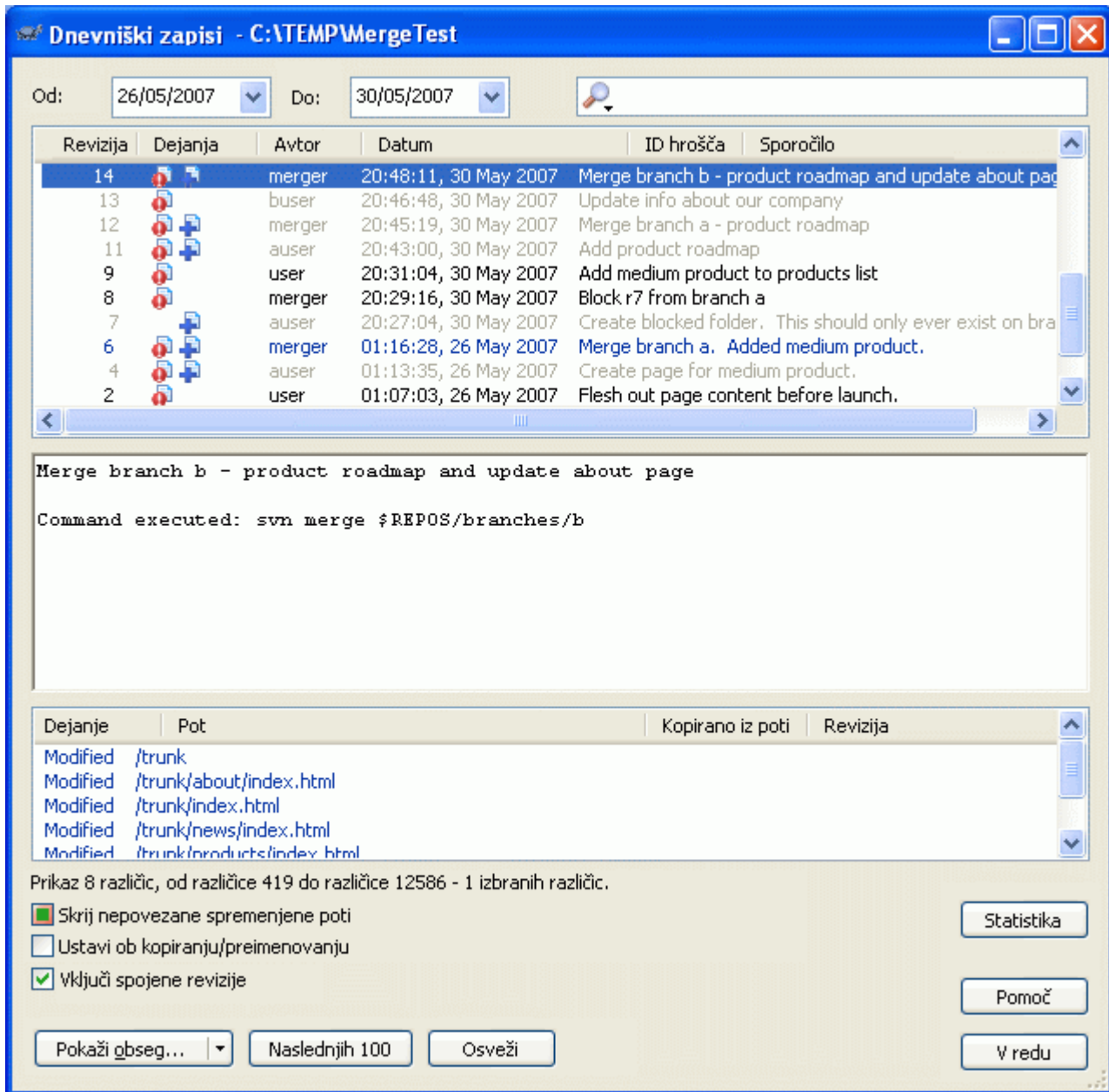
4.9.5. Current Working Copy Revision

Because the log dialog shows you the log from HEAD, not from the current working copy revision, it often happens that there are log messages shown for content which has not yet been updated in your working copy. To help make this clearer, the commit message which corresponds to the revision you have in your working copy is shown in bold.

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

4.9.6. Zmožnosti sledenja spajanja

Subversion 1.5 uvaja sledenje spajanja z uporabo lastnosti. To nam omogoča pridobiti podrobnejšo zgodovino spojenih sprememb. Primer: če na posebni veji razvijate neko novo zmožnost programa in potem spremembe spojite na glavno vejo, boste na glavni veji to novo zmožnost videli kot eno objavo, čeprav ste pri razvoju te lastnosti na posebni veji naredili tisoč objav.



Slika 4.19. Dnevnik prikazuje sledenje spajanja revizij

Če želite videti, katere revizije so bile spojene kot del objave, uporabite potrditveno polje **Vključi spojene revizije**. Dnevniška sporočila se bodo ponovno prenesla, tokrat bodo vključeni tudi zapisi revizij, iz katerih je bilo narejeno spajanje. Spojene revizije so prikazane v sivi barvi, saj predstavljajo spremembe, narejene na drugi veji.

Spajanje seveda nikoli ni enostavno! Med razvojem na veji boste na vejo občasno spajali spremembe iz glavne veje. Torej bo zgodovina spajanja glavne veje vsebovala dodaten nivo zgodovine spajanja. Različni nivoji spajanja so v dnevniku prikazani z zamikanjem.

4.9.7. Spreminjanje sporočila dnevniškega zapisa in avtorja

Revision properties are completely different from the Subversion properties of each item. Revprops are descriptive items which are associated with one specific revision number in the repository, such as log message, commit date and committer name (author).

Včasih želite spremeniti vnešeno sporočilo dnevniškega zapisa - zaradi napake v črkovanju, ker boste želeli izboljšati komentar ali zaradi kakega drugega razloga. Ali pa boste želeli spremeniti avtorja objave, ker ste si nastavili napačno avtentikacijo ali pa...

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/1.5/svn.reposadmin.create.html#svn.reposadmin.create.hooks>] in the Subversion Book for details about how to do that. Read [Razdelek 3.3, "Server side hook scripts"](#) to find some further notes on implementing hooks on a Windows machine.

Once you've set up your server with the required hooks, you can change the author and log message (or any other revprop) of any revision, using the context menu from the top pane of the Log dialog. You can also edit a log message using the context menu for the middle pane.



Pozor

Lastnosti revizij v sistemu Subversion so brez različic, zato bo sprememba lastnosti (n. pr. lastnost `svn:log` za sporočilo objave) *za vedno* prepisala predhodno vrednosti te lastnosti.

4.9.8. Filtriranje dnevniških zapisov

Če želite omejiti izpise na le tiste, ki vas zanimajo, in se tako izogniti brskanju po seznamu, lahko uporabite možnosti filtriranja na vrhu okna. Začetni in končni datum vam omogočata omejitev vpisov na omejeno časovno območje, polje za iskanje pa vam omogoča izpis le tistih revizij, ki vsebujejo iskani niz.

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

To make the filter show all log entries that do *not* match the filter string, start the string with an exclamation mark (!). For example, a filter string `!username` will only show those entries which were not committed by `username`.

Upoštevajte, da filter deluje na sporočilih, ki so že na razpolago. Filter ne nadzira prenosa sporočil iz skladišča.

Poti lahko filtrirate s potrditvijo polja **Skrij nepovezane spremenjene poti**. Povezane poti so tiste, ki vsebujejo pot, ki je bila uporabljena za prikaz Dnevnika. Če pogledte dnevnik za neko mapo, potem to pomeni vsebino te mape. Pri datotekah to pomeni le izbrano datoteko. Potrditveno polje ima tri stanja: prikažete lahko vse poti, posivite nepovezane ali povsem skrijete nepovezane poti.

Včasih delo na projektu zahteva uporabo predpisane oblike sporočila dnevniškega zapisa, kar pomeni, da besedilo, ki opisuje spremembe, ni vidno pri skrajšanem povzetku, ki se prikaže v zgornjem delu okna. Lastnost `tsvn:logsummary` lahko uporabite, da izvečete del besedila, ki se prikaže kot povzetek. Za več informacij o tej lastnosti reberite [Razdelek 4.17.2, "Projektne lastnosti TortoiseSVN"](#).



No Log Formatting from Repository Browser

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.

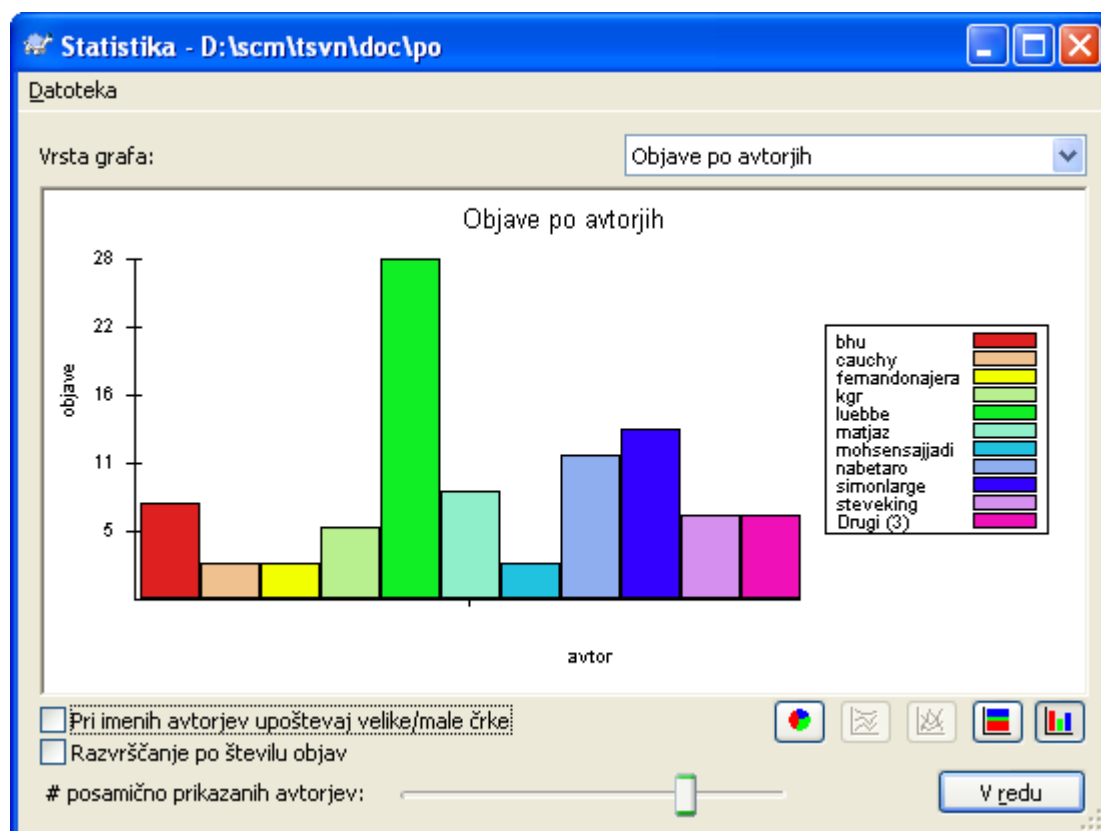
4.9.9. Statistične informacije

Klik na gumb Statistika postreže z nekaj zanimivimi statističnimi informacijami o revizijah, prikazanih v oknu dnevnika. Prikazano je, koliko avtorjev je sodelovalo, koliko objav so naredili, napredek po tednih in še veliko več. Sedaj lahko enostavno vidite, kdo je delal največ in kdo je zabušaval!)

4.9.9.1. Statistika

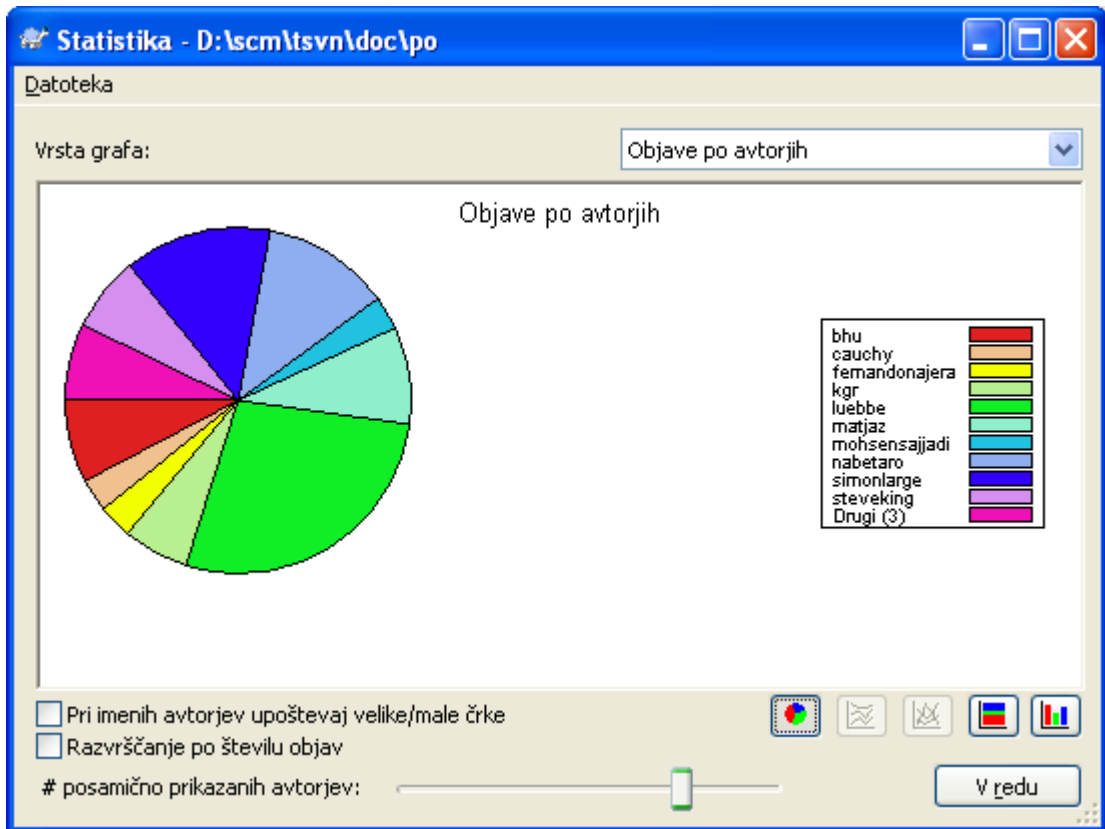
Ta stran vam pokaže vse mogoče številke, predvsem časovno območje in število revizij ter nekatere ekstremne in povprečne vrednosti.

4.9.9.2. Stran objav glede na avtorja



Slika 4.20. Histogram objav glede na avtorja

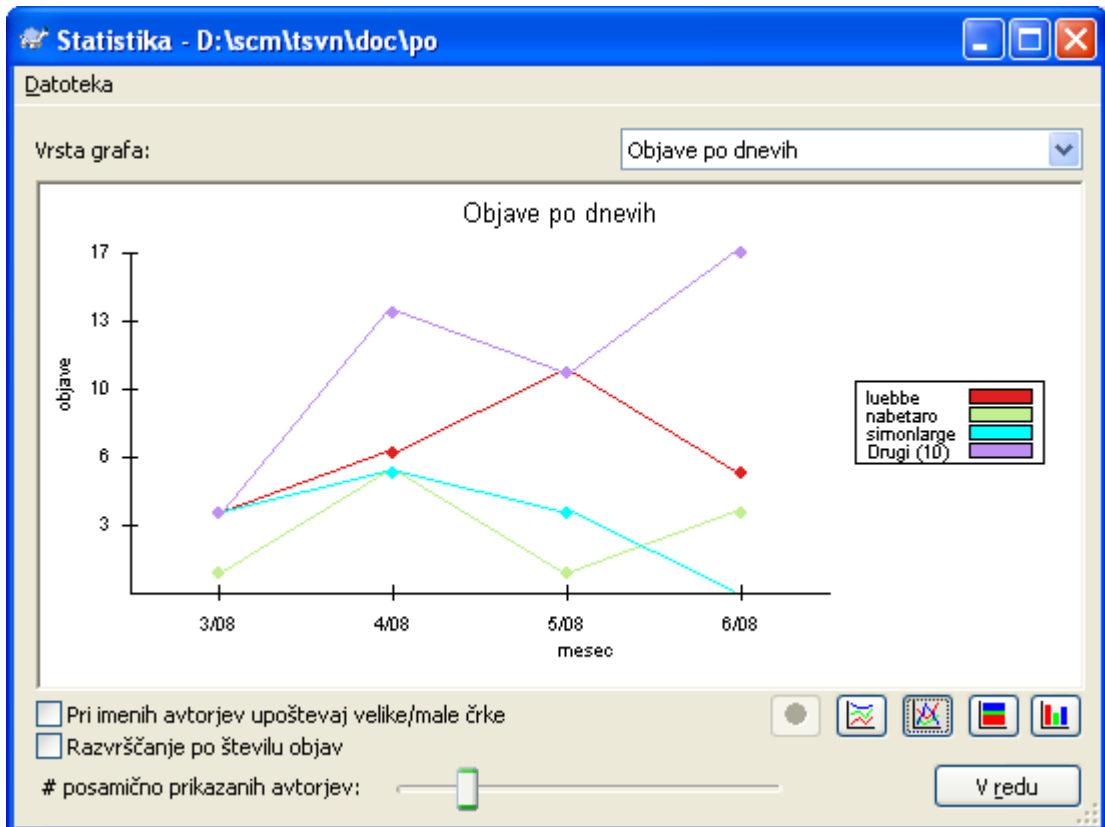
Ta graf prikazuje, kateri avtorji so bili aktivni na projektu v obliki preprostega histograma, zloženega stolpičnega diagrama ali potičnega diagrama.



Slika 4.21. Potični graf objav glede na avtoja

Kadar na projektu sodeluje nekaj glavnih avtorjev in veliko takih, ki prispevajo občasno, veliko majhnih segmentov naredi graf bolj nečitljiv. Drsnik na dnu omogoča nastaviti mejo (v obliki odstotkov). Avtorji, katerih aktivnost je manjša od meje, so združeni v skupino *Ostali*.

4.9.9.3. Objave po datumu



Slika 4.22. Objave po datumu

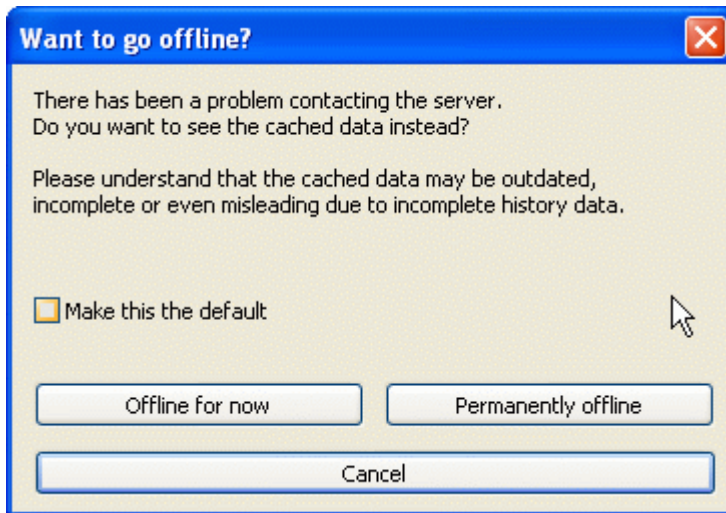
Na tej strani je grafično prikazana projektna aktivnost: število objav *in* avtor. To vam pove, kdaj se na projektu dela in kdo je kdaj delal.

Kadar sodeluje veliko število avtorjev, bo na grafu prikazano veliko število črt. Na voljo sta dva pogleda: *normalen*, kjer je avtorjeva aktivnost relativna na osnovno linijo, in *zložen*, kjer je avtorjeva aktivnost relativna glede na črto pod njim. Ta možnost onemogoča križanje črt, kar olajša branje grafa, vendar je pri tem težje oceniti rezultat posameznega avtorja.

Po privzetih nastavitvah je analiza občutljiva na male in velike črke, torej se uporabnika PeterEgan in PeteRegan obravnavata kot različna avtorja. Vendar pa v veliko primerih uporabniška imena niso občutljiva na male in velike črke in so zato včasih vnešena nekonsistentno. Zato v takšnih primerih želite imeni DavidMorgan in davidmorgan obravnavati kot isto osebo. Za nastavitve tega obnašanja uporabite potrditveno polje **Za avtorje ni razlikovanja velikih/malih črk**.

Statistika pokriva isto časovno območje kot okno dnevnika. Če le-to prikazuje le eno revizijo, potem vam statistika ne pove prav veliko.

4.9.10. Nepovezan način



Slika 4.23. Go Offline Dialog

If the server is not reachable, and you have log caching enabled you can use the log dialog and revision graph in offline mode. This uses data from the cache, which allows you to continue working although the information may not be up-to-date or even complete.

Here you have three options:

Offline for now

Complete the current operation in offline mode, but retry the repository next time log data is requested.

Permanently offline

Remain in offline mode until a repository check is specifically requested. See [Razdelek 4.9.11, "Osveževanje pogleda"](#).

Cancel

If you don't want to continue the operation with possibly stale data, just cancel.

The **Make this the default** checkbox prevents this dialog from re-appearing and always picks the option you choose next. You can still change (or remove) the default after doing this from TortoiseSVN → Settings.

4.9.11. Osveževanje pogleda

Če želite preveriti, ali obstajajo na strežniku novejši zapisi v dnevniku, lahko pogled osvežite z uporabe tipke **F5**. Če uporabljate predpomnilnik dnevnika (privzeta nastavitve), s tem preverite skladišče in prenesete le najnovejše zapise. Če ste pred delali v nepovezanem načinu, bo TortoiseSVN poskušal vzpostaviti povezavo.

Če uporabljate predpomnilnik dnevnika in menite, da se je sporočilo dnevniškega zapisa ali avtor spremenil, lahko za ponoven prenos prikazanih sporočil iz strežnika in posodobitev predpomnilnika uporabite **Shift-F5** ali **Ctrl-F5**. Upoštevajte, da s tem osvežite le trenutno prikazana sporočila in ne celotnega predpomnilnika za trenutno skladišče.

4.10. Pregledovanje razlik

Ena izmed najbolj pogostih zahtev v razvoju projekta je videti spremembe. Mogoče želite videti razlike med dvema revizijama iste datoteke ali razlike med dvema različnima datotekama. TortoiseSVN vsebuje

orodje TortoiseMerge za pregledovanje razlik besedilnih datotek. Za pregledovanje razlik grafičnih datotek TortoiseSVN ponuja TortoiseIDiff. Seveda pa lahko uporabite tudi programe po lastni izbiri.

4.10.1. Spremembe v datoteki

Krajevne spremembe

Če želite videti spremembe, ki ste jih naredili *vi*, uporabite kontekstni meni v Raziskovalcu in izberite TortoiseSVN → Razlikuj.

Razlike v primerjavi z drugo vejo/oznako

Če želite videti, kaj se je spremenilo na glavni veji (če delate na stranski veji) ali na določeni veji (če delate na glavni veji), lahko uporabite kontekstni meni Raziskovalca. Med desnim klikom na datoteko držite pritisnjeno tipko **Shift**. Potem izberite TortoiseSVN → Razlikuj z URL. V okno vnesite naslov URL skladišča, s katerim želite primerjati krajevno datoteko.

Lahko uporabite tudi brskalnik po skladišču in izberete dve drevesi, ki ju želite primerjati, mogoče dve oznaki ali vejo/oznako in glavno vejo. Kontekstni meni vam omogoča primerjanje z ukazom Primerjaj reviziji. Več o tem razloži [Razdelek 4.10.3, "Primerjanje map"](#).

Razlike v primerjavi s prejšnjo revizijo

Če želite videti razlike med določeno revizijo in delovno kopijo, uporabite Dnevnik, izberite revizijo, ki vas zanima, in iz kontekstnega menija izberite Primerjaj z delovno kopijo.

Če želite videti razliko med zadnjo objavljeno revizijo in svojo delovno kopijo, ki ni spremenjena, desno kliknite na datoteko. Nato izberite TortoiseSVN → Razlikuj s prejšno verzijo. Izvedla se bo primerjava med revizijo pred zadnjo objavo (ki je zabeležena v vaši delovni kopiji) in delovno osnovo (BASE). Vidite zadnjo spremembo, narejeno na datoteki, ki je ustvarila vašo delovno kopijo. Spremembe, novejšje kot je vaša delovna kopija, niso prikazane.

Razlike med dvema prejšnjima razlikama

Če želite videti razlike med dvema že objavljenima revizijama, uporabite Dnevnik in izberite reviziji, ki ju želite primerjati (uporabite tipko **Ctrl**). Potem iz kontekstnega menija izberite Primerjaj reviziji.

Če to storite v Dnevniku za mapo, potem pogovorno okno Primerjaj reviziji prikaže seznam spremenjenih datotek v mapi. Več o tem pove [Razdelek 4.10.3, "Primerjanje map"](#).

Vse spremembe, narejeni pri objavi

Če želite videti spremembe na vseh datotekah v določeni reviziji v enem pogledu, lahko izberete obliko poenotena razlika (oblika popravkov GNU). Ta prikaže le razlike skupaj z nekaj bližnjimi vrsticami. Ta oblika je težje berljiva kot vizuelna, vendar lahko vidite vse spremembe skupaj. Iz Dnevnika izberite revizijo in iz kontekstnega menija zaženite ukaz Prikaži razlike kot poenotene razlike.

Razlika med datotekami

Če želite videti razlike med dvema različnima datotekama, lahko to storite neposredno iz Razskovalca, tako da izberete obe datoteki (z uporabo gumba **Ctrl**). Potem iz kontekstnega menija Raziskovalca izberite TortoiseSVN → Razlikuj.

Razlika med datoteko/mapo v delovni kopiji in na naslovu URL

Če želite videti razlike med datoteko v delovni kopiji in datoteko v skladišču Subversion, lahko to storite neposredno iz Raziskovalca. Izberite datoteko, pritisnite tipko **Shift** in desno kliknite, da se pojavi kontekstni meni. Izberite TortoiseSVN → Razlikuj z URL. Enako lahko storite na delovni kopiji. TortoiseMerge prikaže razlike enako kot pri prikazu datoteke popravkov - seznam spremenjenih datotek, ki jih lahko pregledujete eno za drugo.

Razlike z informacijami o krivdi

Če želite poleg razlik videti tudi avtorja, številko revizije in datum sprememb, lahko rezultat razlikovanja in okrivljenja združite. Več o tem si lahko preberete v [Razdelek 4.23.2, "Okrivi spremembe"](#).

Razlika med mapama

Orodja, ki jih dobite zraven programa TortoiseSVN, ne omogočajo pregledovanja razlik med strukturami map. Če imate zunanje orodje, ki to omogoča, ga lahko uporabite. [Razdelek 4.10.5, "Zunanja orodja za razlikovanje/spajanje"](#) pove nekaj o orodjih, ki jih uporabljamo.

Če ste nastavili zunanje orodje za razlikovanje, lahko uporabite **Shift** ob izbiri ukaza Razlikuj in tako poženete zunanje orodje. O nastavljanju zunanjih orodij lahko več preberete v [Razdelek 4.30.5, "Nastavitve zunanjih programov"](#).

4.10.2. Nastavitev zaključkov vrstic in presledkov

Med delom na projektu včasih zamenjate zaključke vrstic - namesto CRLF uporabite LF. Mogoče pa spremenite zamikanje v nekem delu datoteke. S tem žal povzročite, da je veliko vrstic označenih kot spremenjenih, čeprav se vsebina vrstic dejansko ni spremenila. Naslednje možnosti vam omogočajo obvladovanje takšnih sprememb pri primerjanju ali nameščanju popravkov. Nastavitve so na voljo v pogovornih oknih Spoji in Okrivi, prav tako pa tudi v nastavitvah programa TortoiseMerge.

Možnost Prezri zaključke vrstic izloči vse spremembe, ki so posledica sprememb zgolj zaradi različnih zaključkov vrstic.

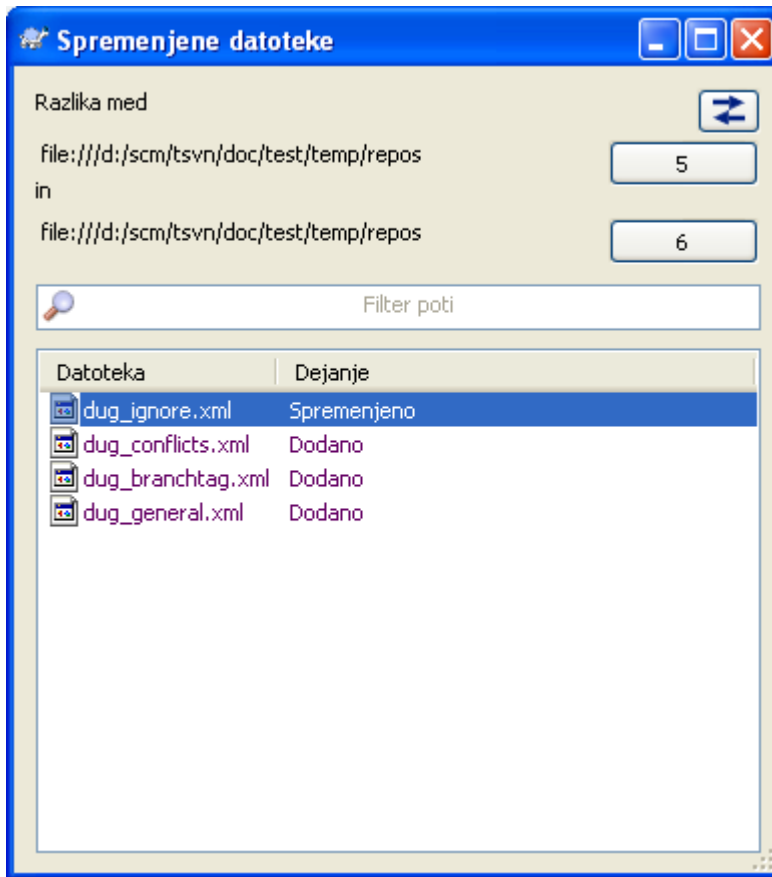
Z možnostjo Primerjaj presledke povzročite, da so vse vrstice, ki vsebuje spremembe presledkov, prikazane kot dodane/odstranjene.

Možnost Prezri razlike presledkov izključi spremembe, ki so posledica spremembe količine ali vrste presledkov, n. pr. spreminjanje zamikanja ali uporaba presledkov namesto tabulatorjev. Dodajanje presledkov, kjer jih prej ni bilo, in popolna odstranitev presledkov se še vedno štejeta kot spremembe.

Možnost Prezri vse presledke izloči vse spremembe, ki so posledica spremembe presledkov.

Seveda so kot spremenjene prikazane vse vrstice s spremenjeno vsebino.

4.10.3. Primerjanje map



Slika 4.24. Okno za primerjanje revizij

Ko v brskalniku po skladišču izberete dve drevesi ali ko izberete dve reviziji mape v dnevniku, lahko uporabite Kontekstni meni → Primerjaj reviziji.

To pogovorno okno prikaže seznam vseh datotek, ki ste jih spremenili, in omogoča posamično primerjavo ali okrivljanje s pomočjo kontekstnega menija.

You can export a *change tree*, which is useful if you need to send someone else your project tree structure, but containing only the files which have changed. This operation works on the selected files only, so you need to select the files of interest - usually that means all of them - and then Context menu → Export selection to.... You will be prompted for a location to save the change tree.

You can also export the *list* of changed files to a text file using Context menu → Save list of selected files to....

If you want to export the list of files *and* the actions (modified, added, deleted) as well, you can do that using Context menu → Copy selection to clipboard.

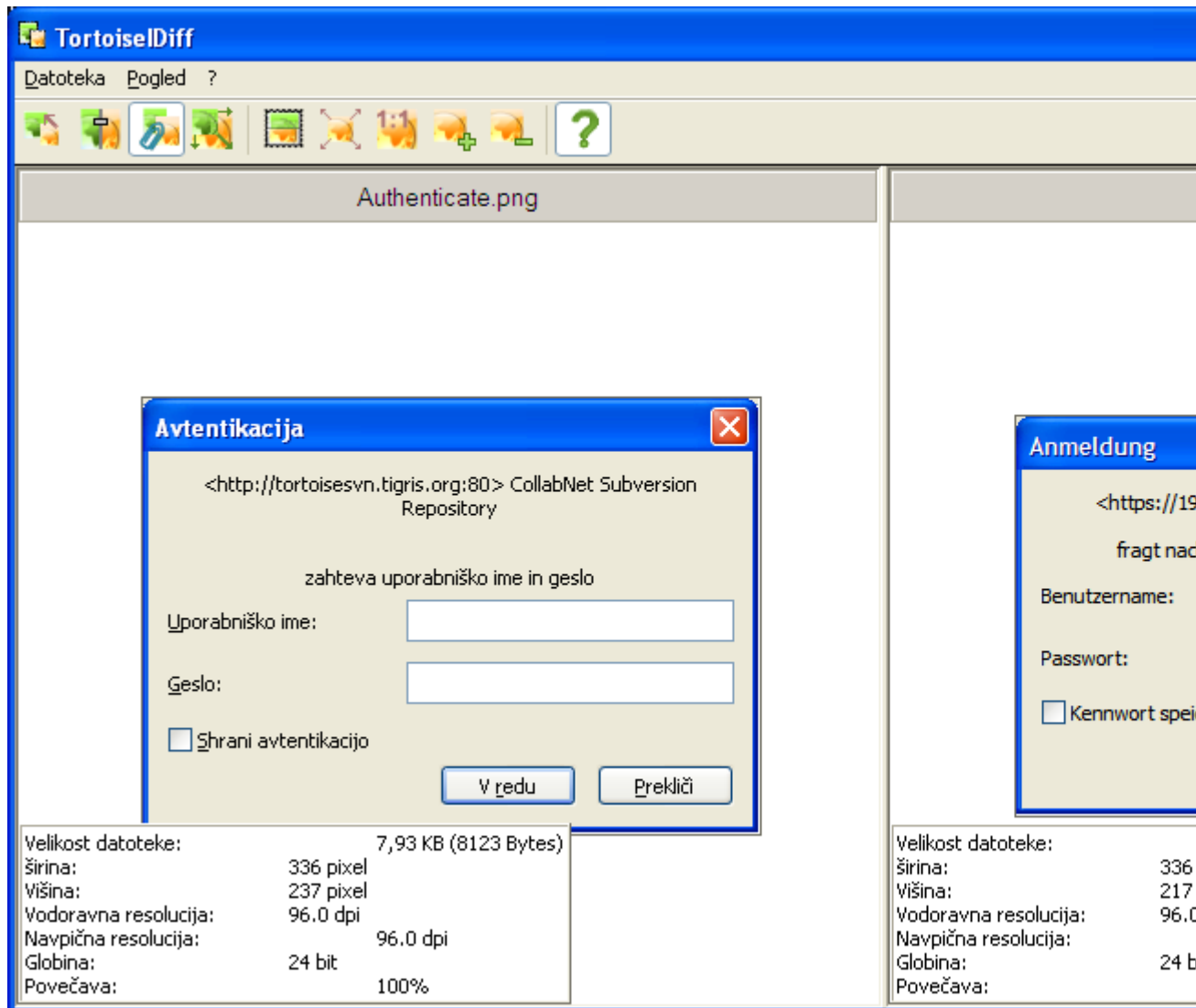
Gumb na vrhu omogoča spremembo smeri primerjave. Lahko prikažete spremembe iz A v B, ali, če vam je lažje, iz B v A.

Gumbi, ki vsebujejo številke revizij, se uporabljajo za spreminjanje območja revizij. Ko območje spremenite, se seznam elementov, ki se razlikujejo med revizijama, samodejno posodobi.

Če je seznam datotek zelo dolg, lahko uporabite polje za iskanje in tako prikažete le datoteke, ki vsebujejo iskane znake. Uporablja se preprosto iskanje, zato za izpis vseh datotek izvorne kode v jeziku C vpišite `.c` in ne `*.c`.

4.10.4. Razlikovanje slik z uporabo programa TortoiseDiff

Obstaja veliko orodij za razlikovanje besedilnih datotek, vključno z našim lastnim orodjem TortoiseMerge. Pogosto pa želimo videti tudi, kako so se spreminjale slike. Zato smo ustvarili orodje TortoiseIDiff.



Slika 4.25. Pregledovalnik razlik med slikami

Uporaba ukaza TortoiseSVN → Razlikuj na datoteki kateregakoli običajnega slikovnega formata zažene program TortoiseIDiff, ki prikaže razlike med slikama. Po privzetih nastavitvah se sliki prikazeta ena ob drugi, v meniju Pogled pa lahko izberete tudi način prikaza ena slika pod drugo ali celo ena slika čez drugo.

Seveda lahko sliko povečate in pomanjšate ter jo premikate. Premikate lahko tudi s potegom z levim miškinim gumbom. Če izberete možnost Poveži slike, so kontrole za premikanje (drsni trak, kolesček na miški) povezane.

Okno Informacije o sliki prikaže podrobnosti o slikovni datoteki, n. pr. velikost v slikovnih točkah, resolucijo in barvno globino. Če vam je okno v napoto, lahko uporabite Pogled → Informacije o sliki. S tem okno skrijete. Iste informacije se prikazujejo v namigu, če z miško mirujete v naslovni vrstici.

Ko se sliki prekrivata, lahko relativno intenzivnost slik (alpha blend) nastavljate z drsnikom na levi strani. Če kliknete kjerkoli na drsniku, boste neposredno nastavili vrednost, lahko pa drsnik potegnete in tako nastavite vrednost interaktivno. Za spreminjanje vrednosti lahko uporabite tudi **Ctrl-Shift-Zasuk** miškega kolesčka.

The button above the slider toggles between 0% and 100% blends, and if you double click the button, the blend toggles automatically every second until you click the button again. This can be useful when looking for multiple small changes.

Včasih želite namesto združene (prekrite) slike raje videti razlike, n. p.r če imate dve reviziji sheme tiskanega vezja in želite izvedeti, katere linije so se spremenile. Če izklopite spajanje (alpha blend) slik, se razlike prikažejo kot *ekskluzivni ALI* barvnih vrednosti točk. Nespremenjena območja bodo bela, spremembe pa bodo obarvane.

4.10.5. Zunanja orodja za razlikovanje/spajanje

Če orodja, ki jih ponujamo v paketu, ne zadostujejo vašim potrebam, lahko poskusite številne odprtokodne ali komercialne programe. Vsakdo ima svoje priljubljeno orodje in ta seznam vsekakor ni popoln, vseeno pa poda nekaj imen aplikacij, ki jih lahko uporabite:

WinMerge

WinMerge [<http://winmerge.sourceforge.net/>] je odlično odprtokodno orodje za razlikovanje, ki zna delati tudi z mapami.

Perforce Merge

Perforce je komercialna različica sistema RCS, vendar lahko njihovo orodje za razlikovanje uporabljate brezplačno. Za več informacij uporabite tole [povezavo](http://www.perforce.com/perforce/products/merge.html) [<http://www.perforce.com/perforce/products/merge.html>].

KDiff3

KDiff3 je brezplačno orodje za razlikovanje, ki zna delati z mapami. Dobite ga [tukaj](http://kdiff3.sf.net/) [<http://kdiff3.sf.net/>].

ExamDiff

ExamDiff Standard je brezplačen program. Zna delati z datotekami, ne pa tudi z mapami. ExamDiff Pro je preizkusni program (shareware) in ima v primerjavi z brezplačno različico veliko dodatkov, vključno z možnostjo razlikovanja in urejanja map. Od verzije 3.2 naprej znata obe različici delati z datotekami tipa unicode. Na voljo sta na strani podjetja *PrestoSoft* [<http://www.prestosoft.com/>].

Beyond Compare

Podobno kot ExamDiff Pro je tudi to odličen preizkusni program (shareware) za razlikovanje, ki zna delati z mapami in datotekami tipa unicode. Dobite ga na strani podjetja *Scooter Software* [<http://www.scootersoftware.com/>].

Araxis Merge

Araxis Merge je uporabno komercialno orodje za razlikovanje in spajanje tako datotek kot map. Uporablja tristransko primerjavo pri spajanju in ima možnost sinhronizacijskih povezav v primeru spremembe vrstnega reda funkcij. Dobite ga na strani podjetja *Araxis* [<http://www.araxis.com/merge/index.html>].

SciTE

Ta urejevalnik besedil uporablja barvno kodiranje za poenotene različice, kar precej olajša branje. Prevezmete ga lahko iz naslova *Scintilla* [<http://www.scintilla.org/SciTEDownload.html>].

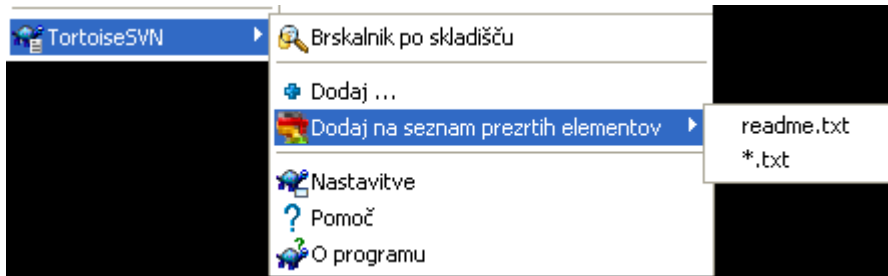
Notepad2

Notepad2 je zasnovan kot zamenjava za standardno beležko operacijskega sistema Windows in je narejen na osnovi odprtokodne kontrole za urejanje besedil Scintilla. Poleg tega, da je zelo primeren

za pregledovanje poenotnih razlik, je za večino operacij precej boljši od Windowsove Beležke. Brezplačno si ga lahko prenesete iz [avtorjeve strani](http://www.flos-freeware.ch/notepad2.html) [http://www.flos-freeware.ch/notepad2.html].

Preberite [Razdelek 4.30.5, "Nastavitve zunanjih programov"](#), če želite izvedeti več o uporabi teh orodij.

4.11. Dodajanje novih datotek in map



Slika 4.26. Kontekstni menu v Raziskovalcu za datoteke brez različic

Če ste med razvojem ustvarili nove datoteke in/ali mape, jih morate dodati v nadzor različic. Izberite datoteke in map in uporabite ukaz TortoiseSVN → Dodaj.

Potem ko ste datoteke/mape dodali v nadzor različic, so označene z ikono *dodano*, kar pomeni, da morate delovno kopijo objaviti, preden so dodane datoteke/mape na razplago ostalim uporabnikom. Zgolj dodajanje datoteke/mape *ne* vpliva na skladišče!



Veliko dodanih elementov

Ukaz Dodaj lahko uporabite tudi na mapi, ki je že pod nadzorom različic. V tem primeru se bodo v pogovornem oknu za dodajanje pojavile vse datoteke brez različic. To vam pomaga, če imate veliko novih datotek, ki jih želite dodati naenkrat.

Če želite dodati datoteke, ki se ne nahajajo v delovni kopiji, uporabite metodo povleci in spusti:

1. izberite datoteke, ki jih želite dodati
2. povlecite jih z desnim miškinim gumbom na novo mesto znotraj delovne kopije
3. sprostite gumb miške
4. izberite Kontekstni meni → SVN - Dodaj datoteke v to delovno kopijo. S tem datoteke prekopirate v delovno kopijo in jih dodate v nadzor različic.

Datoteke, ki se nahajajo v delovni kopiji, lahko dodate tudi tako, da jih povlečete iz Raziskovalca v okno za objavo.

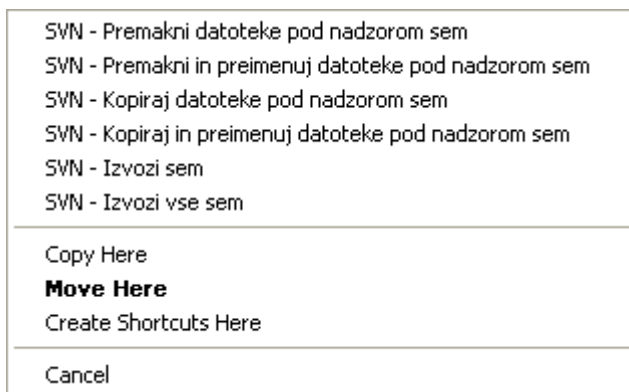
Če datoteko ali mapo dodate po pomoti, lahko dodajanje pred objavo prekličete z ukazom TortoiseSVN → Prekliči dodajanje....

4.12. Kopiranje/premikanje/preimenovanje obstoječih datotek in map

Pogosto se zgodi, da nekatere datoteke že imate v skladišču v okviru drugega projekta in jih želite prekopirati. Lahko jih preprosto prekopirate in dodate v nadzor različic, kot je opisano v prejšnjem

razdelku, vendar v tem primeru izgubite zgodovino datoteke. Če kasneje odkrijete in popravite napako v originalnih datotekah, lahko spremembe samodejno spojite le, če med originalnimi in novimi datotekami obstaja povezava v sistemu Subversion.

Najlažji način kopiranja datotek in map znotraj delovne kopije je z uporabo kontekstnega menija, ki se pojavi v primeru potega z desnim miškinim gumbom. Ko datoteko ali mapo povlečete z desnim miškinim gumbom iz ene delovne kopije v drugo, ali pa celo znotraj iste mape, se ob sprostitvi miškinega gumba prikaže kontekstni meni.



Slika 4.27. Meni ob premikanju mape, ki je pod nadzorom različic

Sedaj lahko obstoječe elemente pod nadzorom prekopirate na drugo lokacijo in jih po potrebi istočasno preimenujete.

You can also copy or move versioned files within a working copy, or between two working copies, using the familiar cut-and-paste method. Use the standard Windows Copy or Cut to copy one or more versioned items to the clipboard. If the clipboard contains such versioned items, you can then use TortoiseSVN → Paste (note: not the standard Windows Paste) to copy or move those items to the new working copy location.

Datoteke in mape lahko prekopirate na drugo lokacijo v skladišču z uporabo ukaza TortoiseSVN → Veja/oznaka. Preberite [Razdelek 4.19.1, “Ustvarjanje veje ali oznake”](#).

Starejšo različico datoteke ali mape lahko poiščete v dnevniku in jo skopirate na novo lokacijo v skladišču z uporabo ukaza Kontekstni meni → Ustvari vejo/oznako iz revizije. Za več informacij preberite [Razdelek 4.9.3, “Pridobivanje dodatnih informacij”](#).

Brskalnik po skladišču pa lahko uporabite tudi za iskanje željene vsebine, ki jo lahko prekopirate v delovno kopijo neposredno iz skladišča ali pa jo kopirate med dvema lokacijama v skladišču. Za več informacij preberite [Razdelek 4.24, “Brskalnik po skladišču”](#).

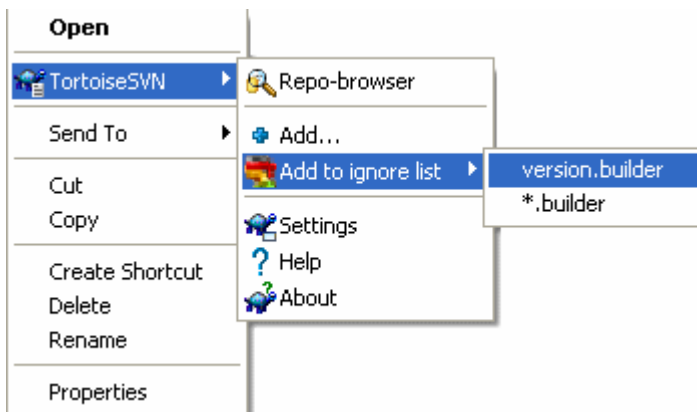


Kopiranje med skladišči ni mogoče

Kopiranje datotek in map *znotraj* skladišča je mogoče, kopiranje ali premikanje elementov med različnimi skladišči pa *ne*. V takšnem primeru obvezno izgubimo zgodovino, tudi če se skladišči nahajata na istem strežniku. Vse, kar lahko naredite, je, da prekopirate vsebino v trenutnem stanju in jo dodate kot novo vsebino v drugo skladišče.

Če niste prepričani, ali dva naslova URL na istem strežniku kažeta na isto skladišče ali na dve različni skladišči, uporabite brskalnik po skladišču. Odprite prvi naslov URL in poiščite njegov koren. Če v enem brskalniku po skladišču vidite obe lokaciji, se obe lokaciji nahajata v istem skladišču.

4.13. Dodajanje datotek in map na seznam prezrtih elementov



Slika 4.28. Kontekstni menu v Raziskovalcu za datoteke brez različic

V večini projektov obstajajo datoteke in mape, ki jih ne dajemo pod nadzor različic. To so datoteke, ki jih naredi prevajalnik (*.obj, *.lst), mape, kamor se shranjujejo izvršne datoteke. Ko objavljate spremembe, TortoiseSVN prikaže datoteke brez različic, kar napolni seznam v oknu za objave. Seveda lahko prikaz datotek brez različic izklopite, vendar lahko to povzroči, da pozabite dodati nove izvorne datoteke v skladišče.

Tem problemom se najlažje izognete, če dodate generirane datoteke na projektni seznam prezrtih elementov. Na ta način se te datoteke ne bodo pojavile v oknu za objave, medtem ko boste tam še vedno videli izvorne datoteke brez različic.

Če desno kliknete na posamezno datoteko brez različic in iz kontekstnega menija izberete ukaz TortoiseSVN → Dodaj na seznam prezrtih, se pojavi podmeni, ki vam omogoča, da izberete le izbrano datoteko ali pa vse datoteke z isto končnico. Če izberete več datotek, se podmeni ne pojavi in lahko dodate na seznam prezrtih elementov le izbrane datoteke.

Če želite iz seznama prezrtih odstraniti enega ali več elementov, desno kliknite na elemente in izberite TortoiseSVN → Odstrani s seznama prezrtih. Dostopate pa lahko tudi neposredno do lastnosti `svn:ignore` določene mape. To vam omogoča, da določite bolj splošne vzorce za preziranje, kar je opisano spodaj. Za neposredno nastavljanje lastnosti preberite [Razdelek 4.17, "Nastavitve projekta"](#). Upoštevajte, da je vsak vzorec za prezrte datoteke potrebno vpisati v svojo vrstico. Ločevanje s presledki ne deluje.



Splošni seznam prezrtih elementov

Drug način, kako prezreti datoteke, je dodajanje na *splošni seznam prezrtih elementov*. Razlika je v tem, da je splošen seznam lastnost odjemalca. Velja za vse projekte v sistemu Subversion, vendar le na računalniku, kjer je nameščen odjemalec. Kjer je to le mogoče, je v splošnem bolje uporabiti lastnost `svn:ignore`, ker jo lahko nastavimo na določene dele projekta, poleg tega pa velja za vse uporabnike, ki prevzamejo projekt. Za več informacij preberite [Razdelek 4.30.1, "Splošne nastavitve"](#).



Dodajanje elementov pod nadzorom različic na seznam prezrtih

Datotek in map pod nadzorom različic ni mogoče prezreti. Če ste dali datoteko pod nadzor različic po pomoti, preberite [Razdelek B.8, "Kako dodam na seznam prezrtih datoteke, ki so že pod nadzorom"](#), kjer je opisano, kako jo iz sistema nadzora "odstranite".

4.13.1. Iskanje vzorcev v seznamu prezrtih elementov

Iskanje vzorcev v sistemu Subversion uporablja tehniko, ki izvira iz sistema Unix in se uporablja za določanje datotek z uporabo meta-znakov. Naslednji znaki imajo poseben pomen:

*

Poišče katerikoli niz znakov ali prazen niz (brez znakov).

?

Poišče katerikoli znak.

[...]

Poišče katerikoli znak, ki je podan v oglatem oklepaju. Par znakov znotraj oklepajev, ločen z “-”, pomeni katerikoli znak med tema dvema znakoma. Primer: `[AGm-p]` išče kateregakoli izmed naslednjih znakov: A, G, m, n, o ali p.

Iskanje vzorcev je občutljivo na velike in male črke, kar lahko v sistemu Windows povzroča težave. Neobčutljivost lahko vsilite s podvajanjem znakov. Primer: za prezrtje `*.tmp` ne glede na uporabo velikih in malih črk uporabite vzorec `*.[Tt][Mm][Pp]`.

Če vas zanima uradna definicija iskanja vzorcev v datotekah, jo najdete v Specifikaciji IEEE za ukaze lupine *Pattern Matching Notation* [http://www.opengroup.org/onlinepubs/009695399/utilities/xcu_chap02.html#tag_02_13].

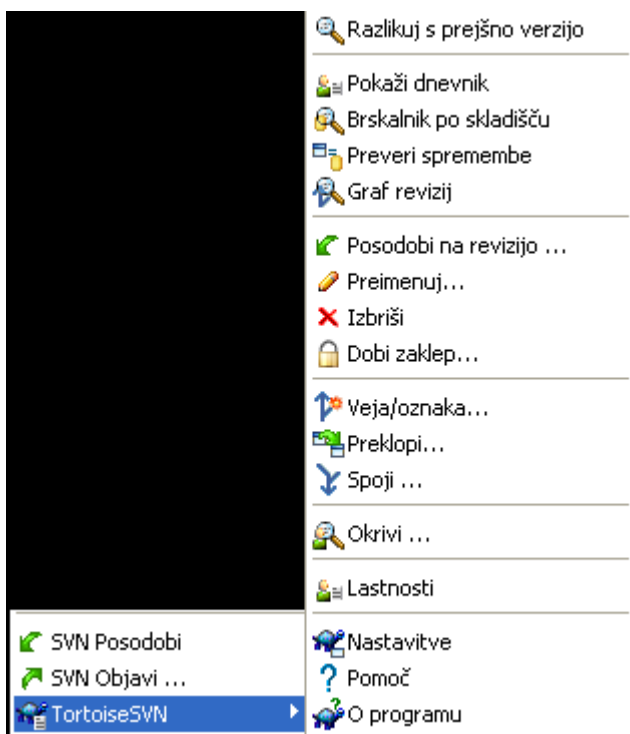


No Paths in Global Ignore List

You should not include path information in your pattern. The pattern matching is intended to be used against plain file names and folder names. If you want to ignore all CVS folders, just add CVS to the ignore list. There is no need to specify `CVS */CVS` as you did in earlier versions. If you want to ignore all tmp folders when they exist within a prog folder but not within a doc folder you should use the `svn:ignore` property instead. There is no reliable way to achieve this using global ignore patterns.

4.14. Brisanje, preimenovanje in premikanje

V nasprotju s sistemom CVS Subversion dovoljuje preimenovanje in premikanje datotek in map. Zato se v podmeniju TortoiseSVN nahajajo ukazi za brisanje in preimenovanje.



Slika 4.29. Kontekstni meni v Raziskovalcu za datotek pod nadzorom različic

4.14.1. Branje datotek in map

Za brisanje datotek ali map iz sistema Subversion uporabite TortoiseSVN → Izbriši.

Ko uporabite ukaz TortoiseSVN → Izbriši na datoteki, se ta takoj izbriše iz delovne kopije, prav tako pa se označi za brisanje v skladišču. Branje se bo zgodilo ob naslednji objavi. Nadrejena mapa datoteke ima ikono “izbrisano”. Dokler sprememb ne objavite, lahko izbrisano datoteko pridobite nazaj z uporabo ukaza TortoiseSVN → Povrni, ki ga izvedete na nadrejeni mapi.

Ko uporabite ukaz TortoiseSVN → Izbriši na mapi, ta ostane v delovni kopiji, njena ikona pa se spremeni in nakazuje, da je mapa označena za brisanje. Dokler sprememb ne objavite, lahko mapo pridobite nazaj z uporabo ukaza TortoiseSVN → Povrni na mapi sami. Vzrok za razliko pri obnašanju je v delovnjaju sistema Subversion, ne sistema TortoiseSVN.

Če želite izbrisati element iz skladišča, vendar ga želite obdržati krajevno kot datoteko/mapo brez različic, uporabite ukaz Razširjeni kontekstni meni → Izbriši (ohrani krajevno). Za prikaz razširjenega kontekstnega menija morate držati tipko **Shift** ob desnem kliku na element v Raziskovalcu.

Če *datoteko* izbrišete iz Raziskovalca, namesto da bi uporabili ukaz iz kontekstnega menija TortoiseSVN, vam okno za objavo to datoteko prikaže in vam omogoča, da jo pred objavo odstranite tudi iz sistema različic. Če pa delovno kopijo posodobite, bo Subversion manjkajočo krajevno datoteko nadomestil z zadnjo različico te datoteke v skladišču. Če želite izbrisati datoteko pod nadzorom različic, vedno uporabite ukaz TortoiseSVN → Izbriši, da sistemu Subversion ne bo potrebno ugibati, kaj pravzaprav želite.

Če *mapo* izbrišete iz Raziskovalca, namesto da bi uporabili ukaz iz kontekstnega menija TortoiseSVN, bo vaša delovna kopija pokvarjena in sprememb ne boste mogli objavljati. Če delovno kopijo posodobite, bo Subversion manjkajočo mapo nadomestil z najnovejšo različico mape iz skladišča. Potem jo lahko izbrišete na pravilen način z ukazom TortoiseSVN → Izbriši.



Kako dobiti nazaj izbrisane datoteke ali mape

Če ste datoteko ali mapo izbrisali in spremembe že objavili, potem ne morete uporabiti ukaza TortoiseSVN → Povrni, da bi jo dobili nazaj. Datoteka ali mapa pa seveda ni izgubljena. Če poznate številko revizije, v kateri je bila datoteka ali mapa izbrisana (če je ne poznate, jo poiščite v Dnevniku), odprite brskalnik po skladišču in preklopite na to revizijo. Potem izberite datoteko ali mapo in izvedite Kontekstni meni → Kopiraj v..., kot cilj kopiranja pa izberite pot v delovni kopiji.

4.14.2. Premikanje datotek in map

Če želite datoteko preimenovati, uporabite Kontekstni meni → Preimenuj... Vpišite novo ime za element in preimenovanje je opravljeno.

Če želite datoteke premakniti znotraj delovne kopije, naprimer v drugo podmapo, uporabite poteg z desnim miškinim gumbom:

1. izberite datoteke ali mape, ki jih želite premakniti
2. povlecite jih z desnim miškinim gumbom na novo mesto znotraj delovne kopije
3. sprostite gumb miške
4. v meniju izberite Kontekstni meni → SVN - Premakni datoteke pod nadzorom sem



Objavi nadrejeno mapo

Ker je preimenovanje izvedeno kot zaporedni operaciji brisanja in dodajanja, morate narediti objavo na nadrejeni mapi datoteke, da se bo tudi izbrisana datoteka prikazala v oknu za objavo. Če izbrisane datoteke ne objavite, bo le-ta ostala v skladišču in ne bo odstanjena iz delovnih kopij vaših sodelavcev, ko bodo ti posodabljali mapo. V delovni kopiji bodo *obe* datoteki - tako novo kot staro.

Preimenovanje mape *morate* objaviti preden začnete spreminjati njeno vsebino, v nasprotnem primeru lahko pride do okvare delovne kopije.

Za premikanje elementov lahko uporabite tudi brskalnik po skladišču. Za več informacij preberite [Razdelek 4.24, "Brskalnik po skladišču"](#).



Zunanjih elementov ne premikajte z ukazi Subversion

Ukazov Premakni in Preimenuj *ne* uporabljajte na mapah, ki imajo nastavljeno lastnost `svn:externals`. To bi povzročilo izbris zunanjega elementa iz nadrejene mape, kar bi najbrž razjezilo precej uporabnikov. Če želite premakniti zunanjo mapo, uporabite premikanje v Raziskovalcu, nato pa uskladite lastnosti `svn:externals` nadrejenih map vira in cilja.

4.14.3. Spreminjanje velikih in malih črk v imenih datotek

Spreminjanje imena, ki se od starega razlikuje le v velikih in malih črkah, je na sistemu Windows malce nerodno. Med preimenovanjem morata namreč za kratek čas obstajati obe datoteki. Ker pa datotečni sistema operacijskega sistema Windows ne loči med velikimi in malimi črkami, ukaza Preimenuj ne morete uporabiti.

Na srečo obstajata (vsaj) dve možni rešitvi, kako preimenovati datoteko brez izgube zgodovine. Pomembno je, da jo preimenujete v sistemu Subversion. Preimenovanje v Raziskovalcu pokvari delovno kopijo!

Rešitev A) (priporočena)

1. Objavite spremembe v delovni kopiji.
2. Datoteko preimenujete (zamenjajte ustrezne velike in male črk) neposredno v skladišču s pomočjo brskalnika po skladišču.
3. Posodobite delovno kopijo

Rešitev B)

1. Preimenujte datoteko, tako da uporabite ukaz Preimenuj iz kontekstnega menija TortoiseSVN. Novo ime naj bo enako staremu, dodajte mu le znak _ na koncu.
2. Objavite spremembe
3. Preimenujte datoteko (zamenjajte ustrezne velike in male črke).
4. Objavite spremembe

4.14.4. Obravnava težav pri imenih datotek zaradi velikih in malih črk

Če se v skladišču že nahajata dve datoteki, ki se razlikujeta samo v velikih in malih črkah (n. pr. TEST.TXT in test.txt), mape, kjer se ti dve datoteki nahajata, na sistemu Windows ne morete posodobiti ali prevzeti. Subversion namreč omogoča ločevanje imen, ki se razlikujejo samo v velikih in malih črkah, sistem Windows pa ne.

Ta situacija je možna, če dva uporabnika iz dveh različnih delovnih kopij objavita datoteko, ki se razlikuje le v velikih in malih črkah. Možna pa je tudi takrat, ko uporabniki uporabljajo operacijski sistem, ki v datotečnem sistemu ločuje velike in male črke, n. pr. Linux.

V takšnem primeru se morate odločiti, katero izmed dveh datotek boste obdržali, drugo pa morate v skladišču izbrisati ali preimenovati.



Kako preprečiti dve datoteki z enakima imenoma

Na naslovu <http://svn.collab.net/repos/svn/trunk/contrib/hook-scripts/> je na razpolago ukazna datoteka akcije, ki prepreči objave, ki vodijo do spora zaradi velikih in malih črk.

4.14.5. Popravljanje preimenovanj datotek

Včasih vaše prijazno razvojno orodje preimenuje datoteke kot del refaktoriranja in seveda tega ne pove sistemu Subversion. Če poskusite takšne spremembe objaviti, vidite staro datoteko kot manjkajočo in novo kot datoteko brez različic. Lahko bi enostavno dodali datoteko brez različic, vendar bi v tem primeru izgubili zgodovino, saj Subversion ne ve, da sta datoteki povezani.

Boljši način je, da sistemu Subversion poveste, da gre v tem primeru dejansko za preimenovanje datoteke in sicer lahko to storite v oknih Objavi in Preveri posodobitve. Enostavno izberite tako staro datoteko (manjkajočo) kot novo (brez različic) in uporabite Kontekstni meni → Popravi premik. S tem datoteki povežete v preimenovanje.

4.14.6. Branje datotek brez različic

Običajno nastavite sezname prezrtih elementov v sistemu v Subversion tako, da se prezrejo vse generirane datoteke. Kaj pa, če želite počistiti vse prezrte datoteke, da bi naredili čisto gradnjo? Običajno naredite

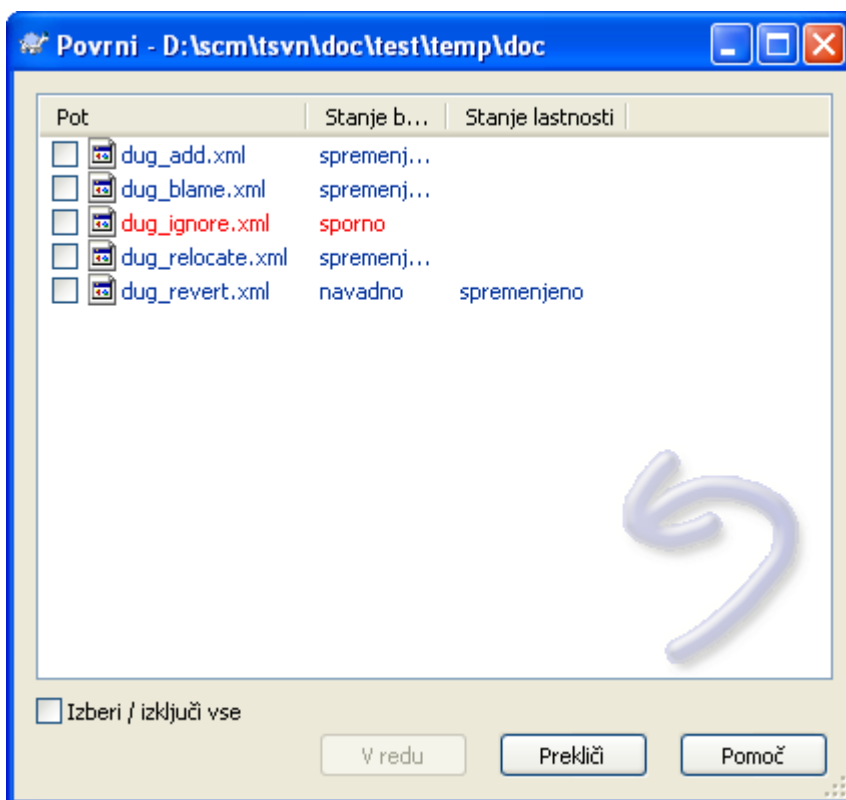
to v skripti makefile, če pa to skripto razhroščujete ali delate spremembe v sistemu gradnje, je koristno, če lahko delovno kopijo počistite.

TortoiseSVN vam ponuja to možnost z uporabo ukaza **Razširjeni kontekstni meni** → **Izbriši datoteke brez različic**.... Da vidite razširjeni kontekstni meni, med desnim klikom na mapo v Raziskovalcu držite pritisnjeno tipko **Shift**. Prikazalo se bo pogovorno okno s seznamom vseh datotek brez različic v delovni kopiji. Izberete lahko tiste, ki jih želite izbrisati.

Pri brisanju elementov se le-te premakne v Koš, tako da jih lahko v primeru napake še vedno pridobite nazaj.

4.15. Razveljavljanje sprememb

Če želite v datoteki povrniti vse spremembe, ki ste jih naredili od zadnje posodobitve, desno kliknite, da se pojavi kontekstni meni in izberite ukaz **TortoiseSVN** → **Povrni**. Pojavi se pogovorno okno s seznamom vseh spremenjenih datotek, ki jih lahko povrnete. Izberite tiste, ki jih želite povrniti in kliknite na gumb **V redu**.



Slika 4.30. Okno za povrnitev

Če želite preklicati brisanje ali preimenovanje, morate ukaz **Prekliči** uporabiti na nadrejeni mapi, saj izbrisane datoteke na obstajajo in tako ne morete desno klikniti nanje.

Če želite preklicati dodajanje elementa, lahko iz kontekstnega menija izberete ukaz **TortoiseSVN** → **Prekliči dodajanje**.... Dejansko se izvede ukaz **Povrni**, ki pa smo ga preimenovali, da je njegovo dejanje bolj očitno.

Stolpce v tem oknu lahko prilagodite enako kot stolpce v oknu **Preveri spremembe**. Za več informacij preberite [Razdelek 4.7.3, "Krajevno in oddaljeno stanje"](#).



Povrnitev sprememb, ki so že bile objavljene

Z ukazom Povrni lahko povrnete le spremembe, ki jih še niste objavili, *ne* pa tudi sprememb, ki ste jih že objavili. Če želite povrniti vse spremembe, ki ste jih objavili v določeni reviziji, preberite [Razdelek 4.9, "Pogovorno okno Dnevnik"](#).



Povrnitev je počasna

Povrnitev krajevnih sprememb lahko traja dalj časa, kot mogoče pričakujete. To se zgodi, ker TortoiseSVN spremenjeno različico datoteke pošlje v Koš, tako da jo lahko pridobite, če ste ukaz Povrni izvedli po pomoti. Če je Koš poln, potrebuje operacijski sistem več časa, da najde prostor, kamor datoteko shrani. Rešitev je enostavna: izpraznite Koš ali pa v nastavitvah izklopite polje Pri uporabi ukaza Povrni uporabi koš.

4.16. Čiščenje

Če se nek ukaz sistema Subversion ne more izvesti uspešno, denimo zaradi težav s strežnikom, lahko vaša delovna kopija ostane v nekonsistentnem stanju. V tem primeru morate na mapi uporabiti ukaz TortoiseSVN → Počisti. Najbolje, da to storite na vrhnji mapi delovne kopije.

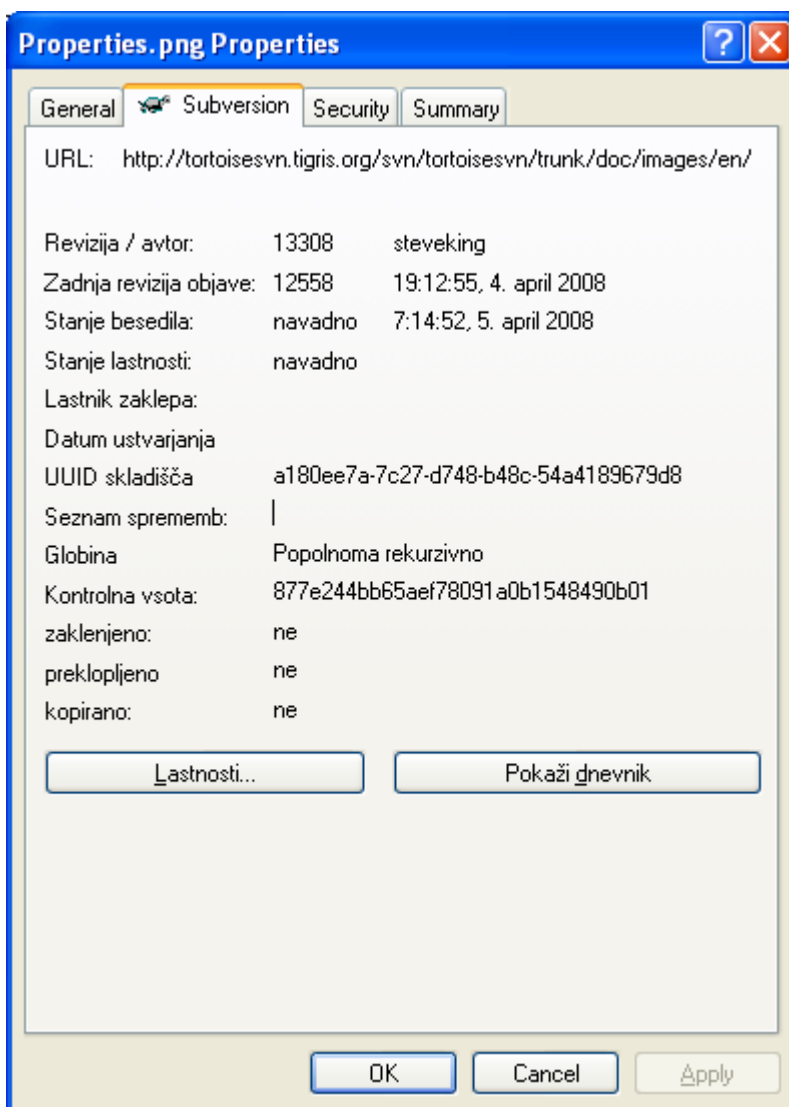
Čiščenje ima še en uporaben stranski učinek. Če se datum datoteke spremeni, njena vsebina pa ne, lahko Subversion ugotovi, ali gre za dejansko spremembo vsebine le s popolno primerjavo trenutne datoteke z začetno datoteko. Če imate v delovni kopiji veliko datotek s spremenjenim datumom, lahko pridobivanje dejanskega stanja datotek poteka zelo dolgo, kar upočasni odzivnost pogovornih oken. Če na delovni kopiji izvedete ukaz Počisti, bo le-ta popravil datoteke z nepravilnim časovnim žigom in s tem povečal hitrost pridobivanja stanja datotek.



Uporabi časovni žig objave

Nekatere starejše različice sistema Subversion so vsebovale napako, ki je povzročila nepravilne časovne žige datotek v primeru uporabe možnosti Uporabi časovni žig objave. Uporabite ukaz Počisti, da pohitrite odzivnost takšne delovne kopije.

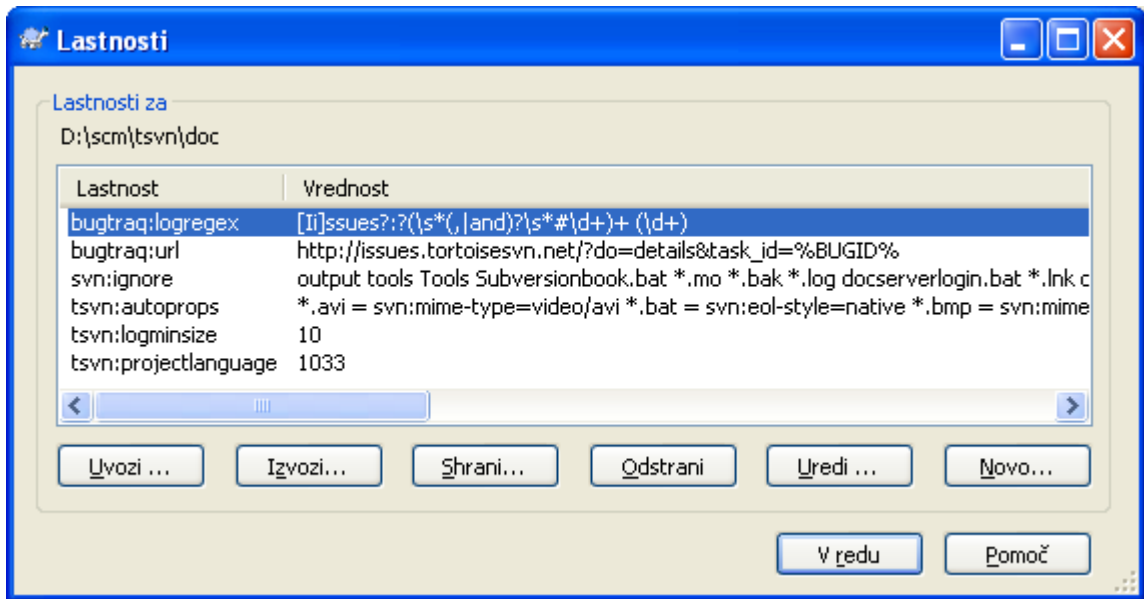
4.17. Nastavitve projekta



Slika 4.31. Stran Lastnosti v Raziskovalcu, zavihek Subversion

Včasih hočete o datoteki/mapi izvedeti več, kot pove prekrivna ikona. Vse informacije, ki jih Subversion poda, lahko najdete v oknu Lastnosti. V Raziskovalcu izberite datoteko in v kontekstnem meniju izberite **Meni Windows** → **Lastnosti** (pazite: to je običajno okno Lastnosti, ki ga prikaže Raziskovalec, ne tisti v podmeniju TortoiseSVN!). V to okno za datoteke/mape pod nadzorom različic TortoiseSVN doda zavihek Subversion, kjer dobite vse potrebne informacije o izbrani datoteki/mapi.

4.17.1. Lastnosti Subversion



Slika 4.32. Lastnosti v sistemu Subversion

Lastnosti Subversion lahko preberete in nastavite v oknu Lastnosti, ki ga prikaže sistem Windows. Poleg tega vidite te lastnosti tudi, če uporabite ukaz TortoiseSVN → Lastnosti. V pogovornih oknih TortoiseSVN, kjer se izpišejo sezname datotek, lahko lastnosti prikažete z ukazom Kontekstni meni → Lastnosti.

You can add your own properties, or some properties with a special meaning in Subversion. These begin with `svn:`. `svn:externals` is such a property; see how to handle externals in [Razdelek 4.18, "External Items"](#).

4.17.1.1. `svn:keywords`

Subversion omogoča (podobno kot sistem CVS) razširitev ključnih besed, ki se lahko uporabljajo za vstavljanje imena datoteke in številke revizije v vsebino datoteke. Trenutno so podprte naslednje ključne besede:

`$Date$`

Datum zadnje znane objave glede na informacije iz zadnje posodobitve delovne kopije. *Ne* preverja, če so v skladišču novejša spremembe.

`$Revision$`

Revizija zadnje znane objave.

`$Author$`

Avtor zadnje znane objave.

`$HeadURL$`

Celoten naslov URL te datoteke v skladišču.

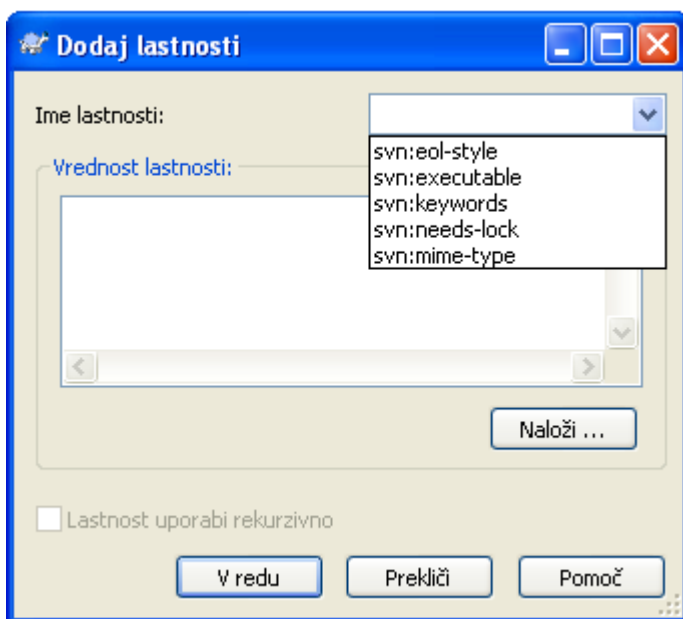
`Id`

Stisnjena kombinacija prejšnjih štirih ključnih besed.

To find out how to use these keywords, look at the [svn:keywords section](http://svnbook.red-bean.com/en/1.5/svn.advanced.props.special.keywords.html) [http://svnbook.red-bean.com/en/1.5/svn.advanced.props.special.keywords.html] 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/1.5/svn.advanced.props.html>].

4.17.1.2. Dodajanje in urejanje lastnosti



Slika 4.33. Dodajanje lastnosti

Če želite dodati novo lastnost, najprej kliknite na gumb **Dodaj...** Iz spustnega menija izberite zahtevano lastnost ali vpišite ime svoje lastnosti, nato pa v vnosno polje vpišite vrednost. Pri lastnostih, ki dopuščajo več vrednosti, n. pr. seznam prezrtih elementov, vnesite po eno vrednost v vrstico. S klikom na gumb **V redu** dodate lastnost na seznam.

Če želite lastnost nastaviti več elementom naenkrat, v Raziskovalcu izberite datoteke/mape, nato pa pokličite ukaz **Kontekstni meni** → **Lastnosti**

Če želite nastaviti lastnost *vsaki* datoteki in mapi v hierarhiji pod trenutno mapo, uporabite potrditveno polje **Rekurzivno**.

Nekatere lastnosti, n. pr. `svn:needs-lock`, lahko nastavite le na datotekah, tako da se ime lastnosti ne pojavi v spustnih menijih pri mapah. Še vedno lahko takšno lastnost nastavite rekurzivno na vseh datotekah hierarhije, vendar morate ime lastnosti vpisati sami.

Če želite urediti obstoječo lastnost, jo izberite iz seznama obstoječih lastnosti in kliknite na gumb **Uredi...**

Če želite odstraniti nastavljen lastnost, jo izberite iz seznama obstoječih lastnosti in kliknite na gumb **Odstrani**.

The `svn:externals` property can be used to pull in other projects from the same repository or a completely different repository. For more information, read [Razdelek 4.18, "External Items"](#).

4.17.1.3. Izvažanje in uvažanje lastnosti

Pogosto isti nabor lastnosti nastavite na več datotekah/mapah, naprimer `bugtraq:logregex`. Za enostaven prenos lastnosti iz enega projekta na drugega lahko uporabite zmožnost **Izvoz/Uvoz**.

Na datoteki ali mapi, ki vsebuje nastavljene lastnosti, uporabite **TortoiseSVN** → **Lastnosti**, izberite lastnosti, ki jih želite izvoziti in kliknite na gumb **Izvozi...** Vpisati boste morali ime datoteke, v katero naj se lastnosti shranijo.

Na mapah, na katerih želite nastaviti lastnosti, uporabite TortoiseSVN → Lastnosti in kliknite na gumb **Uvozi...** Izberite datoteko, iz katere želite uvoziti lastnosti, zato izberite datoteko, ki ste jo ustvarili pri izvažanju. Lastnosti bodo v mapo dodane nerekurzivno.

Če želite drevesu dodati lastnosti rekurzivno, naredite zgoraj opisane korake, potem pa v oknu lastnosti izberite vsako lastnost, kliknite na gumb **Uredi...**, potrdite polje **Uporabi lastnost rekurzivno** in kliknite na gumb **V redu**.

Izvozna datoteka je binarna in lastna sistemu TortoiseSVN. Njena edina uporaba je prenos lastnosti z uporabo ukazov **Izvozi** in **Uvozi**, tako da teh datotek ne urejamo.

4.17.1.4. Binarne lastnosti

TortoiseSVN lahko z uporabo datotek dodaja lastnostim dvojiške vrednosti. Če želite dvojiško lastnost prebrati, jo shranite v datoteko z gumbom **Shrani...** Če želite dvojiško lastnost nastaviti, uporabite urejevalnik datotek ali drugo primerno orodje, ki vam omogoča ustvariti datoteko z željeno vsebino, potem pa datoteko naložite s klikom na gumb **Naloži...**

Čeprav se dvojiške lastnosti ne uporabljajo prav pogosto, pa so včasih precej uporabne. Primer: če shranjujete obsežne grafične datoteke ali če je aplikacija, ki slike nalaga, obsežna, lahko pomanjšane slike shranite kot lastnosti datotek in jih tako hitreje pregledujete.

4.17.1.5. Samodejna nastavitve lastnosti

Subversion in TortoiseSVN lahko prilagodite tako, da se lastnosti samodejno nastavijo datotekam in mapam, ko se te dodajo v skladišče. To lahko storite na dva načina.

To zmožnost lahko vključite neposredno v konfiguracijski datoteki za Subversion. Na strani **Splošno** v nastavitvah se nahaja gumb, ki odpre konfiguracijsko datoteko v urejevalniku besedil. Konfiguracijska datoteka je besedilna datoteka, s katero nastavite določene zmožnosti programa Subversion. Spremeniti morate dve stvari: najprej v odseku `miscellany` izbršite komentar pred vrstico `enable-autoprops = yes`. Nato v naslednjem odseku definirajte, katere lastnosti naj se pripnejo določenim vrstam datotek. S tem določite nastavitve, ki veljajo za vse odjemalce Subversion, vendar jih je potrebno nastaviti na vsakem odjemalcu posebej - možnost, da jih nastavite v skladišču, od koder jih odjemalci preberejo, ne obstaja.

Druga možnost je nastavitve lastnosti `tsvn:autoprops` na mapah, kot je opisano v naslednjem odseku. Ta metoda deluje le pri odjemalcih TortoiseSVN, vendar se pri posodobitvi prenese na vse delovne kopije.

Ne glede na uporabljen metodo upoštevajte, da se samodejne lastnosti upoštevajo le pri dodajanju datotek v skladišče. Te lastnosti nikoli ne spreminjajo lastnosti datotek, ki so že pod nadzorom različic.

Če želite stodontno zagotoviti, da imajo datoteke ob dodajanju v skladišče nastavljen ustrezne lastnosti, namestite v skladišče akcijske skripte, ki se izvedejo pred objavo (`pre-hook script`).



Objavite lastnosti

Lastnosti v sistemu Subversion so pod nadzorom različic. Potem, ko spremenite ali dodate lastnosti, morate spremembe objaviti.



Spori pri lastnostih

Če pri objavi sprememb pride do spora, ker je nek drug uporabnik spremenil isto lastnost, Subversion ustvari datoteko `.prej`. Ko rešite spor, to datoteko izbršite.

4.17.2. Projektne lastnosti TortoiseSVN

TortoiseSVN ima nekaj svojih posebnih lastnosti, ki se začnejo s `tsvn`:

- `tsvn:logminsize` nastavi najmanjšo dolžino sporočila dnevniškega zapisa za objavo. Če vnesete krajši zapis, objava ni mogoča. Ta lastnost je uporabna, ker vas opozori, da morate pri vsaki objavi podati primerno opisno sporočilo. Če ta lastnost ni nastavljena ali je nastavljena na vrednost nič, sistem omogoča objavo brez sporočila.

`tsvn:lockmsgminsize` nastavi najmanjšo dolžino sporočila pri zaklepu. Če vnesete krajši zapis, pridobitev zaklepa ni mogoča. Ta lastnost je uporabna, ker vas opozori, da morate pri pridobivanju zaklepa podati primerno opisno sporočilo. Če ta lastnost ni nastavljena ali je nastavljena na vrednost nič, sistem omogoča pridobivanje zaklepa brez vnosa sporočila.

- `tsvn:logwidthmarker` se uporablja pri projektih, ki zahtevajo, da je sporočilo dnevniškega zapisa oblikovano tako, da imajo vrstice določeno največjo dolžino (tipično je ta dolžina 80 znakov) pred novo vrstico. Nastavitev vrednosti, različne od nič, ima dve posledici v oknu za vnos dnevniškega zapisa: nariše mejo največje dovoljene dolžine vrstice in onemogoči oblikovanje besedila, tako da vidite, katere vrstice so predolge. Opomba: ta zmožnost pravilno deluje le, če je pisava, uporabljena za sporočila dnevniških zapisov, konstantne širine.
- `tsvn:logtemplate` se uporablja pri projektih, kjer je oblika sporočil dnevniških zapisov predpisana. Ta lastnosti vsebuje besedilo v več vrsticah, ki se vstavi v okno za vnos sporočila dnevniškega zapisa ob pričetku objave. Zapis nato uredite in vnesete zahtevane informacije. Opomba: če hkrati s to lastnostjo uporabljate tudi lastnost `tsvn:logminsize`, nastavite najmanjšo dolžino dnevniškega zapisa na vrednost, ki je večja od dolžine predloge, sicer določanje najmanjše velikosti zapisa nima več zaščitne vloge.
- Subversion omogoča nastavljanje samodejnih lastnosti ("autoprops"), ki se nastavijo vsem novim ali uvoženim datotekam, odvisno od končnice datoteke. Vsak uporabnik mora imeti v svoji konfiguracijski datoteki odjemalca nastavljene ustrezne samodejne lastnosti. Lastnost `tsvn:autoprops` lahko nastavite tudi neposredno na mapah. Pri dodajanju ali uvažanju datotek se te lastnosti spojijo z lastnostmi, ki so nastavljene v konfiguracijski datoteki. Oblika je enaka kot pri samodejnih lastnostih Subversion, n. pr: `*.sh = svn:eol-style=native;svn:executable` nastavi dve lastnosti na datotekah s končnico `.sh`.

Če pride do spora med krajevnimi samodejnimi lastnostmi in lastnostjo `tsvn:autoprops`, se upoštevajo nastavitve projekta, ker so specifične za ta projekt.

- V oknu za objavo lahko prilepite seznam spremenjenih datotek, vključno s stanjem vsake datoteke (dodano, spremenjeno...). Lastnost `tsvn:logfilelistenglish` določa, ali se stanje vstavlja v angleškem jeziku ali v jeziku, ki ga uporabljate v programu TortoiseSVN. Če te lastnosti ne nastavite, se upošteva privzeta vrednost `true`, kar pomeni, da so podatki o stanju prilepljeni v angleškem jeziku.
- V programu TortoiseSVN lahko uporabljate črkovalnike, ki jih uporabljata tudi programa OpenOffice in Mozilla. Če imate nameščeno katero izmed teh dveh aplikacij, ta lastnost definira, kateri črkovalnik naj se uporabi, torej v katerem jeziku se vpisujejo sporočila dnevniških zapisov. Lastnost `tsvn:projectlanguage` nastavi jezikovni modul, ki naj se uporabi za preverjanje sporočil dnevniških zapisov. Vrednosti za vaš jezik najdete na naslednji strani: [MSDN: Language Identifiers](http://msdn2.microsoft.com/en-us/library/ms776260.aspx) [http://msdn2.microsoft.com/en-us/library/ms776260.aspx].

Vrednost lahko vnesete v desetiški obliki ali v šestnajstiški obliki - v tem primeru ji dodajte predpono `0x`. Primer: za uporabo ameriške angleščine vnesite `0x0409` ali `1033`.

- Lastnost `tsvn:logsummary` se uporablja za izluščenje dela sporočila dnevniškega zapisa, ki se izpiše v dnevniku kot povzetek sporočila.

Vrednost lastnosti `tsvn:logsummary` mora biti enovrstični regularni izraz, ki vsebuje eno skupino. Rezultat te skupine se uporabi za povzetek.

Primer: `\[SUMMARY\]:\s+(.*)` najde vso besedilo, ki se nahaja za nizom "[SUMMARY]" in ga uporabi kot povzetek.

- Kadar želite dodati novo lastnost, jo lahko izberete iz seznama v spustnem polju ali pa vpišete poljubno ime lastnosti. Če na projektu uporabljate uporabniško določene lastnosti in želite, da se te lastnosti pojavijo v spustnem polju (v izogib tipkarskim napakam pri pisanju imena), lahko ustvarite seznam lastnih lastnosti z uporabo `tsvn:userfileproperties` in `tsvn:userdirproperties`. Te lastnosti nastavite mapi. Ko boste urejali lastnosti kateregakoli elementa v tej mapi, se bodo lastne lastnosti pojavile na seznamu preddefiniranih imen.

Nekatere lastnosti `tsvn`: zahtevajo vrednost `true/false`. TortoiseSVN razume tudi besedo `yes` kot sinonim za `true` in besedo `no` kot sinonim za `false`.

TortoiseSVN zna vključiti nekatera orodja za sledenje zadev. Za to se uporabljajo lastnosti, ki se začnejo z `bugtraq:`. Za več informacij preberite [Razdelek 4.28, "Integracija s sistemi za sledenje zadev"](#).

Vključi lahko tudi nekatera spletna orodja za brskanje po skladiščih. Za to uporablja lastnosti, ki se začnejo z `webviewer:`. Za več informacij preberite [Razdelek 4.29, "Integracija z internetno naravnanimi pregledovalniki skladišč"](#).



Nastavite projektne lastnosti na mapah

Te posebne projektne lastnosti morate nastaviti na *mapah*, da bo sistem deloval. Ko objavite datoteko ali mapo, se lastnosti preberejo iz te mape. Če TortoiseSVN lastnosti ne najde tam, jih išče v nadrejenih mapah, dokler ne pride do mape brez različic ali do korenke mape pogona (n. pr. `C:\`). Če ste prepričani, da bodo uporabniki prevzemali projekt zgolj iz poti n. pr. `trunk/` in ne iz katere izmed podmap, potem je dovolj, da lastnosti nastavite na `trunk/`. Če niste prepričani, potem raje nastavite te lastnosti rekurzivno na vseh podmapah. Lastnosti, ki so globlje v projektni hierarhiji, povezijo lastnosti, ki so višje (bližje poti `trunk/`).

Zgolj za projektne lastnosti lahko uporabite potrditveno polje **Rekurzivno**. S tem nastavite lastnost na vse podmape v hierarhiji, na datoteke pa ne.

Ko s pomočjo programa TortoiseSVN dodate nove podmape, se tja samodejno prenesejo projektne lastnosti nadrejene mape.



Opozorilo

Čeprav so projektne lastnosti TortoiseSVN zelo uporabne, delujejo le z odjemalcem TortoiseSVN, nekatere celo samo z novjšimi različicami TortoiseSVN. Če uporabniki na vašem projektu uporabljajo različne odjemalce za Subversion ali pa starejše različice TortoiseSVN, raje uporabite skripte akcij v skladišču za zagotavljanje spoštovanja pravil. Lastnosti `tsvn`: vam le pomagajo nastaviti pravila, ne morejo pa jih zagotavljati.

4.18. External Items

Sometimes it is useful to construct a working copy that is made out of a number of different checkouts. For example, you may want different files or subdirectories to come from different locations in a repository, or perhaps from different repositories altogether. If you want every user to have the same layout, you can define the `svn:externals` properties to pull in the specified resource at the locations where they are needed.

4.18.1. External Folders

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 **Add...**

Select the `svn:externals` property from the combobox and write in the edit box the repository URL in the format `url folder` or if you want to specify a particular revision, `-rREV url folder`. You can add multiple external projects, 1 per line. Suppose that you have set these properties on `D:\dev\project1`:

```
http://sounds.red-bean.com/repos sounds
http://graphics.red-bean.com/repos/fast%20graphics "quick graphs"
-r21 http://svn.red-bean.com/repos/skin-maker skins/toolkit
```

Now click **Set** and commit your changes. When you (or any other user) update your working copy, Subversion will create a sub-folder `D:\dev\project1\sounds` and checkout the sounds project, another sub-folder `D:\dev\project1\quick_graphs` containing the graphics project, and finally a nested sub-folder `D:\dev\project1\skins\toolkit` containing revision 21 of the skin-maker project.

URLs must be properly escaped or they will not work, e.g. you must replace each space with `%20` as shown in the second example above.

If you want the local path to include spaces or other special characters, you can enclose it in double quotes, or you can use the `\` (backslash) character as a Unix shell style escape character preceding any special character. Of course this also means that you must use `/` (forward slash) as a path delimiter. Note that this behaviour is new in Subversion 1.6 and will not work with older clients.



Use explicit revision numbers

Priporočamo, da pri navajanju naslovov skladišč za zunanje projekte uporabljate tudi številke revizij, kot je opisano zgoraj. S tem si zagotovite, da se boste sami odločili, kdaj boste uporabili drug posnetek zunanjega projekta in natančno katerega. Poleg logičnega pogleda na stvar - da ne boste presenečeni nad spremembami v zunanjem projektu, nad katerim nimate nadzora - to prinaša tudi prednost pri posodabljanju na staro revizijo. V tem primeru se bodo zunanji projekti posodobili na natančno takšno revizijo, ki ustreza reviziji vaših projektov. Pri projektih razvoja programske opreme to pomeni uspešno oziroma neuspešno gradnjo stare različice kompleksnega projekta.



Older `svn:externals` definitions

The format shown here was introduced in Subversion 1.5. You may also see the older format which has the same information in a different order. The new format is preferred as it supports several useful features described below, but it will not work on older clients. The differences are shown in the *Subversion Book* [<http://svnbook.red-bean.com/en/1.5/svn.advanced.externals.html>].

Če se zunanji projekt nahaja v istem skladišču, bodo vse spremembe v njem tudi na seznamu za objavo, ko boste objavljali spremembe v glavnem projektu.

Če se zunanji projekt nahaja v drugem skladišču, bodo vse spremembe, ki jih naredite na zunanjem projektu, prikazane, ko boste objavili glavni projekt, vendar morate zunanje spremembe objaviti posebej.

Če uporabljate absolutne naslove URL v definicijah `svn:externals` in morate delovno kopijo premestiti (n. pr. če se spremeni naslov skladišča), se zunanji ne bodo spremenili in morda ne bodo več delovali.

V izogib tem težavam odjemalci Subversion verzije 1.5 ali novejša podpirajo relativne zunanje naslove URL. Na voljo so štiri načini podajanja relativnega naslova URL. V naslednjih primerih predpostavljamo, da imamo dve skladišči: enega na naslovu `http://primer.com/`

svn/sklad-1 in drugega na naslovu `http://primer.com/svn/sklad-2`. Prezem naslova `http://primer.com/svn/sklad-1/projekt/trunk` naredimo v mapo `C:\Delo`, lastnost `svn:externals` pa je nastavljena na glavno vejo (trunk).

Relativno na nadrejeno mapo

Ti naslovi se vedno začnejo s predpono `../`. Primer:

```
../../widgets/foo common/foo-widget
```

S tem se naslov `http://primer.com/svn/sklad-1/widgets/foo` razširi v `C:\Delo\common\foo-widget`.

Upoštevajte, da je naslov URL relativen na naslov URL mape, ki ima nastavljeno lastnost `svn:externals` in ne na mapo, kjer se zunanji elementi nahajajo.

Relativno na korensko pot skladišča

Ti naslovi se vedno začnejo z `^/`. Primer:

```
^/widgets/foo common/foo-widget
```

Naslov `http://primer.com/svn/sklad-1/widgets/foo` ustreza `C:\Delo\common\foo-widget`.

S tem se enostavno sklicujete na skladišča, ki imajo enako lastnost `SVNParentPath` (skupna mapa, ki vsebuje več skladišč). Primer:

```
^/../sklad-2/hammers/claw common/claw-hammer
```

Naslov `http://primer.com/svn/sklad-2/hammers/claw` ustreza `C:\Delo\common\claw-hammer`.

Relativno na shemo

Naslovi URL, ki se začnejo z nizom `//`, prekopirajo le shemo naslova URL. To je uporabno, kadar je potrebno gostitelja dosežati z različnimi shemami, odvisno od mrežne lokacije; n. pr. odjemalci v intranetu uporabijo `http://`, medtem kot zunanji odjemalci uporabijo `svn+ssh://`. Primer:

```
//primer.com/svn/sklad-1/widgets/foo common/foo-widget
```

ustreza `http://primer.com/svn/sklad-1/widgets/foo` ali `svn+ssh://primer.com/svn/sklad-1/widgets/foo`, odvisno od metode, ki je bila uporabljena za prevzem mape `C:\Delo`.

Relativno na naslov strežnika

Naslovi URL, ki se začnejo z nizom `/`, prekopirajo shemo in gostitelja naslova URL, n. pr.

```
/svn/sklad-1/widgets/foo common/foo-widget
```

ustreza `http://primer.com/svn/sklad-1/widgets/foo` v `C:\Delo\common\foo-widget`. Če ste delovno kopijo prevzeli iz drugega strežnika na naslovu `svn+ssh://drug.streznik.net/svn/sklad-1/projekt1/trunk`, zunanje reference ustrezajo `svn+ssh://drug.streznik.net/svn/sklad-1/widgets/foo`.

You can also specify a peg revision after the URL if required, e.g. `http://sounds.red-bean.com/repos@19`.

Če želite izvedeti več o tem, kako Subversion obravnava lastnosti, preberite [Razdelek 4.17](#), “Nastavitve projekta”.

Za pregled različnih načinov dostopa do skupnih projektov si preberite [Razdelek B.6](#), “Kako vključim skupni podprojekt”.

4.18.2. External Files

As of Subversion 1.6 you can add single file externals to your working copy using the same syntax as for folders. However, there are some restrictions.

- The path to the file external must place the file in an existing versioned folder. In general it makes most sense to place the file directly in the folder that has `svn:externals` set, but it can be in a versioned sub-folder if necessary. By contrast, directory externals will automatically create any intermediate unversioned folders as required.
- The URL for a file external must be in the same repository as the URL that the file external will be inserted into; inter-repository file externals are not supported.

A file external behaves just like any other versioned file in many respects, but they cannot be moved or deleted using the normal commands; the `svn:externals` property must be modified instead.



File externals support incomplete in Subversion 1.6

In subversion 1.6 it is not possible to remove a file external from your working copy once you have added it, even if you delete the `svn:externals` property altogether. You have to checkout a fresh working copy to remove the file.

4.19. Ustvarjanje vej/oznak

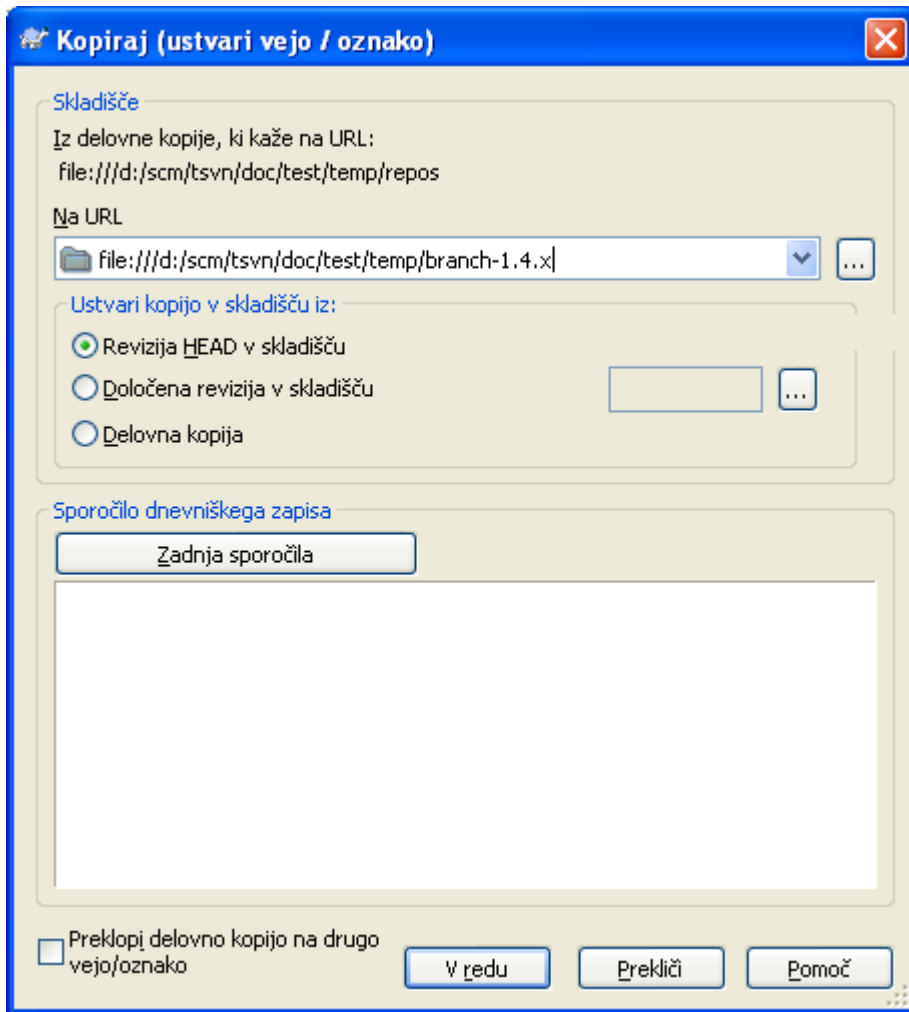
Ena od zmožnosti sistemov za nadzor različic je tudi možnost izoliranja sprememb v svojo razvojno linijo. Linija se prikaže kot *veja*. Veje se pogosto uporabljajo za preizkušanje novih stvari, da ne bi obremenjevali glavne razvojne linije z napakami prevajalnika in hrošči. Ko je nova zmožnost dovolj stabilna, se razvojna veja *spoji* nazaj v glavno vejo.

Še ena lastnost sistemov za nadzor različic je možnost označevanja določenih revizij (n. pr. verzije izdaje), tako da lahko kasneje ponovno zgradite izdajo ali nastavite razvojno okolje. Ta proces se imenuje *označevanje*.

Subversion nima posebnih ukazov za ustvarjanje vej ali oznak, ampak uporablja takoimenovane “poceni kopije”. Poceni kopije so podobne povezavam (hard links) v sistemu Unix, kar pomeni, da se namesto ustvarjanja kopije skladišča naredi notranja povezava, ki kaže na določeno drevo/revizijo. Zato je ustvarjanje vej in oznak zelo hitro, hkrati pa ne zavzema skoraj nobenega dodatnega prostora v skladišču.

4.19.1. Ustvarjanje veje ali oznake

Če ste projekt uvozili v priporočeno sturkturo map, je ustvarjanje vej ali oznak zelo preprosto:



Slika 4.34. Okno za za ustvarjanje veje/oznake

Izberite mapo v delovni kopiji, ki jo želite prekopirati v vejo ali oznako, nato pa izberite TortoiseSVN → Veja/oznaka....

Privzeti ciljni naslov URL za novo vejo bo naslov URL, na katerem je osnovana vaša delovna kopija. Naslov URL boste morali urediti, tako da bo kazal na novo vejo/oznako. Namesto

```
http://svn.collab.net/repos/ProjectName/trunk
```

sedaj uporabite naprimer

```
http://svn.collab.net/repos/ProjectName/tags/Release_1.10
```

Če se ne spomnite, kakšen sistem označevanja ste uporabili pri prejšnjih uporabah ukaza, kliknite na gumb na desni in odpre se vam brskalnik po skladišču. Tako lahko vidite obstoječo strukturo map skladišča.

Sedaj morate izbrati vir kopiranja. Imate tri možnosti:

Revizija HEAD v skladišču

Nova veja se skopira neposredno v skladišču iz revizije HEAD. Prenos podatkov iz delovne kopije ni potreben, zato je ustvarjanje veje zelo hitro.

Določena revizija v skladišču

Nove veja se skopira neposredno v skladišču, vendar se lahko odločite za starejšo revizijo. To je uporabno, če ste prejšnji teden, ko ste naredili novo izdajo, pozabili kreirati oznako. Če se ne spomnite številke revizije, kliknite na gumb na desni in prikazalo se bo okno z dnevnikom revizij, od koder lahko izberete številko revizije. Iz delovne kopije se v skladišče ne prenašajo podatki, tako da je veja ustvarjena zelo hitro.

Delovna kopija

Nova veja je identična kopija krajevne delovne kopije. Če ste nekatere datoteke posodobili na starejšo revizijo, bodo te starejše revizije datotek prav tako skopirane na vejo. Seveda se lahko pri takšnem zahtevnem tipu oznake zgodi, da se podatki prenašajo iz delovne kopije v skladišče, če tam še ne obstajajo.

Če želite, da se vaša delovna kopija samodejno preklopi na novo ustvarjeno vejo, vklopite potrditveno polje **Preklopi delovno kopijo na drugo vejo / oznako**. Če želite to narediti, se najprej prepričajte, da vaša delovna kopija nima krajevnih sprememb. Če jih ima, potem bodo le-te spojene v delovno kopijo veje, ko naredite preklap.

Če želite objaviti novo kopijo v skladišču, pritisnite gumb **V redu**. Ne pozabiti na dnevniški zapis. Ne pozabite, da se kopija ustvari *znotraj skladišča*.

Upoštevajte: če ne potrdite izbire za preklap nove veje na delovno kopijo, ustvarjanje vej in oznak *nima* vpliva na delovno kopijo. Tudi če vejo ustvarite iz delovne kopije, se spremembe objavijo na novi veji in ne na glavni veji, tako da je vaša delovna kopija še vedno lahko označena kot spremenjena (glede na glavno vejo).

4.19.2. Prezvzeti ali preklopiti...

... to (zares) ni vprašanje. Medtem ko prevzem prevzame vse datoteke iz izbrane veje v delovno kopijo, TortoiseSVN → Preklopi... prenese le spremenjene podatke v delovno kopijo. Dobro za obremenitev omrežja in vaše potrpljenje :-)

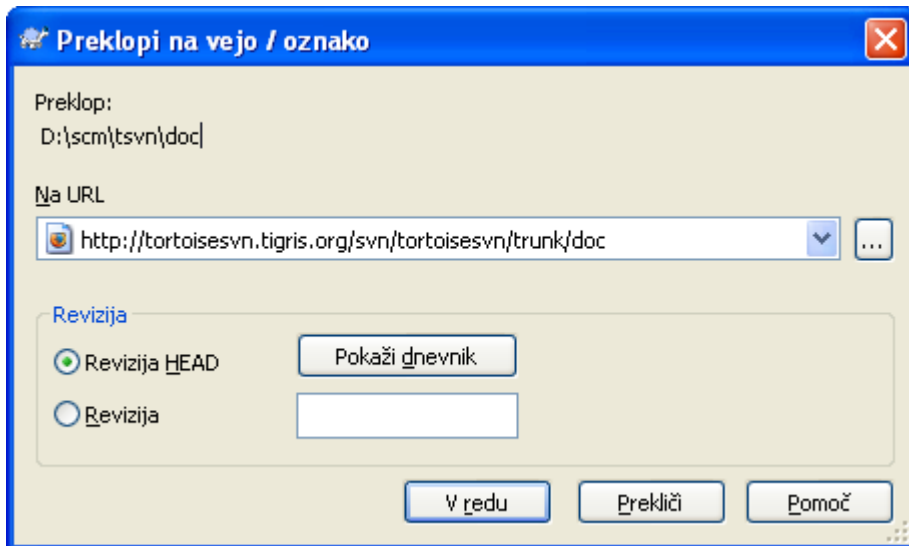
Na voljo je več načinov, kako začnete delati z novo ustvarjeno vejo ali oznako. Lahko:

- Uporabite ukaz TortoiseSVN → Prevzem in ustvarite svež prevzem v prazno mapo. Prevzem lahko naredite na katerikoli lokaciji na svojem disku. Naredite lahko poljubno število prevzemov iz skladišča.
- Preklopite delovno kopijo na novo ustvarjeno kopijo v skladišču. Izberite vrhno mapo projekta in iz kontekstnega menija uporabite TortoiseSVN → Preklopi...

V naslednjem pogovornem oknu vnesite naslov URL veje, ki ste jo ustvarili. Izberite radijski gumb **Revizija HEAD** in kliknite na gumb **V redu**. S tem ste delovno kopijo preklopili na novo vejo/oznako.

Preklap, enako kot posodobitev, nikoli ne povozi krajevnih sprememb. Vse spremembe, ki ste jih naredili v delovni kopiji in še niso bile objavljene, bodo spojene, ko naredite preklap. Če tega ne želite, potem morate spremembe objaviti ali pa delovno kopijo povrniti na objavljeno revizijo (tipično HEAD).

- Če želite delati na glavni in stranski veji, vendar ne želite delati svežih prevzemov, ki so časovno potratni, lahko naredite prevzem glavne veje, potem pa uporabite ukaz TortoiseSVN → Preklopi..., s katerim delovno kopijo posodobite na izbrano vejo.



Slika 4.35. Okno za preklp

Čeprav Subversion ne razlikuje med oznakami in vejami, se pri pri tipični uporabi ta dva ukaza nekoliko razlikujeta.

- Oznake se ponavadi uporabljajo za ustvarjanje statičnega posnetka projekta v določenem trenutku. Kot take se ponavadi ne uporabljajo za razvijanje - za to se uporabljajo veje, zato priporočamo strukturo skladišča `/trunk /branches /tags`. Delo na reviziji oznake *ni dobra ideja*, ker pa datoteke v vaši delovni kopiji niso zaščitene pred pisanjem, se vam napaka hitro zgodi. Če pa boste poskušali narediti objavo v skladišču, ki vsebuje mapo `/tags/`, vas bo TortoiseSVN opozoril.
- Včasih morate narediti dodatne spremembe na izdaji, ki ste jo že označili. Pravilen postopek je, da ustvarite novo vejo iz oznake in vejo objavite. Nato na novi veji naredite potrebne spremembe in ustvarite novo oznako iz veje, n. pr. `Verzija_1.0.1`.
- Če spremenite delovno kopijo, ustvarjeno iz veje, in jo objavite, gredo vse spremembe na novo vejo in *ne* na glavno vejo. Shranijo se le spremembe, nespremenjeni del pa ostane poceni kopija.

4.20. Spajanje

Kadar se veje uporabljajo za ločeno razvijanje, kasneje želite spremembe na eni veji spojiti nazaj v glavno vejo oz. obratno.

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://svnbook.red-bean.com/en/1.5/svn.branchmerge.html) [http://svnbook.red-bean.com/en/1.5/svn.branchmerge.html] in the Subversion book, which gives a full description and many examples of how it is used.

Naslednja stvar, ki jo je treba poudariti, je, da se spajanje *vedno* izvaja v delovni kopiji. Če želite spremembe spojiti v vejo, morate imeti delovno kopijo, prevzeto iz te veje, in na njej izvesti ukaz TortoiseSVN → Spoji....

V splošnem je priporočljivo, da spajanje izvajate ob nespremenjeni delovni kopiji. Če ste naredili kakšne spremembe v delovni kopiji, jih najprej objavite. Če operacija spajanja ne bo uspela, boste verjetno želeli povrniti spremembe. Ukaz **Povrni** bo izbrisal vse spremembe na delovni kopiji, vključno s tistimi, ki ste jih naredili pred poskusom spajanja.

Poznamo tri načine spajanja, ki se med seboj nekoliko razlikujejo. Izbiro načina vam omogoča prva stran čarovnika za spajanje.

Spoji območje revizij

Ta metoda pokriva primer, ko ste na veji (ali na glavni veji) naredili spremembe, ki jih sedaj želite prenesti na drugo vejo.

Od Subversiona zahtevate naslednje: "Izračunaj spremembe, ki so potrebne, da iz revizije 1 ([OD]) veje A dobimo revizijo 7 ([DO]) veje A, in spremembe namesti na delovno kopijo (glavne ali stranske veje B)."

Vključitev veje

Ta način se uporablja v primeru, ko ustvarite vejo za razvoj neke zmožnosti programa, kot je to opisano v knjigi *The Subversion Book*. Vse spremembe, ki so se dogajale na glavni veji, ste vsak teden spajali na vejo. Sedaj je razvoj nove zmožnosti končan, zato želite to kodo spojiti na glavno vejo. Ker ste vejo ves čas sinhronizirali z glavno vejo, sta stranska in glavna veja identični, le da stranska veja vsebuje kodo za novo zmožnost.

To je poseben primer spajanja dreves (opisano spodaj) in zahteva zgolj naslov URL, iz katerega spajate (običajno stranska veja, na kateri ste razvijali novo zmožnost). S sledenjem spajanja se izračuna pravilno območje spajanja in preveri, ali je stranska veja sinhronizirana z glavno. Na ta način preprečimo izgubo objavljenih sprememb na glavni veji od zadnje sinhronizacije.

Po spojitvi je celoten razvoj na stranski veji spojen na glavno razvojno vejo. Stranska veja postane odvečna in jo lahko izbrišemo.

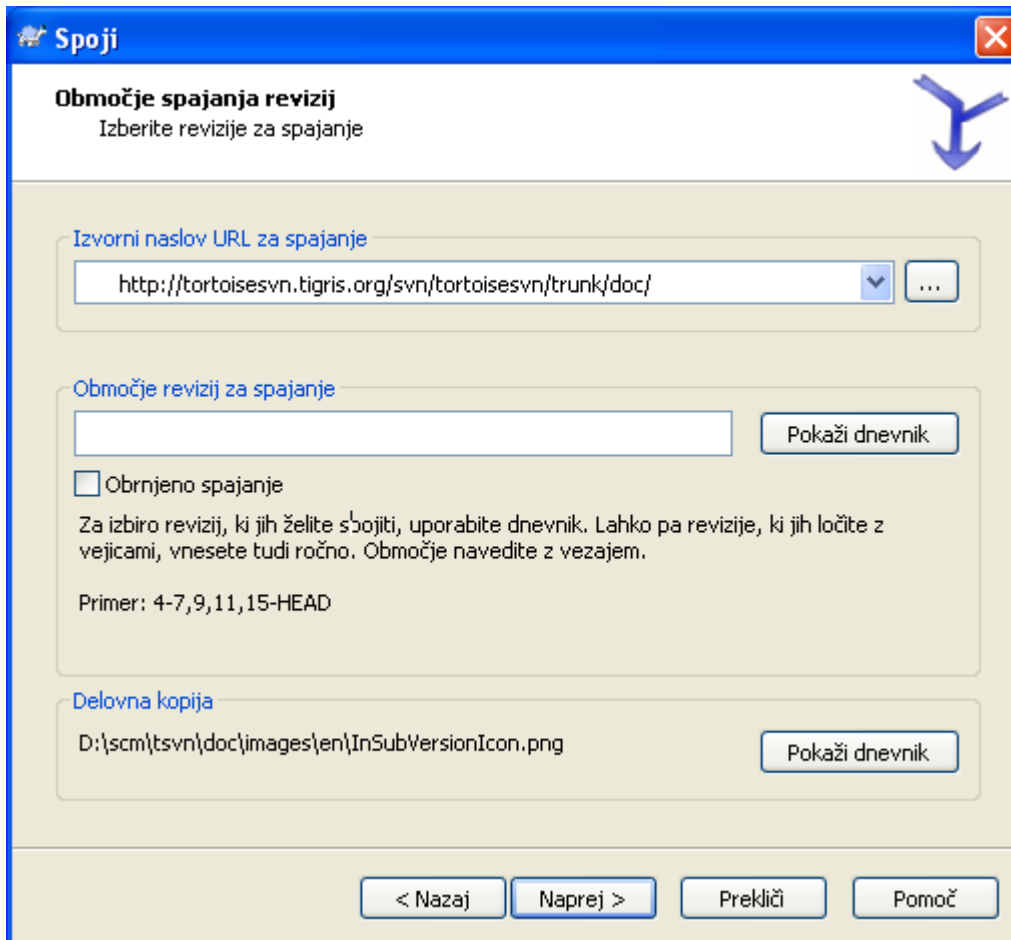
Once you have performed a reintegrate merge you should not continue to use it for development. The reason for this is that if you try to resynchronize your existing branch from trunk later on, merge tracking will see your reintegration as a trunk change that has not yet been merged into the branch, and will try to merge the branch-to-trunk merge back into the branch! The solution to this is simply to create a new branch from trunk to continue the next phase of your development.

Spoji dveh različni drevesi

To je bolj splošen primer vključitve veje. V tem primeru od Subversiona zahtevate naslednje: "Izračunaj spremembe, ki so potrebne, da iz revizije HEAD ([OD]) na glavni veji dobimo revizijo HEAD ([DO]) na stranski veji, in spremembe namesti na delovno kopijo (glavne veje)." Rezultat postopka sta identični glavna in stranska veja.

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/1.5/svn.advanced.vendorbr.html>] in the *Subversion Book*.

4.20.1. Spajanje območja revizij



Slika 4.36. Čarovnik za spajanje - Izberite obseg revizije

V polje **Izvorni naslov URL za spajanje**: vnesite poln naslov URL veje ali oznake s spremembami, ki jih želite spojiti v delovno kopijo. Lahko pa kliknete na gumb S tem pobrsKate po skladišču in najdete željeno vejo. Če ste s te veje že spajali, lahko uporabite spustni seznam, ki prikazuje zgodovino že vnešenih naslovov URL.

V polje **Območje revizij za spajanje** vnesite seznam revizij, ki jih želite spojiti. Lahko vpišete eno revizijo, seznam določenih revizij, ločenih z vejicami, območje revizij, ločenih z vezajem ali kombinacijo vsega naštetega.



Pomembno

There is an important difference in the way a revision range is specified with TortoiseSVN compared to the command line client. The easiest way to visualise it is to think of a fence with posts and fence panels.

With the command line client you specify the changes to merge using two “fence post” revisions which specify the *before* and *after* points.

With TortoiseSVN you specify the changeset to merge using “fence panels”. The reason for this becomes clear when you use the log dialog to specify revisions to merge, where each revision appears as a changeset.

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.

This difference has generated a lot of heat on the mailing lists. We acknowledge that there is a difference from the command line client, but we believe that for the majority of GUI users it is easier to understand the method we have implemented.

Najlažji način, kako izberete večje število revizij, je s klikom na gumb **Pokaži dnevnik**; prikazalo se bo okno s seznamom zadnjih sprememb in pripadajočimi komentarji. Če želite spojiti spremembe iz ene revizije, izberite to revizijo. Če želite spojiti spremembe iz več revizij, izberite območje (z uporabo modifikatorja **Shift**). Kliknite na gumb **V redu** in seznam številke revizij se bo samodejno izpolnil.

Če želite spremembe spojiti nazaj *ven* iz delovne kopije (da prekličete že objavljeno spremembo), izberite revizije, ki jih želite povrniti in potrdite polje **Obratno spajanje**.

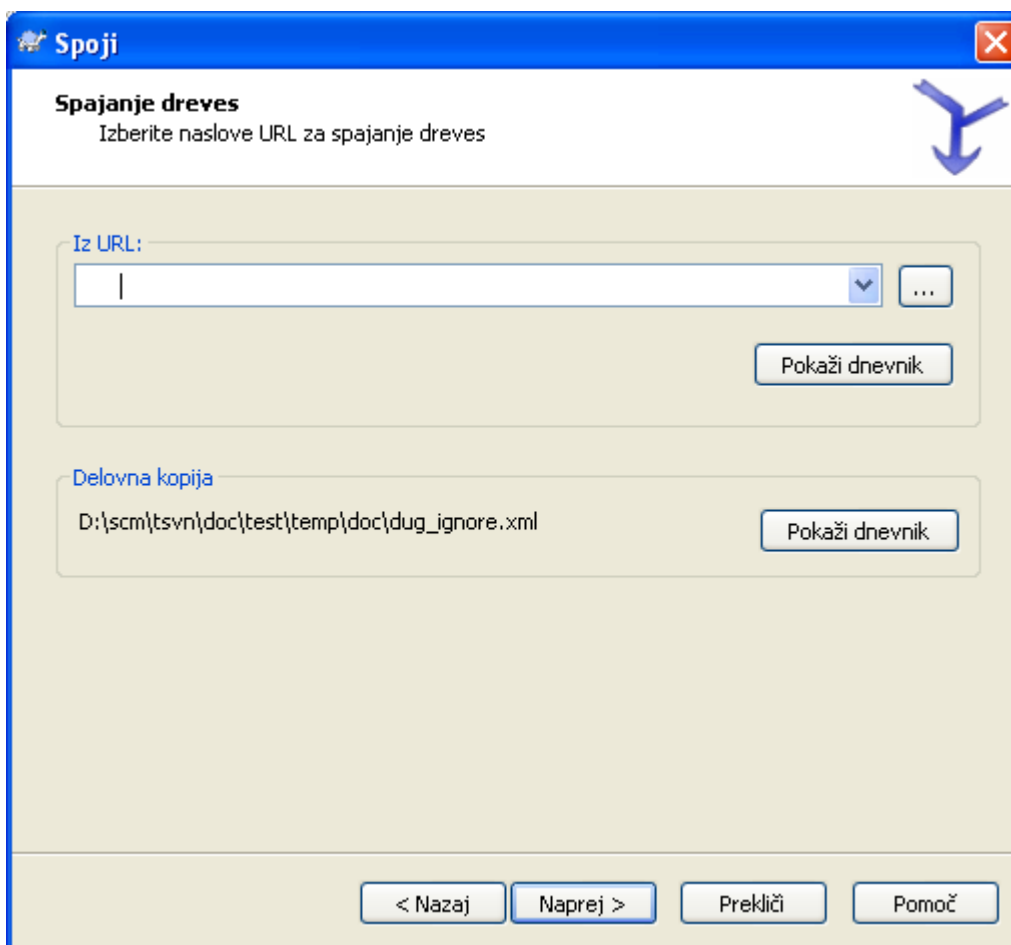
If you have already merged some changes from this branch, hopefully you will have made a note of the last revision merged in the log message when you committed the change. In that case, you can use **Show Log** for the Working Copy to trace that log message. Remembering that we are thinking of revisions as changesets, you should Use the revision after the end point of the last merge as the start point for this merge. For example, if you have merged revisions 37 to 39 last time, then the start point for this merge should be revision 40.

Če uporabljate sledenje spajanja, vam ni potrebno vedeti, katere revizije ste že spojili - Subversion si jih zapomni. Če pustite polje za vnos območja revizij prazno, bodo samodejno vključene vse revizije, ki še niso bile spojene. Za več informacij preberite [Razdelek 4.20.6, "Sledenje spajanja"](#).

Če v skladišču objavljajo tudi drugi uporabniki, potem bodite pri uporabi spajanja do revizije HEAD previdni. Revizija HEAD mogoče ni tista, kot mislite, saj je od vaše zadnje posodobitve delovne kopije lahko kak drug uporabnik naredil objavo.

Kliknite gumb **Naprej** in pojdite na [Razdelek 4.20.4, "Možnosti spajanja"](#)

4.20.2. Vključitev veje



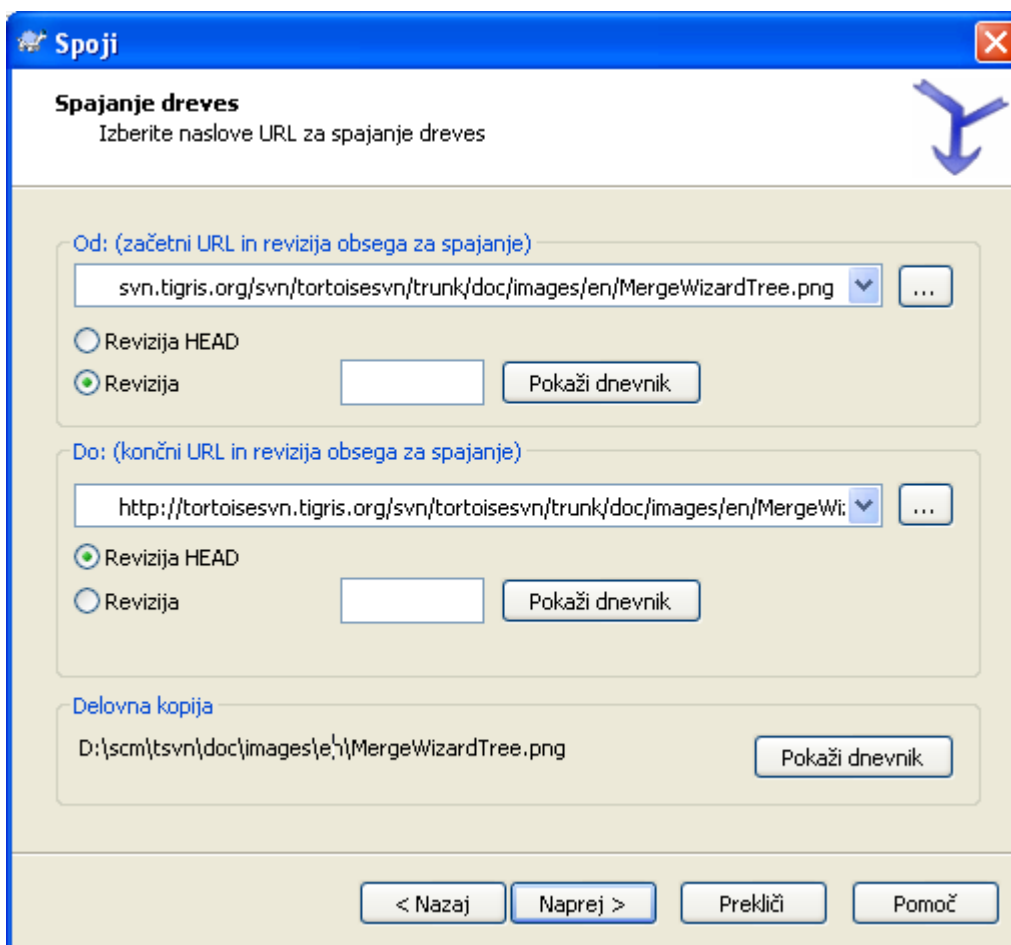
Slika 4.37. Čarovnik za spajanje - Vključitev veje

Za spajanje stranske veje na glavno vejo morate čarovnika za spajanje zagnati iz delovne kopije glavne veje.

V polje **Iz URL:** vpišite poln naslov URL mape stranske veje, ki jo želite spojiti na glavno vejo. Uporabite lahko tudi gumb ... in pobrsate po skladišču.

Za izvajanje vključevanja veje morajo biti izpolnjeni določeni pogoji. Strežnik mora podpirati sledenje spajanja. Delovna kopija mora biti prevzeta v celoti (delni prevzemi niso dovoljeni) in biti brez krajevnih sprememb, preklapov ali elementov, posodobljenih na starejše revizije. Vse spremembe, narejene na glavni veji, morajo biti že spojene na stransko vejo (oziroma morajo biti označene kot spojene). Območje revizij za spajanje se določi samodejno.

4.20.3. Spajanje dveh različnih dreves



Slika 4.38. Čarovnik za spajanje - Spajanje dreves

Če ta način uporabljate za spajanje stranske veje na glavno vejo, morate čarovnika za spajanje zagnati iz delovne kopije glavne veje.

V polje **Od:** vpišite poln naslov URL mape *glavne veje*. To se morda zdi napačno, a ne pozabite, da je glavna veja začetna točka spajanja, na katero želite dodati spremembe iz veje, kjer ste razvijali novo zmožnost. Uporabite lahko tudi gumb ... in pobrsate po skladišču.

V polje **Do:** vpišite poln naslov URL mape veje, kjer ste razvijali novo zmožnost.

V polji **Revizija** vpišite zadnji reviziji, pri kateri sta bili drevesi sinhronizirani. Če ste prepričani, da sprememb ne objavlja nihče drug, lahko v obeh primerih uporabite revizijo HEAD. Če obstaja možnost, da je po sinhronizaciji še kdo drug objavljajal spremembe, potem raje uporabite točno določeno številko revizije, da se izognete izgubi sprememb iz novejših objav.

Za izbiro revizije lahko uporabite tudi gumb **Pokaži dnevnik**.

4.20.4. Možnosti spajanja

Ta stran čarovnika vam pred začetkom spajanja omogoča nastavitve naprednih možnosti. V večini primerov lahko uporabite kar privzete nastavitve.

Določite lahko globino spajanja, ki pove, kako globoko v delovno kopijo spajanje deluje. Za definicijo globine pogledjte [Razdelek 4.3.1, "Globina prevzema"](#). Privzeta globina je delovna kopija, ki uporablja obstoječo nastavitve globine, kar največkrat tudi potrebujete.

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/1.5/svn.branchmerge.advanced.html#svn.branchmerge.advanced.ancestry>]

Določite lahko, kako se upoštevajo zaključki vrstic in presledki. Možnosti so opisane v **Razdelek 4.10.2, "Nastavitev zaključkov vrstic in presledkov"**. Privzeto obnašanje je, da se zaključki vrstic in presledki obravnavajo enako kot vse druge spremembe.

Če uporabljate sledenje spajanja in želite revizijo označiti kot spojeno brez dejanskega spajanja, potrdite polje **Zgolj zabeleži spajanje**. To lahko naredite iz dveh razlogov. Lahko se zgodi, da je spajanje prezahtevno, da bi ga prepustili programu in naredite spremembe ročno, nato pa revizijo označite kot spojeno, da sistem sledenja spajanja ve za to. Drug razlog pa je, da želite preprečiti spajanje določene revizije. Če takšno revizijo označite kot spojeno, s tem preprečite ponovno spajanje (velja le za odjemalce, ki poznajo možnost sledenja spajanja).

Sedaj so vse nastavitve narejene, zato morate le še klikniti na gumb **Spoji**. Če želite prehodno pogledati rezultat spajanja, lahko uporabite gumb **Poizkusno spajanje**, ki izvede spajanje, vendar *ne* spremeni delovne kopije. Pokaže vam seznam datotek, ki bodo spremenjene v primeru dejanskega spajanja in označi območja, kjer se bodo pojavili spori.

Okno spajanja pokaže vsak korak procesa z območjem revizij, ki se uporablja. To obsega eno revizijo več, kot najbrž pričakujete. Če zahtevate spajanje revizije 123, okno prikaže besedilo "Spajanje revizij 122 do 123". Da bi to razumeli, se morate zavedati, da je spajanje tesno povezano z razlikovanjem. Proces spajanja poteka tako, da se ustvari seznam sprememb med dvema točkama v skladišču in namesti te spremembe v delovno kopijo. Okno spajanja tako kaže začetno in končno točko razlike.

4.20.5. Preverjanje rezultatov spajanja

Spajanje je sedaj zaključeno. Dobra praksa je, da rezultat spajanja preverimo. Spajanje je ponavadi kar komplicirano. Če je neka veja precej različna od glavne veje, se pri spajanju pogosto pojavijo spori.

Pri uporabi odjemalcev in strežnika Subversion pred različico 1.5 se informacije o spajanju ne shranjujejo in jih je zato treba voditi ločeno. Ko ste spajanje preverili in ste pripravljeni na objavo spojene revizije, v sporočilo dnevniškega zapisa *vedno* vpišite številke revizij, ki ste jih prenesli. Če boste kasneje spet spajali s te veje, boste morali vedeti, katere revizije ste že spojili. Sprememb ne smete spajati dvakrat. Za več informacij preberite *Best Practices for Merging* [<http://svnbook.red-bean.com/en/1.4/svn.branchmerge.copychanges.html#svn.branchmerge.copychanges.bestprac>] v knjigi The Subversion book.

Če strežnik in vsi odjemalci uporabljajo Subversion 1.5 ali novejši, zmožnost sledenje spajanja shrani spojene revizije in prepreči večkratno spajanje iste revizije. S tem si precej olajšate delo, saj lahko vedno izberete celotno območje revizij, spojene pa bodo le nove revizije.

Upravljanje vej je pomembno. Če želite biti na veji v koraku z glavno vejo, morate dovolj pogosto opravljati spajanje, da se veja in glavna veja ne oddaljita preveč ena od druge. Seveda se morate še vedno izogibati ponavljajočemu se spajanju sprememb, kot je opisano zgoraj.



Namig

V tem primeru veje za razvoj nove zmožnosti ne potrebujete več, ker ste novo kodo že integrirali na glavno vejo. Veja je sedaj odvečna in jo lahko izbrišete iz skladišča, če želite.



Pomembno

Subversion ne more spojiti datoteke z mapo in obratno - spaja lahko mape v mape in datoteke v datoteke. Če izberete datoteko in odprete okno za spajanje, potem morate v oknu navesti

pot do datoteke. Če izberete mapo in odprete okno, potem morate vnesti naslov URL mape za spajanje.

4.20.6. Sledenje spajanja

Subversion 1.5 prinaša zmožnost sledenja spajanja. Ko spajate spremembe iz enega drevesa v drugega, se številke spojenih revizij shranijo. Te informacije se lahko nato uporabijo za različne namene.

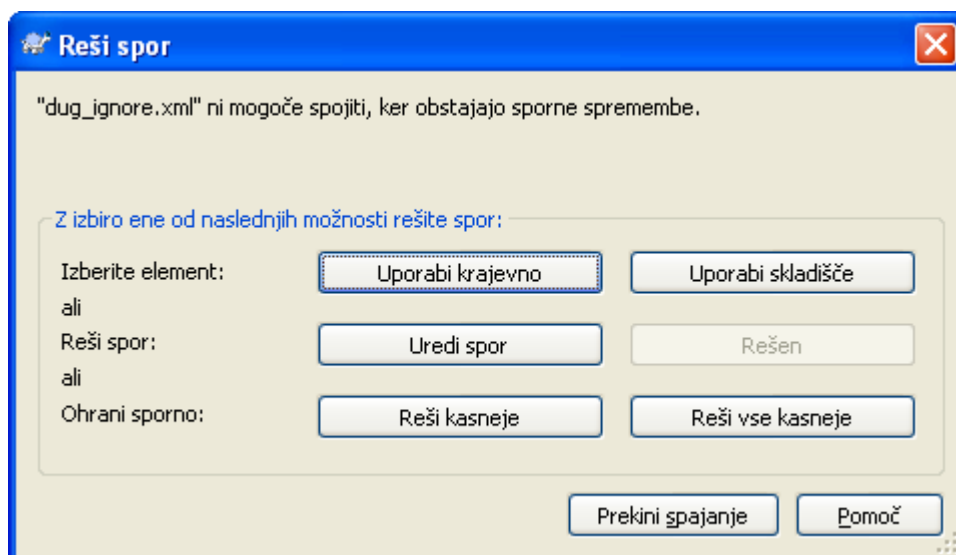
- Večkratnemu spajanju iste revizije se lahko izognete. Ko je revizija označena kot spojena, jo bo Subversion pri naslednjih poskusih enostavno preskočil.
- Ko spremembe na veji spojite nazaj na glavno vejo, vam lahko dnevnik za glavno vejo prikaže tudi objave na posebni veji. S tem lažje sledite spremembam.
- When you show the log dialog from within the merge dialog, revisions already merged are shown in grey.
- Pri prikazu krivdnih informacij datoteke lahko prikažete avtorje spojenih revizij namesto avtorja, ki je opravil spajanje.
- Revizije lahko označite kot *ne spajaj*. To storite tako, da jih dodate na seznam spojenih revizij, čeprav se spajanje dejansko nikoli ni zgodilo.

Informacije o sledenju spajanja odjemalec ob spajanju shrani v lastnost `svn:mergeinfo`. Ko je spajanje objavljeno, strežnik te informacije shrani v bazo podatkov, zato se ob zahtevi po spajanju, prikazu dnevnika ali krivde lahko ustrezno odzove. Da bi sistem deloval, morate nadgraditi tako strežnik kot tudi vse odjemalce. Starejši odjemalci namreč ne shranjujejo lastnosti `svn:mergeinfo`, starejši strežniki pa ne pošiljajo zahtevanih informacij novim odjemalcem.

Več o sledenju spajanja preberite na strani [Merge tracking documentation](http://subversion.tigris.org/merge-tracking/index.html) [http://subversion.tigris.org/merge-tracking/index.html].

4.20.7. Reševanje sporom med spajanjem

Spajanje ne poteka vedno gladko. Včasih se pojavi spor in če spajate območje revizij, je najbolje spor rešiti pred nadaljnjim spajanjem. TortoiseSVN vam pri tem pomaga tako, da vam pokaže pogovorno okno *povratne informacije o sporih spajanja*.



Slika 4.39. Povratne informacije o sporih spajanja

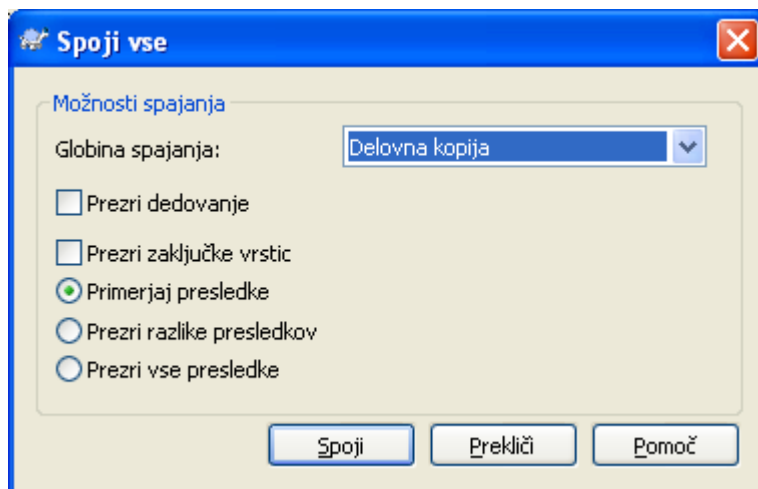
V primeru spora imate tri možnosti reševanja:

1. Včasih se odločite, da so krajevne spremembe bolj pomembne, zato želite pozabiti na spremembe iz skladišča in obdržati krajevno verzijo. Ali pa naredite obratno. V nobenem primeru se spajanje ne izvede - uporabi se ena ali druga verzija.
2. Običajno boste spore pregledali in jih rešili. Za to uporabite gumb **Uredi spor**, ki zažene orodje za spajanje. Ko ste z rezultatom zadovoljni, kliknite na gumb **Rešen**.
3. Zadnja možnost je, da reševanje spora odložite na kasnejši čas in nadaljujete s spajanjem. Lahko se odločite, da to storite le za trenutno datoteko ali pa za celotno spajanje. Upoštevajte, da v primeru nadaljnjih sprememb iste datoteke spajanja ne bo moč zaključiti.

If you do not want to use this interactive callback, there is a checkbox in the merge progress dialog **Merge non-interactive**. If this is set for a merge and the merge would result in a conflict, the file is marked as in conflict and the merge goes on. You will have to resolve the conflicts after the whole merge is finished. If it is not set, then before a file is marked as conflicted you get the chance to resolve the conflict *during* the merge. This has the advantage that if a file gets multiple merges (multiple revisions apply a change to that file), subsequent merges might succeed depending on which lines are affected. But of course you can't walk away to get a coffee while the merge is running ;)

4.20.8. Spajanje zaključene veje

Če želite spojiti vse spremembe iz glavne veje na stransko vejo, na kateri trenutno delate, lahko uporabite TortoiseSVN → **Vključitev veje...** iz razširjenega kontekstnega menija (ob desnem kliku držite pritisnjeno tipko **Shift**).



Slika 4.40. Okno Vključitev veje

Pogovorno okno je enostavno. Vse, kar morate storiti, je nastaviti možnosti za spajanje, kot je to opisano v [Razdelek 4.20.4, "Možnosti spajanja"](#). Vse ostalo naredi TortoiseSVN sam s pomočjo sledenja spajanja.

4.20.9. Vzdrževanje stranskih vej

Pri razvijanju nove zmožnosti na stranski veji se je koristno odločiti, kako se razvoj na stranski veji vključi v glavno vejo razvoja. Če se na glavni veji (`trunk`) dogajajo spremembe, se lahko zgodi, da se po določenem času veji zelo razlikujeta, tako da združevanje postane prava nočna mora.

Če je zmožnost enostavna, njen razvoj pa ne bo dolgotrajen, se lahko določite, da stransko vejo povsem ločite od glavne. Ko je razvoj na stranski veji končan, spremembe spojite nazaj na glavno vejo. V

čarovniku za spajanje uporabite opcijo **poji območje revizij**, pri čemer za območje spajanja uporabite območje revizij stranske veje.

Če bo razvoj na stranski veji potekal dalj časa, pri tem pa morate upoštevati tudi spremembe na glavni veji, morate poskrbeti, da sta vejo sinhronizirani. To preprosto pomeni, da morate v določenih časovnih razmakih spajati spremembe iz glavne veje na stransko. Tako stranska veja vsebuje vse spremembe iz glavne veje, *poleg tega pa* tudi spremembe zaradi razvoja nove zmožnosti. Za sinhronizacij uporabljate možnost **Spoji območje revizij**. Ko je razvoj na stranski veji končan, lahko spojite spremembe nazaj na glavno vejo z uporabo možnosti **Vključitev veje** ali **Spoji dve različni drevesi**.

4.21. Zaklepanje

Subversion v splošnem najbolje deluje brez zaklepanja, z uporabo modela “kopiraj-spremeni-spoji”, kot je to opisano v [Razdelek 2.2.3](#), “**Rešitev kopiraj-spremeni-spoji**”. Se pa vseeno pojavijo situacije, kjer je potrebno uveljaviti določeno obliko zaklepanja.

- Za delo uporabljate datoteke, ki jih ni mogoče spajati, naprimer grafične datoteke. Če dva uporabnika spremenita isto datoteko, spajanje ni mogoče, zato bo eden izmed njih svoje spremembe izgubil.
- Vaše podjetje je v preteklosti vedno uporabljalo nadzor različic z zaklepanjem, zato se je vodstvo odločilo, da je “zaklepanje najboljša rešitev”.

Najprej morate strežnik nadgraditi na različico Subversion 1.2 ali novejšo. Starejše različice namreč ne podpirajo zaklepanja. Če za dostop uporabljate protokol `file://`, potem morate seveda nadgraditi samo odjemalca.

4.21.1. Zaklepanje v sistemu Subversion

Po privzetih nastavitvah elementi niso zaklenjeni in vsakdo, ki ima pravice za pisanje v skladišču, lahko kadarkoli objavi spremembe katerekoli datoteke. Ostali uporabniki bodo periodično posodabljali svojo delovno kopijo. Spremembe v skladišču bodo spojene s krajevnimi spremembami.

Če za datoteko *pridobite zaklep*, potem lahko le vi objavite spremembe na tej datoteki. Objave s strani drugih uporabnikov bo sistem zaustavil, vse dokler zaklepa ne sprostite. Zaklenjene datoteke v skladišču ni mogoče spremeniti na noben način, tako da se je ne da izbrisati ali preimenovati, razen če niste ste lastnik zaklepa.

Ni pa nujno, da uporabniki vedo, da ste datoteko zaklenili. Če statusa zaklepanja ne preverjajo redno, bodo to opazili šele, ko bo poskusili datoteko objaviti in objava ne bo uspešna. To pa ni najbolj uporabno. Da bi bilo delo z zaklepi lažje, je na voljo nova lastnost `svn:needs-lock`. Če je ta lastnost nastavljena (na katerokoli vrednost), bo krajevna kopija datoteke ob prevzemu ali posodobitvi na voljo samo za branje, *razen* če je ta delovna kopija lastnica zaklepa za to datoteko. To deluje kot opozorilo, da datoteke ne urejajte, preden zanj niste pridobili zaklepa. Datoteke pod nadzorom različic, ki so hkrati tudi označene samo za branje, imajo v programu TortoiseSVN posebno prekrivno ikono, ki vas opozarja, da morate pred urejanjem pridobiti zaklep.

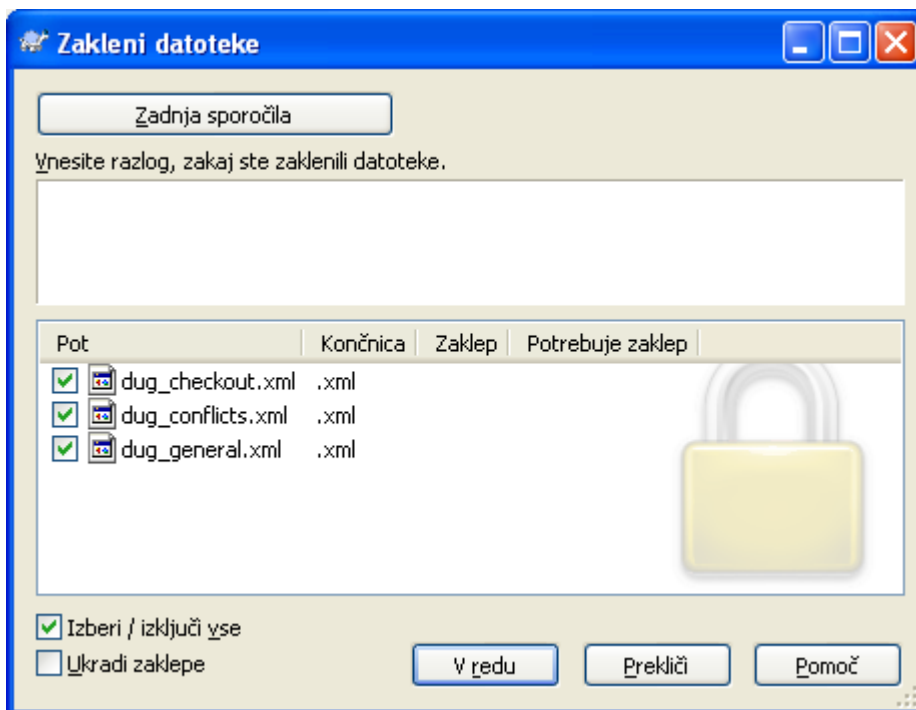
Zaklep se pridobi za določeno delovno kopijo in lastnika. Če imate v lasti več delovnih kopij (doma, v službi), ste lahko lastnik zaklepa datoteke samo v *eni* od teh delovnih kopij.

Če eden od vaših sodelavcev pridobi zaklep in odide na dopust, ne da bi zaklep sprostil, kaj storite? Subversion omogoča, da prisilno prevzamete zaklep. Sprostitev zaklepa nekoga drugega je znana pod imenom *prelom* zaklepa, prisilen prevzem zaklepa, ki je v lasti nekoga drugega, pa *kraja* zaklepa. Seveda teh ukazov ne smete zlorabljati, če želite ostati v dobrih odnosih s sodelavci.

Zaklepi so shranjeni v skladišču, žeton za zaklep pa je shranjen v krajevni delovni kopiji. Če se zgodi prelom zaklepa, krajevni zaklep postane neveljaven. Informacija v skladišču se vedno uporablja kot edina referenčna.

4.21.2. Pridobivanje zaklepa

V delovni kopiji izberite datoteke, za katere želite pridobiti zaklep, potem pa izberite TortoiseSVN → Dobi zaklep....



Slika 4.41. Okno zaklepov

Pojavi se pogovorno okno, ki vam omogoča vpis sporočila, tako da lahko ostali uporabniki vidijo, zakaj ste datoteko zaklenili. Komentar ni obvezen in se uporablja le v primeru uporabe strežnika svnservice. Če (in *samo* če) morate nekemu ukrasti zaklep, potrdite polje **Ukradi zaklep**, potem pa kliknite na gumb **V redu**.

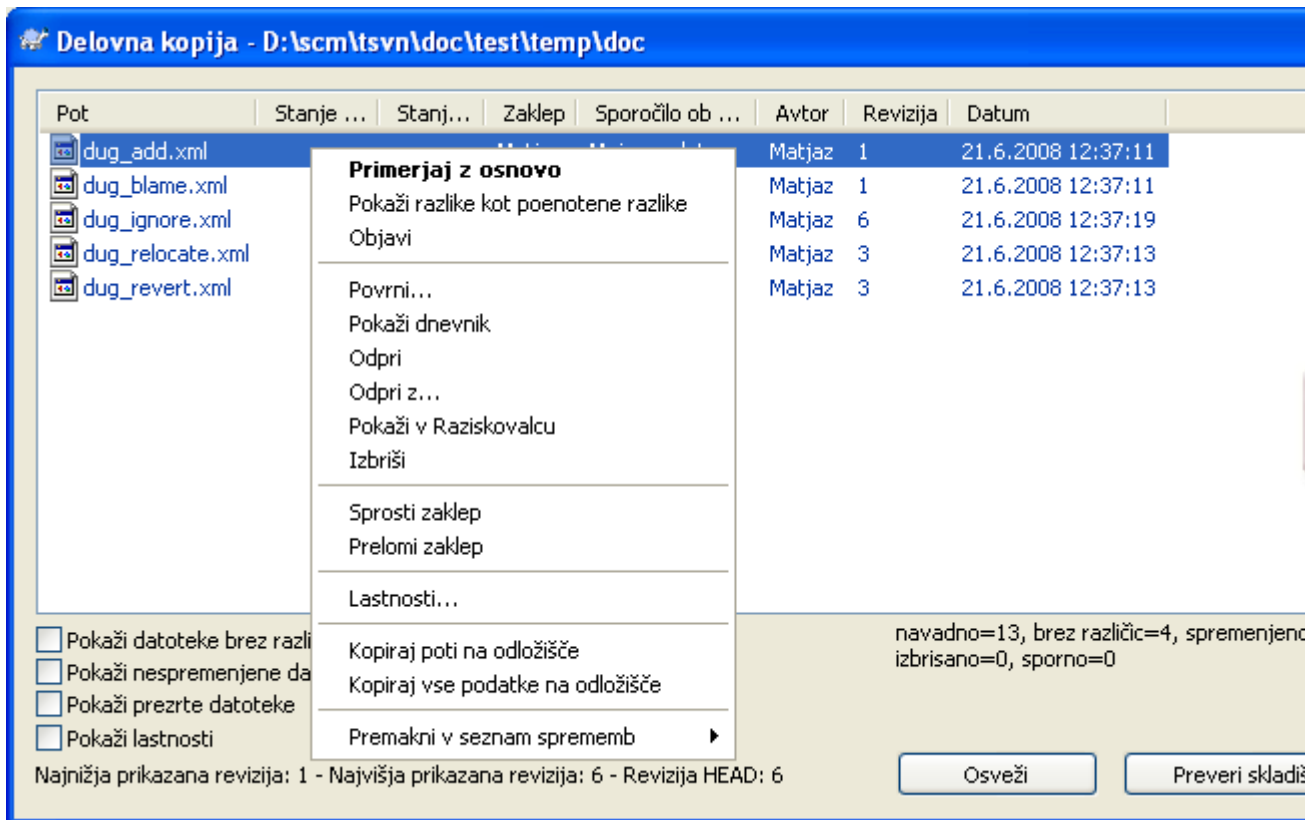
Če izberete mapo in potem uporabite TortoiseSVN → Dobi zaklep..., se bo pojavilo pogovorno okno Zaklep z *vsemi* datotekami v *vseh* podmapah. Če zares želite zakleniti celotni hierarhijo, je to pravi postopek, vendar lahko postanete zelo nepriljubljeni pri sodelavcih, če jim zaklenete celoten projekt. Bodite previdni...

4.21.3. Sprostitev zaklepa

Da ne bi pozabili sprostiti zaklepa, ki ga ne potrebujete več, so zaklenjene datoteke prikazane v oknu za objavo in izbrane. Če nadaljujete z objavo, se zaklepi na izbranih datotekah sprostijo, tudi če datotek niste spreminjali. Če določenih zaklepov ne želite sprostiti, ustreznih datotek ne označite (če jih niste spreminjali). Če želite obdržati zaklepe na datotekah, ki ste jih spreminjali, pred objavo potrdite polje **Obdrži zaklepe**.

Za ročno sproščanje zaklepa v delovni kopiji izberite ustrezne datoteke in izvedite ukaz TortoiseSVN → **Sprosti zaklep**. Ker dodatni podatki niso potrebni, TortoiseSVN kontaktira skladišče in sprosti zaklep. Ukaz lahko izvedete tudi na mapi. V tem primeru se sprostitvev izvede rekurzivno na vseh zaklenjenih datotekah.

4.21.4. Preverjanje stanja zaklepanja



Slika 4.42. Pogovorno okno Preveri posodobitve

Za preverjanje, katere zaklepe imate v lasti vi in ostali uporabniki, lahko uporabite ukaz TortoiseSVN → Preveri spremembe.... Zaklepi, ki so v vaši lasti, se pokažejo takoj. Za seznam zaklepov, ki jih imajo v lasti ostali uporabniki, uporabite gumb Preveri skladišče. S tem vidite tudi, če so kateri izmed vaših zaklepov prelomljeni ali ukradeni.

Iz kontekstnega menija lahko pridobite ali sprostite zaklep, prav tako pa lahko prelomite in ukradete zaklep, ki je v lasti drugih.



Izognite se prelamljanju in kraji zaklepov

Če nekomu prelomite ali ukradete zaklep, ne da bi uporabniku to povedali, lahko povzročite izgubo podatkov. Če delate z datotekami, ki se jih ne da spajati, in nekomu ukradete zaklep, lahko ta uporabnik po vaši sprostitvi zaklepa objavi svojo različico datoteke in s tem povozi vašo. Subversion nikoli ne izgubi podatkov, izgubite pa lahko varnost dela v teamu, ki vam jo daje zaklepanje.

4.21.5. Nastavite nezaklenjene datoteke samo za branje

Kot je že bilo omenjeno, je najbolj učinkovita metoda za uporabo zaklepanja uporaba lastnosti `svn:needs-lock`. Za več informacij o tem, kako datotekam nastaviti lastnosti, preberite [Razdelek 4.17, "Nastavitve projekta"](#). Datoteke, ki imajo nastavljeno to lastnost, bodo ob prevzemu ali posodobitvi vedno dobile status datoteke samo za branje, razen če je delovna kopija lastnica zaklepa.



Kot opomnik, da je datoteka namenjena samo za branje, uporablja TortoiseSVN posebno prekrivno ikono.

Če velja na projektu pravilo, da je potrebno vsako datoteko zakleniti, potem bo verjetno lažje, da uporabite samodejne lastnosti (auto-props). Tako bo ta lastnost samodejno pripeta vsaki novo dodani datoteki. Za nadaljnje informacije preberite [Razdelek 4.17.1.5, "Samodejna nastavitve lastnosti"](#).

4.21.6. Ukazna datoteka akcije za zaklepanje

Ko ustvarite novo skladišče s sistemom Subversion 1.2 ali novejšim, se v podmapi `hooks` ustvarijo štiri predloge za ukazne datoteke akcij. Kličejo se pred in po pridobivanju zaklepa ter pred in po sprostitvi zaklepa.

Na strežniku je priporočljivo nastaviti ukazne datoteke akcij za `post-lock` in `post-unlock`, ki pošljejo elektronsko pošto, ko se datoteka zaklene. S pomočjo teh datotek so vsi uporabniki obveščeni, ko nekdo zaklene/odklene datoteko. Primer ukazne datoteke se nahaja v datoteki `hooks/post-lock.tmpl` v mapi skladišča.

Nastavite lahko tudi ukazno datoteko akcije, ki preprečuje prelom ali krajo zaklepa, ali pa te operacije dovoli le administratorju. Lahko pa nastavite pošiljanje elektronske pošte lastniku zaklepa, kadar mu nekdo zaklep prelomi ali ukrade.

Za dodatne informacije preberite [Razdelek 3.3, "Server side hook scripts"](#).

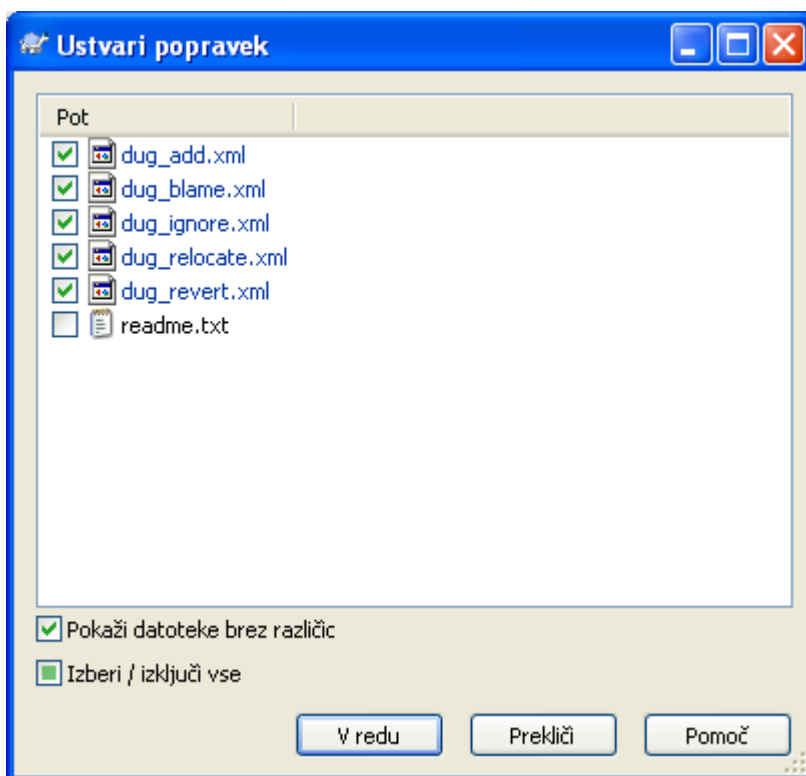
4.22. Ustvarjanje in nameščanje popravkov

V primeru odprtokodnih projektov (kot je tale) imajo pravice branja skladišča vsi in vsakdo lahko prispeva k projektu. Kako pa se ti prispevki nadzorujejo? Če bi lahko vsakdo objavljaj spremembe, bi projekt kmalu postal nestabilen in neuporaben. V takšnem primeru se spremembe pošljejo razvojni ekipi, ki ima pravice za objavljanje, v obliki datoteke *popravkov*. Ekipa popravek pregleda in ga objavi v skladišču ali pa vrne avtorju.

Datoteke popravkov so poenotene razlike datotek, ki prikazujejo razlike med vašo delovno kopijo in osnovno revizijo.

4.22.1. Ustvarjanje datoteke popravkov

Najprej morate spremembe narediti *in jih preizkusiti*. Potem namesto ukaza TortoiseSVN → Objavi... na nadrejeni mapi uporabite TortoiseSVN → Ustvari popravek...



Slika 4.43. Okno za ustvarjanje popravkov

Sedaj izberete datoteke, ki jih želite vključiti v popravek, prav tako, kot to storite ob objavi. Ustvari se ena datoteka, ki vsebuje podatke o vseh spremembah, ki ste jih naredili na izbranih datotekah od zadnje posodobitve iz skladišča.

Stolpce v tem oknu lahko prilagodite enako kot stolpce v oknu **Preveri spremembe**. Za več informacij preberite [Razdelek 4.7.3, "Krajevno in oddaljeno stanje"](#).

Izdelate lahko več popravkov, ki vsebujejo spremembe različnih skupin datotek. Seveda, če izdelate datoteko popravkov, nato naredite še dodatne spremembe in izdelate novo datoteko popravkov, bo ta vsebovala spremembe *obeh* skupin sprememb.

Shranite datoteko s poljubnim imenom. Datoteke popravkov imajo lahko poljubno končnico, vendar imajo po konvenciji končnico `.patch` ali `.diff`. Sedaj lahko datoteko oddate.

Popravek lahko shranite tudi v odložišče namesto v datoteko. Nato ga lahko prilepite v elektronsko sporočilo in ga pošljete v pregled ostalim. Če imate dve delovni kopiji na enem računalniku in želite prenesti spremembe iz ene kopije v drugo, je kopiranje popravka v odložišče pripraven način za izvedbo operacije.

4.22.2. Nameščanje datoteke popravkov

Datoteke popravkov se namestijo v delovno kopijo. Nameščanje je potrebno narediti na istem nivoju map, na katerem je bil popravek narejen. Če ne veste, kateri nivo to je, pogledajte v prvo vrstico datoteke popravkov. Primer: če je prva datoteka popravka `doc/source/english/chapter1.xml` in je prva vrstica popravka `Index: english/chapter1.xml`, potem morate popravek namestiti na mapo `doc/source/`. Vendar, če le delate na pravi delovni kopiji, bo TSVN zaznal, da ste izbrali napačen nivo in vam bo predlagal pravi nivo.

Da bi namestili datoteko popravkov na delovni kopiji, morate imeti v skladišču vsaj pravice za branje. Razlog je v tem, da mora imeti program za spajanje referenco na revizijo, ki jo je oddaljeni programer uporabil kot osnovo.

Iz kontekstnega menija v mapi izberite TortoiseSVN → Namesti popravek... Prikazalo se bo pogovorno okno za izbiro datoteke popravkov, ki jo želite namestiti. Prikazane so le datoteke s končnicami `.patch` in `.diff`, vendar pa lahko prikažete tudi "Vse datoteke". Če ste popravek shranili na odložišče, lahko v oknu uporabite gumb Odpri iz odložišča....

Če ima datoteka popravkov končnico `.patch` ali `.diff`, lahko najno kliknete z desnim gumbom in izberete TortoiseSVN → Namesti popravek... V tem primeru boste povprašani po lokaciji delovne kopije.

Ti dve metodi omogočata isti postopek na različna načina. S prvo metodo izberete delovno kopijo in poiščete datoteko popravkov, z drugo pa izberete datoteko popravkov in poiščete delovno kopijo.

Ko ste izbrali datoteko popravkov in lokacijo delovne kopije, TortoiseMerge začne postopek spajanja sprememb iz datoteke popravkov v delovno kopijo. Majhno okno vsebuje seznam datotek, ki so bile spremenjene. Dvokliknite na vsako izmed njih, preglejte spremembe in shranite spojene datoteke.

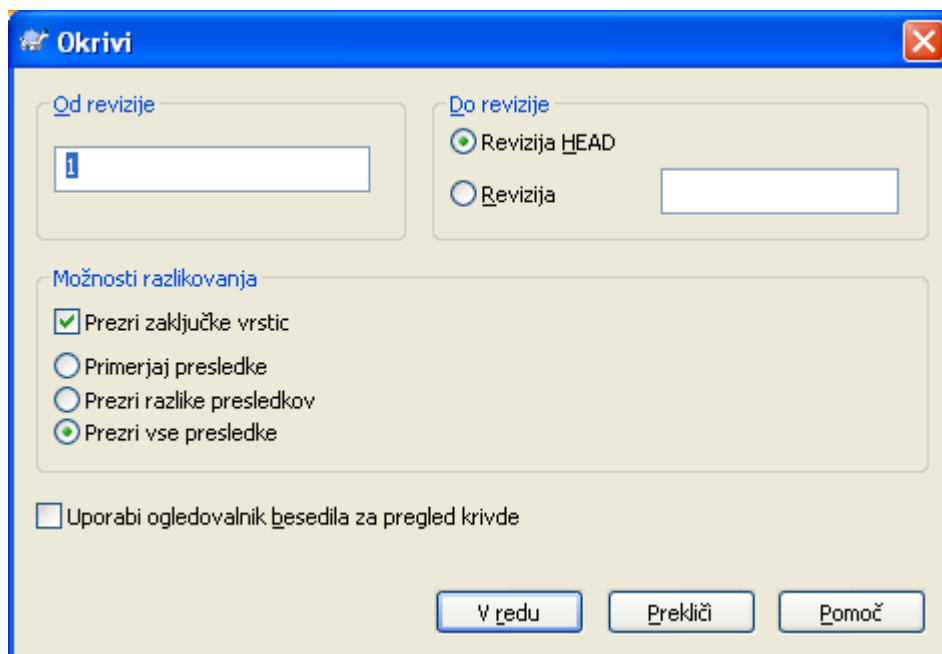
Popravek oddaljenega sodelavca je sedaj nameščen na vaši delovni kopiji, ki jo morate objaviti, da bodo lahko vsi uporabniki dostopali do sprememb v skladišču.

4.23. Kdo je spremenil posamezno vrstico?

Včasih vas ne zanima samo, katere vrstice so bile spremenjene, ampak tudi, kdo je določene vrstice spremenil. Takrat je uporaben ukaz TortoiseSVN → Okrivi.... Včasih ga imenujemo tudi *pohvali*.

Ta ukaz za vsako vrstico v datoteki izpiše avtorja in številko revizije, v kateri je bila vrstica spremenjena.

4.23.1. Okrivi datoteke



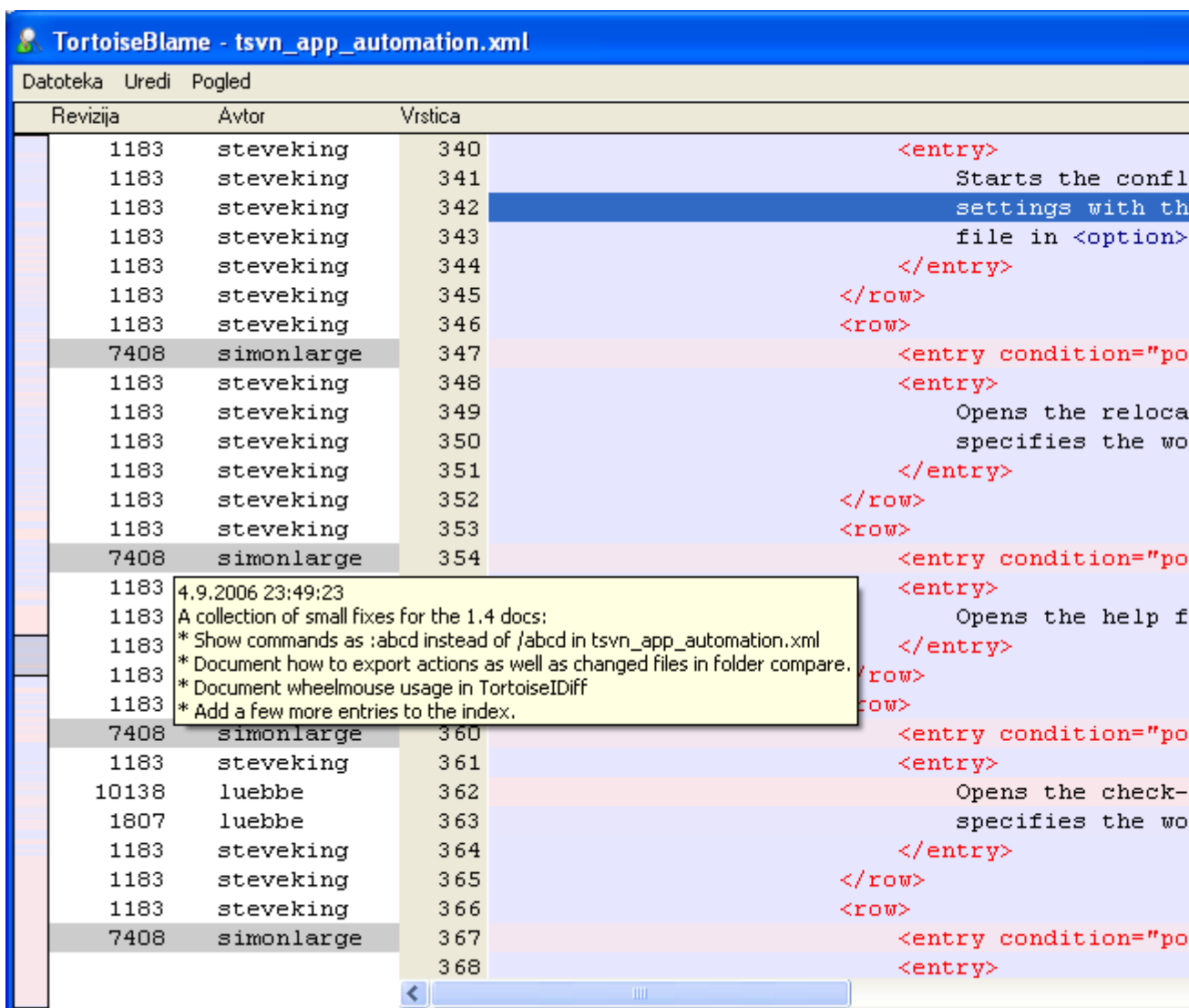
Slika 4.44. Okno hvali/okrivi

Če vas spremembe v zgodnejših revizijah ne zanimajo, lahko nastavite številko revizije, s katero naj se okrivljanje začne. Nastavite vrednost 1, če želite okrivljanje *vseh* revizij.

Po privzetih nastavitvah se krivda prikazuje s programom *TortoiseBlame*, ki naglasi različne revizije, kar vam olajša branje. Če želite informacije o krivdi natisniti ali urediti, izberite Uporabi ogledovalnik besedila za pregled krivde

Nastavite lahko, kako se obravnavajo zaključki vrstic in presledki. Možnosti so opisane v [Razdelek 4.10.2, "Nastavitev zaključkov vrstic in presledkov"](#). Privzeta nastavitev je, da se zaključki vrstic in presledki obravnavajo kot vse ostale spremembe. Če pa želite zamikanje prezreti in poiskati izvirnega avtorja, lahko tu nastavite ustrezno možnost.

Ko pritisnete na gumb **V redu**, začne TortoiseSVN pridobivati podatke, da ustvari datoteko krivd. Pazite: to lahko traja kar nekaj minut, odvisno od tega, koliko se je datoteka spreminjala in seveda od povezave do strežnika skladišča. Po končanem procesu se rezultat zapiše v začasno datoteko in prikaže na zaslону.



Slika 4.45. TortoiseBlame

TortoiseBlame, ki je del paketa TortoiseSVN, olajša pregledovanje datoteke krivd. Ko z miško nekaj trenutkov mirujete na vrstici v stolcu z informacijami, se vse vrstice iste revizije prikažejo s temnejšim ozadjem. Vrstice drugih revizij, ki jih je spremenil isti avtor, so prikazane s svetlim ozadjem. Barvno označevanje ne bo delovalo dobro, če uporabljate le 256 barv.

Če na vrstico kliknete z levim gumbom, se označijo vse vrstice iste revizije, vrstice, ki jih je spremenil isti avtor, pa so označene s svetlejšo barvo. Označevanje obvelja tudi če premaknete miško. Če želite označevanje izključiti, še enkrat kliknite na vrstico.

Komentar revizije (sporočilo dnevniškega zapisa) je prikazan v oknu z namigi, ko z miško za kratko počakate na stolpcu s krivdnimi informacijami. Če želite prekopirati dnevniško sporočilo za to revizijo, uporabite kontekstni meni, ki se pojavi ob desnem kliku.

Iskanje po krivdni datoteki izvedete z ukazom **Uredi** → **Najdi...** Iščete lahko po številki revizije, avtorju in po sami vsebini datoteke. Sporočila dnevniških zapisov niso vključena v iskanje - za to uporabite Dnevnik.

Z ukazom **Uredi** → **Pojdi na vrstico...** skočite na določeno vrstico v datoteki.

Ko ste z miško nad stolcem s krivdnimi informacijami, lahko uporabite kontekstni meni, ki vam omogoča primerjanje revizij in pregled zgodovine. Številka revizije vrstice pod kazalcem miške se uporabi kot referenca. Ukaz **Kontekstni meni** → **Okrivi prejšnjo revizijo** ustvari krivdno poročilo za isto datoteko, vendar uporabi predhodno revizijo kot zgornjo mejo. To vam da krivdno poročilo za stanje datoteke točno preden se je vrstica, ki jo gledate, zadnjič spremenila. Ukaz **Kontekstni meni** → **Pokaži spremembe** požene pregledovalnik razlik, ki prikazuje, kaj se je spremenilo v referenčni reviziji. Ukaz **Kontekstni meni** → **Prikaži dnevnik** prikaže dnevnik, ki se začne z referenčno revizijo.

Če potrebujete boljše vizuelno razlikovanje med starimi in novimi spremembami, izberite **View** → **Prikaz starosti vrstic z barvami**. Novejše vrstice bodo prikazane z rdečimi barvnimi odtenki, starejše pa z modrimi. Privzete barve so dokaj svetle, vendar jih lahko spremenite v nastavitvah za TortoiseBlame.

If you are using Merge Tracking, where lines have 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. If you do not want merged lines shown in this way, uncheck the **Include merge info** checkbox.

Če želite videti poti, ki so predmet spajanja, izberite **Pogled** → **Poti spajanja**.

Nastavitve za TortoiseBlame prikažete z izbiro **TortoiseSVN** → **Nastavitve...** na zavihku TortoiseBlame. Preberite [Razdelek 4.30.9, "Nastavitve TortoiseBlame"](#).

4.23.2. Okrivi spremembe

Ena od omejitev krivdnega poročila je, da vam prikaže datoteko, kot je izgledala v določeni reviziji, in avtorja zadnje spremembe posamezne vrstice. Včasih želite izvedeti, kakšna sprememba je bila narejena in kdo jo je naredil. Potrebujete kombinacijo razlikovanja in okrivljanja.

Dnevnik ponuja več možnosti, ki vam to omogočajo.

Okrivi revizije

V zgornjem delu izberite dve reviziji in poženite **Kontekstni meni** → **Okrivi revizije**. S tem boste prenesli krivdne podatke za dve reviziji, potem pa vam bo pregledovalnik razlik prikazal razlike med obema krivdnima datotekama.

Okrivi spremembe

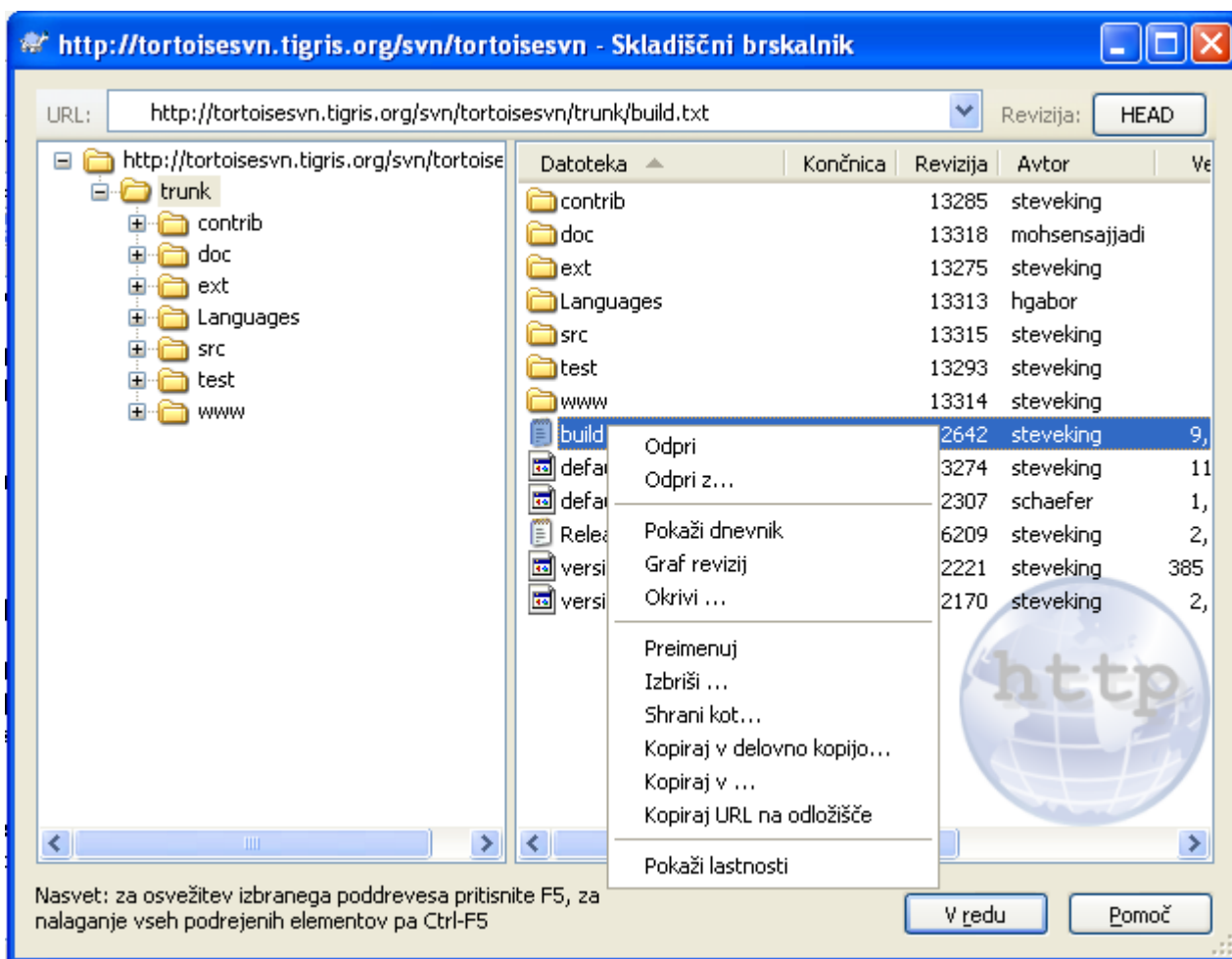
V zgornjem delu izberite revizijo, potem pa v spodnjem delu izberite datoteko in izberite **Kontekstni meni** → **Okrivi spremembe**. S tem boste prenesli krivdne podatke za izbrano revizijo in prejšnjo revizijo, potem pa vam bo pregledovalnik razlik prikazal razlike med krivdnima datotekama.

Primerjaj in okrivi z delovno osnovo BASE

Prikažite dnevnik za eno datoteko in v zgornjem delu izberite revizijo, potem pa poženite **Kontekstni meni** → **Primerjaj in okrivi z delovno osnovo**. S tem prenesete krivdne podatke za izbrano revizijo in za datoteko v delovni osnovi, potem pa se prikaže pregledovalnik razlik in primerja obe krivdni datoteki.

4.24. Brskalnik po skladišču

Včasih morate nekaj postoriti neposredno v skladišču, brez uporabe delovne kopije. Za to se uporablja *brskalnik po skladišču*. Prav tako kot vam Raziskovalec in prekrivne ikone omogočajo pregled delovne kopije, vam brskalnik po skladišču omogoča vpogled v strukturo in stanje skladišča.



Slika 4.46. Brskalnik po skladišču

Z brskalnikom po skladišču lahko izvajate kopiranje, premikanje, preimenovanje... neposredno v skladišču.

Brskalnik po skladišču je zelo podoben Raziskovalcu, s to razliko, da prikazuje vsebino skladišča v določeni reviziji namesto datotek na vašem računalniku. V levem delu lahko vidite strukturo map v obliki drevesa, v desnem pa vsebino izbrane mape. Na vrhu okna lahko vnesete naslov URL skladišča in revizijo, ki jo želite videti.

Tako kot v Raziskovalcu lahko tudi tu kliknete na glave stolpcev na desni strani, če želite definirati vrstni red razvrščanja. In prav tako kot v Raziskovalcu so v obeh delih na voljo kontekstni meniji.

Kontekstni meni za datoteko vam omogoča, da:

- Odprete izbrano datoteko s privzetim pregledovalnikom ali programom po lastni izbiri.
- Shranite kopijo datoteke brez različic na trdi disk.
- Prikažete dnevnik za datoteko ali graf vseh revizij, tako da lahko vidite, od kje datoteka prihaja
- Okrivite datoteko, da vidite, kdo in kdaj je spremenil določeno vrstico.
- Izbrišete ali preimenujete datoteko
- Ustvarite kopijo datoteke - v drug del skladišča ali v delovno kopijo, ki kaže na isto skladišče.
- Pogledate/uredite lastnosti datoteke

Kontekstni meni za mapo vam omogoča, da:

- Prikažete dnevnik za mapo ali graf vseh revizij, da vidite, od kje mapa prihaja.
- Izvozite mapo v mapo brez različic na trdem disku.
- Prevzamete mapo in si s tem ustvarite delovno kopijo na trdem disku.
- Ustvarite novo mapo v skladišču.
- Dodate datoteke ali mape neposredno v skladišče.
- Izbrišete ali preimenujete mapo.
- Ustvarite kopijo mape - v drug del skladišča ali v delovno kopijo, ki kaže na isto skladišče.
- Pogledate/uredite lastnosti mape
- Označite mapo za primerjanje. Označena mapa je izpisana polkrepko.
- Primerjate mapo s predhodno označeno mapo - kot poenoteno razliko ali kot seznam spremenjenih datotek, katerih razlike lahko vizuelno pregledate s privzetim orodjem za razlikovanje. To je uporabno pri primerjanju dveh oznak ali pri primerjanju veje in glavne veje, da vidite, kaj se je spremenilo.

Če na desni izberete dve mapi, lahko pregledate razlike med njima - kot poenoteno razliko ali kot seznam datotek, katerih razlike lahko vizuelno pregledate s privzetim orodjem za razlikovanje.

Če na desni strani izberete več map, jih lahko prevzamete sočasno v skupno mapo.

Če izberete dve oznaki, ki sta skopirani iz iste korenske mape (ponavadi /`trunk/`), lahko uporabite Kontekstni meni → Pokaži dnevnik... in s tem prikazete seznam revizij med obema oznakama.

Kot je to običajno, lahko uporabite tipko **F5** za osvežitev pogleda. S tem osvežite vse, kar je trenutno prikazano. Če želite v naprej prenesti ali osvežiti informacije o vozliščih, ki še niso bila odprta, uporabite **Ctrl-F5**. Po tem bo razširjanje kateregakoli vozlišča potekalo zelo hitro, saj ne bo zamud zaradi prenosa podatkov preko omrežja.

Brskalnik po skladišču lahko uporabite tudi za izvajanje operacij z metodo povleci-in-spusti. Če povlečete v brskalnik po skladišču mapo iz Raziskovalca, jo s tem uvozite. Upoštevajte, da se v primeru, ko povlečete več elementov naenkrat, elementi uvozijo v ločenih objavah.

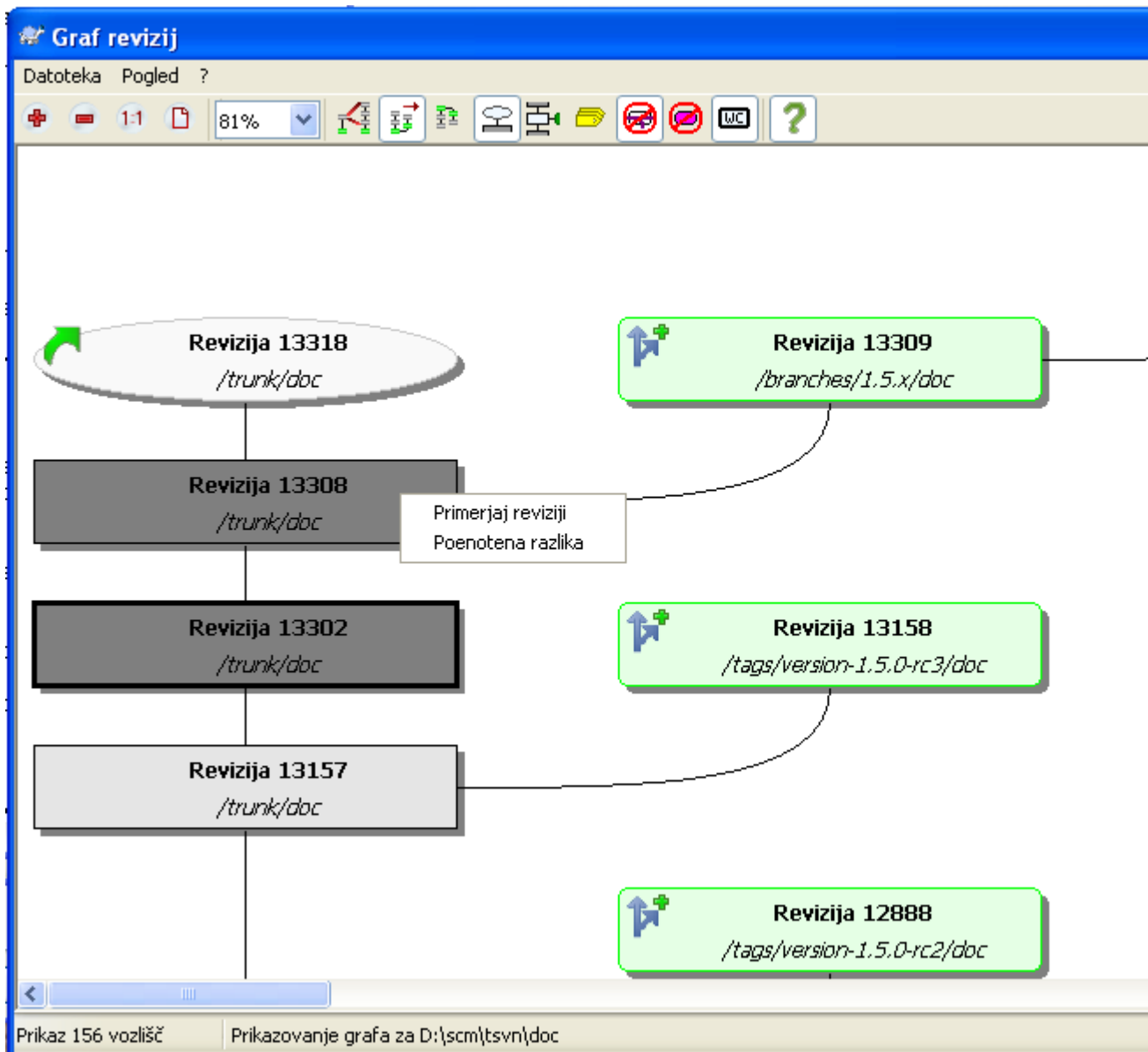
Če želite elemente premikati znotraj skladišča, jih enostavno povlecite z levim gumbom na novo lokacijo. Če želite namesto premikanja ustvariti kopijo, ob potegu držite pritisnjeno tipko **CTRL**. Pri kopiranju ima kazalnik dodan znak "plus", prav tako kot v Raziskovalcu.

Če želite datoteko ali mapo premakniti/prekopirati na novo lokacijo in jo istočasno preimenovati, jo lahko povlečete z desnim gumbom (lahko držite pritisnjeno tipko **CTRL**) namesto z levim gumbom. V tem primeru se pokaže okno se vpis novega imena datoteke ali mape.

Ko ste naredili spremembe v skladišču z uporabo ene od teh metod, se pojavi okno za vnos sporočila dnevnikega zapisa. Če ste kaj premaknili po pomoti, imate sedaj možnost, da operacijo prekličete.

Včasih se pri odpiranju poti namesto vsebine prikaže sporočilo o napaki. Vzrok je lahko napačen naslov URL, nezadostne pravice ali težave s strežnikom. Če želite sporočilo o napaki prekopirati (da ga vključite v elektronsko sporočilo), iz kontekstnega menija izberite Kontekstni meni → Kopiraj sporočilo o napaki na odložišče ali uporabite **Ctrl+C**.

4.25. Grafi revizij



Slika 4.47. Graf revizije

Včasih želite izvedeti, od kje izhajajo veje in oznake in najlažji način, da to izveste, je graf revizije ali drevesne strukture. Takrat uporabite TortoiseSVN → Graf revizij...

Ta ukaz analizira zgodovino revizij in poskuša ustvariti drevo, ki prikazuje, kje so se zgodila kopiranja oziroma kje so bile veje/oznake izbrisane.



Pomembno

Za generiranje grafa mora TortoiseSVN iz strežnika prenesti dnevniške zapise vseh revizij. Verjetno ni potrebno poudarjati, da lahko pri skladiščih z nekaj tisoč revizijami to traja kar nekaj minut, odvisno tudi od hitrosti strežnika, pasovne širine omrežja... Če poskusite izdelati graf revizij na projektu, kot je denimo *Apache*, ki ima trenutno več kot pol milijona revizij, boste čakali kar lep čas.

The good news is that if you are using log caching, you only have to suffer this delay once. After that, log data is held locally. Log caching is enabled in TortoiseSVN's settings.

4.25.1. Vozlišča grafa revizij

Each revision graph node represents a revision in the repository where something changed in the tree you are looking at. Different types of node can be distinguished by shape and colour. The shapes are fixed, but colours can be set using TortoiseSVN → Settings

Added or copied items

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

Deleted items

Deleted items eg. a branch which is no longer required, are shown using an octagon (rectangle with corners cut off). The default colour is red.

Renamed items

Renamed items are also shown using an octagon, but the default colour is blue.

Revizija na vrhu veje

The graph is normally restricted to showing branch points, but it is often useful to be able to see the respective HEAD revision for each branch too. If you select **Show HEAD revisions**, each HEAD revision nodes will be shown as an ellipse. Note that HEAD here refers to the last revision committed on that path, not to the HEAD revision of the repository.

Working copy revision

If you invoked the revision graph from a working copy, you can opt to show the BASE revision on the graph using **Show WC revision**, which marks the BASE node with a bold outline.

Modified working copy

If you invoked the revision graph from a working copy, you can opt to show an additional node representing your modified working copy using **Show WC modifications**. This is an elliptical node with a bold outline in red by default.

Normal item

Vsi drugi elementi so prikazani s pravokotnikom

Note that by default the graph only shows the points at which items were added, copied or deleted. Showing every revision of a project will generate a very large graph for non-trivial cases. If you really want to see *all* revisions where changes were made, there is an option to do this in the **View** menu and on the toolbar.

The default view (grouping off) places the nodes such that their vertical position is in strict revision order, so you have a visual cue for the order in which things were done. Where two nodes are in the same column the order is very obvious. When two nodes are in adjacent columns the offset is much smaller because there is no need to prevent the nodes from overlapping, and as a result the order is a little less obvious. Such optimisations are necessary to keep complex graphs to a reasonable size. Please note that this ordering uses the *edge* of the node on the *older* side as a reference, i.e. the bottom edge of the node when the graph is shown with oldest node at the bottom. The reference edge is significant because the node shapes are not all the same height.

4.25.2. Spreminjanje pogleda

Ker je graf revizij pogosto kar zapleten, imate na voljo kar nekaj možnosti, da ga priredite po svojem okusu. Te možnosti so na voljo v meniju **Pogled** in v orodni vrstici.

Razdeljevanje vej v skupine

The default behavior (grouping off) has all rows sorted strictly by revision. As a result, long-living branches with sparse commits occupy a whole column for only a few changes and the graph becomes very broad.

Ta način daje v skupine spremembe po vejah, tako da splošnega razvrščanja po revizijah ni. Zaporedne revizije na veji bodo prikazane v zaporednih vrsticah. Podveje so urejene tako, da se kasnejše veje prikažejo v istem stolpcu nad starejšimi vejami. Tako ostane graf ozek. Rezultat tega je, da lahko vrstica vsebuje spremembe različnih revizij.

Starejše na vrhu

Običajno graf prikazuje starejše revizije na dnu, drevo pa potem raste navzgor. Z vključitvijo te možnosti graf raste od vrha navzdol.

Align trees on top

When a graph is broken into several smaller trees, the trees may appear either in natural revision order, or aligned at the bottom of the window, depending on whether you are using the Group Branches option. Use this option to grow all trees down from the top instead.

Zmanjšaj križanje črt

If the layout of the graph has produced a lot of crossing lines, use this option to clean it up. This may make the layout columns appear in less logical places, for example in a diagonal line rather than a column, and the graph may require a larger area to draw.

Differential path names

Long path names can take a lot of space and make the node boxes very large. Use this option to show only the changed part of a path, replacing the common part with dots. E.g. if you create a branch `/branches/1.2.x/doc/html` from `/trunk/doc/html` the branch could be shown in compact form as `/branches/1.2.x/..` because the last two levels, `doc` and `html`, did not change.

Show all revisions

This does just what you expect and shows every revision where something (in the tree that you are graphing) has changed. For long histories this can produce a truly huge graph.

Pokaži revizijo HEAD

S to možnostjo zagotovite, da je najnovejša revizija vsake veje vedno prikazana na grafu.

Natančni izvori kopiranja

When a branch/tag is made, the default behaviour is to show the branch as taken from the last node where a change was made. Strictly speaking this is inaccurate since the branches are often made from the current HEAD rather than a specific revision. So it is possible to show the more correct (but less useful) revision that was used to create the copy. Note that this revision may be younger than the HEAD revision of the source branch.

Zloži oznake

When a project has many tags, showing every tag as a separate node on the graph takes a lot of space and obscures the more interesting development branch structure. At the same time you may need to be able to access the tag content easily so that you can compare revisions. This option hides the nodes for tags and shows them instead in the tooltip for the node that they were copied from. A tag icon on the right side of the source node indicates that tags were made.

Hide deleted paths

Hides paths which are no longer present at the HEAD revision of the repository, e.g. deleted branches.

Hide unchanged branches

Hides branches where no changes were committed to the respective file or sub-folder. This does not necessarily indicate that the branch was not used, just that no changes were made to *this* part of it.

Show WC revision

Marks the revision on the graph which corresponds to the update revision of the item you fetched the graph for. If you have just updated, this will be HEAD, but if others have committed changes since your last update your WC may be a few revisions lower down. The node is marked by giving it a bold outline.

Show WC modifications

If your WC contains local changes, this option draws it as a separate elliptical node, linked back to the node that your WC was last updated to. The default outline colour is red. You may need to refresh the graph using **F5** to capture recent changes.

Filter

Včasih graf revizij prikaže več podrobnosti, kot si želite. Ta možnost odpre pogovorno okno, ki omogoča omejevanje prikazanega območja revizij in skrivanje določenih poti po imenu.

Tree stripes

Where the graph contains several trees, it is sometimes useful to use alternating colours on the background to help distinguish between trees.

Show overview

Shows a small picture of the entire graph, with the current view window as a rectangle which you can drag. This allows you to navigate the graph more easily. Note that for very large graphs the overview may become useless due to the extreme zoom factor and will therefore not be shown in such cases.

4.25.3. Uporaba grafa revizij

Za lažjo navigacijo po grafu revizij uporabite predogledno okno. Ta prikaže celoten graf v majhnem oknu, pri tem pa je trenutno prikazan del posebej označen. Označen del lahko premikate in s tem spremenite pogled grafa.

Datum revizije, avtor in komentarji so prikazani v namigu, ko miško pridržite na polju revizije.

Če izberete dve reviziji (uporabite **CTRL**-levi klik), lahko uporabite kontekstni meni, da prikažete razlike med revizijama. Prikažete lahko tudi razlike med začetnimi točkami vej, vendar običajno prikazujemo razlike med končnimi točkami vej (revizije HEAD).

Razlike lahko prikažete kot datoteko poenotene različice, ki prikaže vse razlike v eni datoteki z minimalno vsebino. Če izberete **Kontekstni meni** → **Primerjaj reviziji**, se vam prikaže seznam spremenjenih datotek. Dvokliknite na datoteko, da prenesete obe reviziji datoteke in ju primerjate z grafičnim orodjem za razlikovanje.

Če kliknete z desnim gumbom na revizijo, lahko uporabite **Kontekstni meni** → **Pokaži dnevnik**.

You can also merge changes in the selected revision(s) into a different working copy. A folder selection dialog allows you to choose the working copy to merge into, but after that there is no confirmation dialog, nor any opportunity to try a test merge. It is a good idea to merge into an unmodified working copy so that you can revert the changes if it doesn't work out! This is a useful feature if you want to merge selected revisions from one branch to another.



Learn to Read the Revision Graph

First-time users may be surprised by the fact that the revision graph shows something that does not match the user's mental model. If a revision changes multiple copies or branches of a file or folder, for instance, then there will be multiple nodes for that single revision. It is a good practice to start with the leftmost options in the toolbar and customize the graph step-by-step until it comes close to your mental model.

All filter options try lose as little information as possible. That may cause some nodes to change their color, for instance. Whenever the result is unexpected, undo the last filter operation and try to understand what is special about that particular revision or branch. In most cases, the initially expected outcome of the filter operation would either be inaccurate or misleading.

4.25.4. Osveževanje pogleda

Če želite preveriti, ali obstajajo na strežniku novejša informacije, lahko pogled osvežite z uporabo tipke **F5**. Če uporabljate predpomnilnik dnevnika (privzeta nastavitve), s tem preverite, ali se na skladišču nahajajo nove objave, in prenesete le najnovejše zapise. Če ste pred delali v nepovezanem načinu, bo TortoiseSVN poskušal vzpostaviti povezavo.

Če uporabljate predpomnilnik dnevnika in menite, da se je sporočilo dnevnikega zapisa ali avtor spremenil, uporabite okno dnevnika za osvežitev željenih sporočil. Ker se graf revizij sestavlja iz korena skladišča, bi bilo potrebno preveriti celoten predpomnilnik. Posodabljanje bi trajalo *zelo* dolgo.

4.25.5. Pruning Trees

A large tree can be difficult to navigate and sometimes you will want to hide parts of it, or break it down into a forest of smaller trees. If you hover the mouse over the point where a node link enters or leaves the node you will see one or more popup buttons which allow you to do this.



Click on the minus button to collapse the attached sub-tree.



Click on the plus button to expand a collapsed tree. When a tree has been collapsed, this button remains visible to indicate the hidden sub-tree.



Click on the cross button to split the attached sub-tree and show it as a separate tree on the graph.

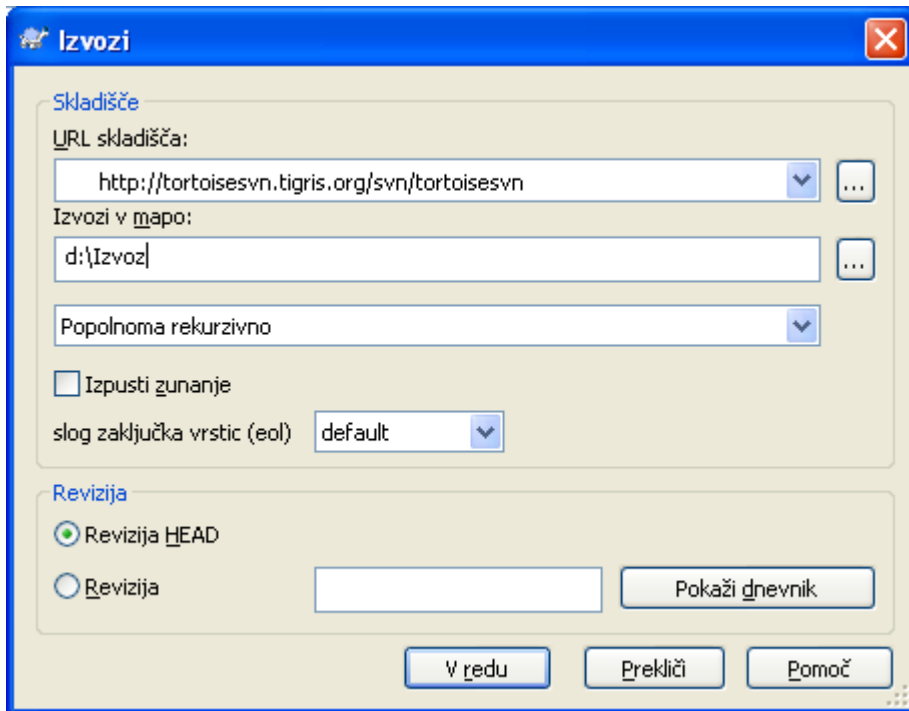


Click on the circle button to reattach a split tree. When a tree has been split away, this button remains visible to indicate that there is a separate sub-tree.

Click on the graph background for the main context menu, which offers options to **Expand all** and **Join all**. If no branch has been collapsed or split, the context menu will not be shown.

4.26. Izvažanje delovne kopije sistema Subversion

Včasih potrebujete delovno kopijo brez map `.svn`, n. pr. zato, da naredite paket izvorne kode ali da jo izvozite na spletni strežnik. Namesto ročnega kopiranja in brisanja map `.svn` TortoiseSVN ponuja ukaz TortoiseSVN → Izvozi... Izvažanje neposredno iz naslova URL in izvažanje iz delovne kopije se nekoliko razlikujeta.



Slika 4.48. Okno Uvoz-iz-URL

Če izvedete ta ukaz na mapi brez različic TortoiseSVN predpostavlja, da je izbrana mapa cilj, in odpre pogovorno okno za vnos naslova URL in številke revizije za izvoz. Okno omogoča izvoz samo vrhnje mape, omogoča izpustitev zunanjih referenc, prav tako pa omogoča spreminjanje sloga zaključevanja vrstic pri datotekah, ki imajo nastavljeno lastnost `svn:eol-style`.

Seveda lahko izvažate tudi neposredno iz skladišča. Z brskalnikom po skladišču pojdite na ustrezno poddrevo skladišča in uporabite Kontekstni meni → Izvozi. Prikazalo se bo okno Izvoz iz URL, kot je to opisano zgoraj.

Če izvedete ta ukaz na delovni kopiji, vas TortoiseSVN vpraša, kam naj shrani *čisto* delovno kopijo brez map `.svn`. Po privzetih nastavitvah se izvozijo le datoteke pod nadzorom različic, vendar lahko potrdite polje Izvozi tudi datoteke brez različic, da izvozite tudi datoteke brez različic, ki obstajajo v delovni kopiji, ne pa tudi v skladišču. Po želji lahko izpustite zunanje reference (z nastavljeno lastnostjo `svn:externals`).

Druga možnost za izvoz delovne kopije je, da vrhno mapo delovne kopije povlečete z desnim gumbom na drugo lokacijo in izberete Kontekstni meni → SVN - Izvozi sem ali Kontekstni meni → SVN - Izvozi vse sem. Zadnja možnost izvozi tudi datoteke brez različic.

Pri izvažanju iz delovne kopije se v primeru, da v ciljni mapi že obstaja datoteka z enakim imenom kot tista, ki jo izvažate, pojavi okno in vam ponudi možnost, da obstoječo vsebino prepisete ali da se ustvari mapa s samodejno določenim imenom, n. pr. Cilj (1).



Izvažanje posameznih datotek

Pogovorno okno za izvažanje ne dovoli izvažanja posameznih datotek, čeprav je Subversion to zmožen narediti.

Za izvoz posamezne datoteko s programom TortoiseSVN, morate uporabiti brskalnik po skladišču (Razdelek 4.24, "Brskalnik po skladišču"). Datoteko, ki jo želite izvoziti, preprosto odvedite v Raziskovalca ali pa uporabite kontekstni meni.



Exporting a Change Tree

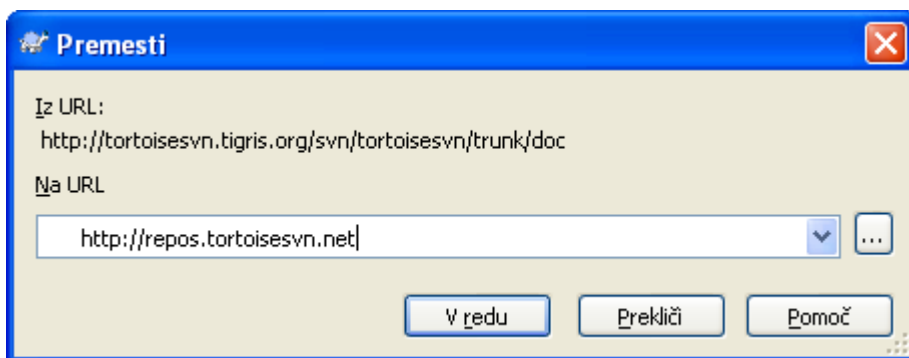
If you want to export a copy of your project tree structure but containing only the files which have changed in a particular revision, or between any two revisions, use the compare revisions feature described in [Razdelek 4.10.3, "Primerjanje map"](#).

4.26.1. Kako odstranim delovno kopijo iz nadzora različic

Včasih hočete svoje delovno kopijo pretvoriti v navadno mapo brez podmap `.svn`. Potrebujete ukaz `izvozi-na-mestu`, ki namesto ustvarjanja novega drevesa map odstrani kontrolne podmape.

Odgovor je presenetljivo preprost - izvozite mapo samo vase! TortoiseSVN ta poseben primer zazna in vas vpraša, če želite delovno kopijo pretvoriti v mapo, ki ni pod nadzorom različic. Če odgovorite *z da*, se kontrolne mape odstranijo, tako da dobite navadno drevesno strukturo, ki ni pod nadzorom.

4.27. Premeščanje delovne kopije



Slika 4.49. Okno za premeščanje

Včasih se lokacija (IP/URL) vašega skladišča spremeni. Mogoče ste celo v stiski in ne morete objavljati, nočete pa ponovno prevzeti delovne kopije iz nove lokacije in premikati spremenjenih podatkov v novo delovno kopijo. Iščete ukaz TortoiseSVN → Premesti. V bistvu ta naredi zelo malo: preišče vse zapise v mapah `.svn` in spremeni naslove URL v vseh zapisih na novo vrednost.

You may be surprised to find that TortoiseSVN contacts the repository as part of this operation. All it is doing is performing some simple checks to make sure that the new URL really does refer to the same repository as the existing working copy.



Pozor

Ta operacija se uporablja zelo redko. Ukaz Premesti se uporablja *le*, če se spremeni naslov URL korenke mape skladišča. Možni razlogi za to so:

- Spremenil se je IP naslov strežnika.
- Spremenil se je protokol (n.pr. `http://` v `https://`).
- Spremenila se je korenska pot skladišča.

Povedano drugače, ukaz Premesti uporabite takrat, ko vaša delovna kopija kaže na isto lokacijo v istem skladišču, se je pa samo skladišče preselilo.

Ne velja, če:

- se želite premakniti v drugo skladišče Subversion. V tem primeru je najbolje narediti svež prevzem iz novega skladišča.
- želite narediti preklon na drugo vejo ali mapo znotraj istega skladišča. Za to uporabite TortoiseSVN → Preklopi.... Za več informacij preberite **Razdelek 4.19.2, "Prevzeti ali prekloniti..."**.

Če uporabite ukaz Premesti v kateremkoli od zgoraj naštetih dveh primerih, *boste pokvarili svojo delovno kopijo* in prikazalo se vam bo veliko nerazložljivih napak pri objavljanju, posodabljanju... Če se vam to zgodi, je edina rešitev svež prevzem.

4.28. Integracija s sistemi za sledenje zadev

V razvoju programske opreme se pogosto dogaja, da so spremembe vezane na določenega hrošča ali številko zadeve. Uporabniki sistemov za sledenje zadev pogosto želijo asociirati spremembe v sistemu Subversion z določeno številko zadeve v sistemu za sledenje zadev. Večina sistemov za sledenje zadev omogoča pisanje ukaznih datotek akcije pred objavo, ki pregleda sporočilo dnevniškega zapisa in poišče številko zadeve. Tak način je podvržen napakam, saj se zanaša na dejstvo, da bodo uporabniki napisali sporočilo dnevniškega zapisa v pravilni obliki.

TortoiseSVN lahko pomaga uporabniku na dva načina:

1. Ko uporabnik vpiše sporočilo dnevniškega zapisa, lahko samodejno doda predhodno določeno vrstico, ki vsebuje številko zadeve, ki se nanaša na objavo. Tako se zmanjša možnost, da uporabnik vpiše številko zadeve v obliki, ki jo orodje za sledenje zadev ne zna pravilno razčleniti.

Lahko pa TortoiseSVN označi del vnešenega sporočila dnevniškega zapisa, ki ga sledilnik zadev prepozna. Tako uporabnik ve, da je sporočilo dnevniškega zapisa mogoče pravilno razčleniti.

2. Ko uporabnik brska po dnevniških zapisih, TortoiseSVN ustvari povezavo za vsako številko hrošča v sporočilu dnevniškega zapisa, ki požene spletni brskalnik in naloži stran ustrezne zadeve.

4.28.1. Dodajanje številke zadev dnevniškim zapisom

V TortoiseSVN lahko integrirate sledilnik zadev po vaši izbiri. Za to morate definirati nekaj lastnosti, ki se začnejo z `bugtraq:`. Nastavljene morajo biti na mapah: (**Razdelek 4.17, "Nastavitve projekta"**)

V TortoiseSVN lahko integrirate sisteme za sledenje na dva načina. Prvi je na osnovi enostavnih nizov, drugi pa na osnovi *regularnih izrazov*. Lastnosti, ki se uporabljajo pri obeh pristopih, so:

`bugtraq:url`

Nastavite vrednost te lastnosti na naslov URL orodja za sledenje zadev. Naslov mora biti pravilne oblike (URI) in mora vsebovati niz `%BUGID%`. Niz `%BUGID%` se zamenja s številko zadeve, ki ste jo vnesli. To omogoča sistemu TortoiseSVN, da prikaže povezavo v sporočilu dnevniškega zapisa, tako da pri pregledovanju dnevniških zapisov lahko skočite neposredno na stran orodja za sledenje zadev. Nastavitev te lastnosti ni nujna; če je ne nastavite, boste v sporočilu dnevniškega zapisa videli le številko zadeve namesto povezave. Primer: projekt TortoiseSVN uporablja vrednost `http://issues.tortoisesvn.net/?do=details&id=%BUGID%`

Namesto absolutnih lahko uporabljate tudi relativne naslove URL. To je uporabno v primeru, ko je sledilnik zadev na isti domeni/strežniku kot skladišče Subversion. Če se ime domeni kdaj spremeni, vam tako ni potrebno spreminjati lastnosti `bugtraq:url`. Obstajata dva načina, kako določiti relativen naslov URL:

Če se začne z nizom `^/`, to pomeni pot, relativno glede na korensko mapo skladišča. Primer: `^/../?do=details&id=%BUGID%` se razširi v `http://tortoisesvn.net/?`

`do=details&id=%BUGID%` če se skladišče nahaja na `http://tortoisesvn.net/svn/trunk/`.

Naslov URL, ki se začne z nizom `/`, je relativen na ime strežnika. Primer: `/?do=details&id=%BUGID%` se razširi v `http://tortoisesvn.net/?do=details&id=%BUGID%`, če se skladišče nahaja kjerkoli na naslovu `http://tortoisesvn.net`.

`bugtraq:warnifnoissue`

Nastavite to vrednost na `true`, če želite, da vas TortoiseSVN opozori, če pustite polje za vnos številke zadeve prazno. Dovoljene vrednosti so `true/false`. Če vrednost ni nastavljena, se predpostavlja `false` (brez opozarjanja).

4.28.1.1. Številka zadeve v vnosnem polju

Pri enostavnem načinu TortoiseSVN uporabniku pokaže posebno polje za vnos številke hrošča. Nato se na začetek/konec dnevniškega zapisa doda posebna vrstica.

`bugtraq:message`

Ta lastnost aktivira način z *vnosnim poljem*. Če nastavite to lastnost, vas TortoiseSVN ob objavi povpraša po številki zadeve. Številka se vnese na konec sporočila dnevniškega zapisa. Vsebovati mora niz `%BUGID%`, ki se ob objavi zamenja s številko zadeve. To zagotavlja, da sporočilo vedno vsebuje referenco na številko zadeve, ki je podana v pravilni obliki in jo orodje za sledenje zadev lahko razčleni ter poveže s to objavo. Primer: projekt TortoiseSVN uporablja vrednost `Issue : %BUGID%`, vendar je to odvisno od orodja, ki ga uporabljate.

`bugtraq:append`

Ta lastnost določa, ali se številka hrošča (zadeve) pripne na konec sporočila dnevniškega zapisa (`true`) ali na začetek (`false`). Veljavni vrednosti sta `true/false`. Če vrednost ni nastavljena, se predpostavlja `true`, zato da obstoječi projekti še vedno delujejo.

`bugtraq:label`

To besedilo se prikaže v oknu za objave in opisuje vnosno polje za vpis številke zadeve. Če ni nastavljeno, se pojavi napis `Bug-ID / Issue-Nr.`. Upoštevajte, da se okno ne bo samodejno povečalo, zato omejite število znakov v napisu na 20 do 25.

`bugtraq:number`

Če je vrednost nastavljena na `true`, je v polje za številko zadeve dovoljeno vpisati le številke. Izjema je le vejica, ki jo uporabite, da vnesete več številk zadeve. Veljavni vrednosti sta `true/false`. Če vrednost ni nastavljena, se predpostavlja `true`.

4.28.1.2. Številka zadeve z uporabo regularnih izrazov

Pri pristopu z *regularnimi izrazi* TortoiseSVN ne prikaže vnosnega polja, ampak označi del sporočila dnevniškega zapisa, ki ga orodje za sledenje zadev prepozna. To se dogaja med vnosom besedila. To pomeni tudi, da se številka zadeve lahko nahaja kjerkoli v sporočilu. Ta metoda je precej bolj fleksibilna in jo uporablja tudi sam projekt TortoiseSVN.

`bugtraq:logregex`

Ta lastnost aktivira sistem sledenja zadev v načinu *Regex*. Vsebuje enega ali dva regularna izraza, ločena z novo vrstico.

Če sta nastavljena dva izraza, se prvi uporablja za iskanje izrazov, ki vsebujejo ID hrošča. Drugi izraz izlušči ID hrošča iz rezultata prvega izraza. To omogoča uporabo več IDjev hroščev v povezavi z govornimi jezikom, n. pr. če odpravite več hroščev naenkrat v vključite naslednji niz: “S to spremembo so odpravljene napake #23, #24 in #25”

Če želite ujeti ID hrošča v sporočilu dnevniškega zapisa, lahko uporabite naslednja regularna izraza, ki jih uporablja projekt TortoiseSVN: `[Nn]apak[aie]:?(\s*(,|in)?\s*\#(d+)+and(\d+)`

Prvi izraz iz sporočila dnevniškega zapisa izlušči niz "napake #23, #24 in #25". Drugi regularni izraz iz rezultata prvega izraza izlušči številke zadev. V tem primeru vrne "23", "24" in "25", ki se uporabijo kot številke zadev.

Če malce razčlenimo prvi regularni izraz: začetni se mora z besedo "napaka". Beseda se lahko začne z veliko začetnico, lahko pa je zapisana v obliki dvojine ali množine ("[aie]"). Besedi lahko sledi dvopičje. Sledi ena ali več skupin, vsaka lahko vsebuje vodilne presledke, vejico ali niz "in" in zaključne presledke. Na koncu se nahaja obvezen znak "#" in obvezna številka.

Če je nastavljen le en izraz, potem mora le številka zadeve ustrezati skupini v regularnem izrazu. Primer: `[Nn]apaka(?:)?#?(\\d+)` To metodo zahtevajo nekateri sledilniki zadev, n. pr. trac, in jih je težje zgraditi. Zato priporočamo, da to metodo uporabite le, če tako velevajo navodila izbranega sledilnika zadev.

Če regularnih izrazov ne poznate, lahko nekaj o njih preberete na naslovu http://en.wikipedia.org/wiki/Regular_expression, dokumentacijo in priročnik pa poiščite na <http://www.regular-expressions.info/>.

Če sta nastavljeni obe lastnosti - `bugtraq:message` in `bugtraq:logregex` - se upošteva `logregex`.



Namig

Tudi če nimate orodja za sledenje zadev, ki ga nastavite v skripto akcij za akcijo pred objavo, lahko še vedno uporabljate ta sistem, da številke zadev v sporočilu spremenite v povezave!

Povezav mogoče niti ne potrebujete, številke zadev pa se vseeno pojavijo v posebnem stolpcu v dnevniku, tako da lahko hitreje najdete spremembe, ki so povezane z določeno zadevo.

Nekatere lastnosti `tsvn`: zahtevajo vrednost `true/false`. TortoiseSVN razume tudi besedo `yes` kot sinonim za `true` in besedo `no` kot sinonim za `false`.



Nastavite lastnosti na mapah

Da bo sistem deloval, morate te lastnosti nastaviti na mapah. Ko objavite datoteko ali mapo, se lastnosti preberejo iz te mape. Če TortoiseSVN lastnosti ne najde, išče naprej v nadrejeni mapi, dokler ne pride do mape, ki ni pod nadzorom različic ali do korenenske mape pogona (n. pr. `C:\`). Če ste prepričani, da bodo uporabniki prevzemali samo iz mape n.pr. `trunk/` in ne iz katere od podmap, potem lahko lastnosti nastavite na mapi `trunk/`. Če niste prepričani, potem morate lastnosti nastaviti rekurzivno na vseh podmapah. Lastnost, nastavljena globlje v projektni hierarhiji, vedno povozi nastavitve na višjih nivojih (bližje mapi `trunk/`).

Zgolj za lastnosti `tsvn`: lahko uporabite potrditveno polje **Rekurzivno**. S tem nastavite lastnost na vse podmape v hierarhiji, na datoteke pa ne.



No Issue Tracker Information from Repository Browser

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.

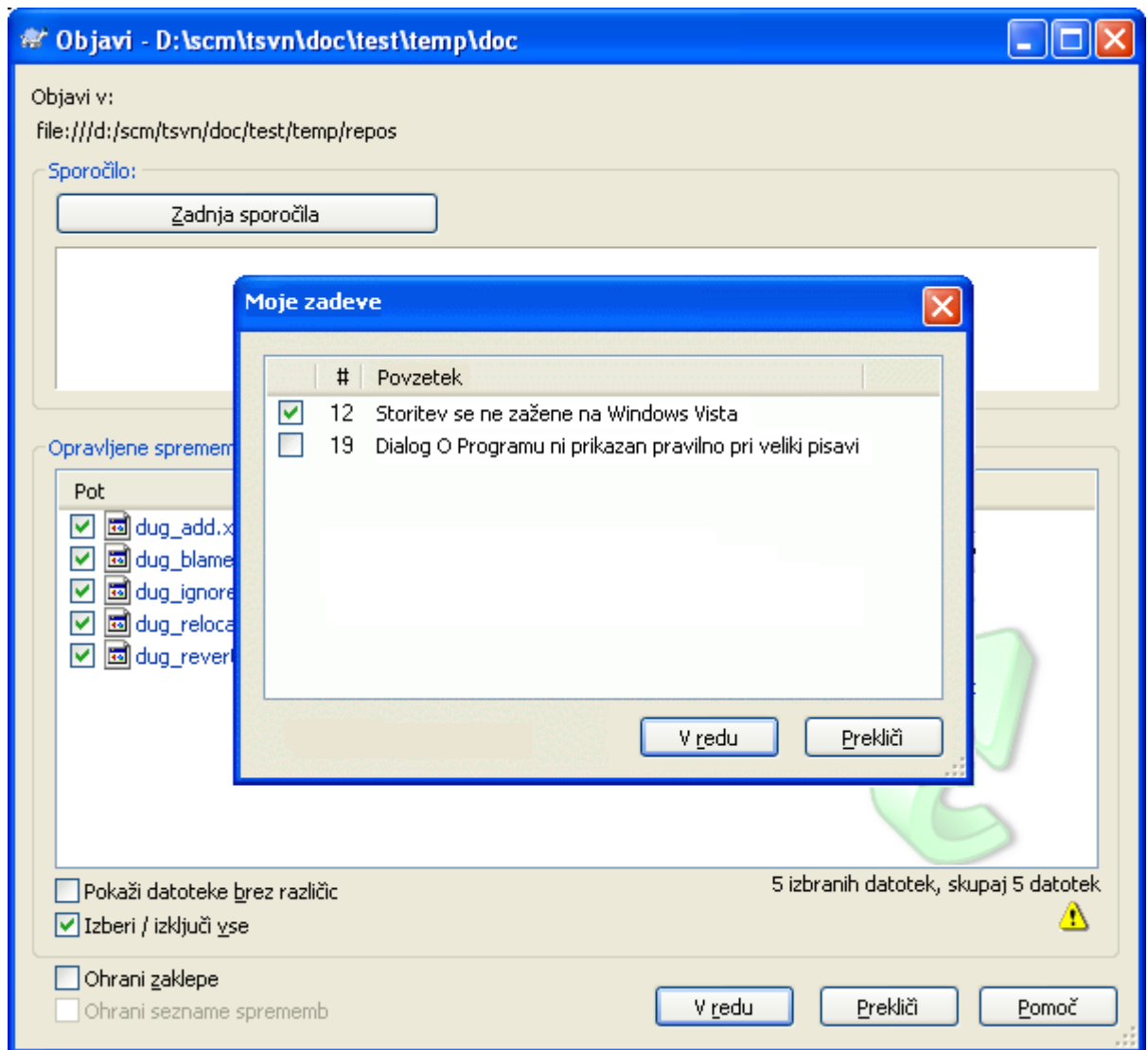
This issue tracker integration is not restricted to TortoiseSVN; it can be used with any Subversion client. For more information, read the full *Issue Tracker Integration Specification* [<http://tortoisesvn.googlecode.com/svn/trunk/doc/issuetrackers.txt>] in the TortoiseSVN source repository. (Razdelek 3, "TortoiseSVN je brezplačen!") explains how to access the repository).

4.28.2. Pridobivanje informacij iz sledilnika zadev

The previous section deals with adding issue information to the log messages. But what if you need to get information from the issue tracker? The commit dialog has a COM interface which allows integration an external program that can talk to your tracker. Typically you might want to query the tracker to get a list of open issues assigned to you, so that you can pick the issues that are being addressed in this commit.

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* [<http://tortoisesvn.googlecode.com/svn/trunk/contrib/issue-tracker-plugins>]. (Razdelek 3, "TortoiseSVN je brezplačen!" explains how to access the repository). A summary of the API is also given in *Poglavje 6, 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.

Za ilustracijo si predstavljajmo, da vam je skrbnik sistema priskrbel vtičnik za sledilnik zadev, ki ste ga namestili, sedaj pa ga morate uporabiti pri delovnih kopijah. Vse potrebno postorite v oknu Nastavitve programa TortoiseSVN. Ko odprete okno za objave v delovni kopiji, ki ima nastavljen vtičnik, se na vrhu pogovornega okna pojavi nov gumb.



Slika 4.50. Primer poizvedovalnika sledilnika zadev

V tem primeru lahko izberete eno ali več odprtih zadev. Vtičnik nato ustvari posebej oblikovano besedilo, ki ga doda v sporočilo dnevniškega zapisa.

4.29. Integracija z internetno naravnanimi pregledovalniki skladišč

Obstaja kar nekaj spletnih pregledovalnikov skladišč za Subversion, n. pr. *ViewVC* [<http://www.viewvc.org/>] in *WebSVN* [<http://websvn.tigris.org/>]. TortoiseSVN nudi možnost povezave s temi pregledovalniki.

V TortoiseSVN lahko integrirate pregledovalnik skladišč po izbiri. Za to morate nastaviti nekaj lastnosti, ki definirajo povezavo. Nastavljene morajo biti na mapah: ([Razdelek 4.17](#), "Nastavitve projekta")

webviewer:revision

To lastnost nastavite na naslov URL spletnega pregledovalnika, da vidite vse spremembe v določeni reviziji. Vrednost mora biti pravilno oblikovan naslov (URI) in mora vsebovati izraz `%REVISION%`. Ta se zamenja z zahtevano številko revizije. To sistemu TortoiseSVN omogoča prikaz ukaza v kontekstnem meniju v Dnevniku - Kontekstni meni → Poglej revizijo v spletnem brskalniku

webviewer:pathrevision

Vrednost te lastnosti nastavite na naslov URL spletnega pregledovalnika, da vidite spremembe določene datoteke v določeni reviziji. Vrednost mora biti pravilno oblikovan naslov (URI) in mora vsebovati izraza `%REVISION%` in `%PATH%`. Izraz `%PATH%` se zamenja z relativno potjo do korenške mape skladišča. To sistemu TortoiseSVN omogoča prikaz ukaza v kontekstnem meniju Dnevnik - Kontekstni meni → Poglej revizijo in pot v spletnem brskalniku Primer: če v spodnjem delu dnevnika kliknete na datoteko `/trunk/src/file`, potem se izraz `%PATH%` v naslovu URL zamenja z vrednostjo `/trunk/src/file`.

Namesto absolutnih naslovov URL lahko uporabljate tudi relativne. To je uporabno, če se vaš spletni brskalnik po skladišču nahaja na isti domeni/strežniku kot skladišče. Če se ime domene kadarkoli spremeni, vam lastnosti `webviewer:revision` in `webviewer:pathrevision` ni potrebno spreminjati. Oblika je enaka kot za lastnost `bugtraq:url`. Poglejte [Razdelek 4.28](#), "Integracija s sistemi za sledenje zadev".



Nastavite lastnosti na mapah

Da bo sistem deloval, morate te lastnosti nastaviti na mapah. Ko objavite datoteko ali mapo, se lastnosti preberejo iz te mape. Če TortoiseSVN lastnosti ne najde, išče naprej v nadrejeni mapi, dokler ne pride do mape, ki ni pod nadzorom različic ali do korenške mape pogona (n. pr. `C:\`). Če ste prepričani, da bodo uporabniki prevzemali samo iz mape n.pr. `trunk/` in ne iz katere od podmap, potem lahko lastnosti nastavite na mapi `trunk/`. Če niste prepričani, potem morate lastnosti nastaviti rekurzivno na vseh podmapah. Lastnost, nastavljena globlje v projektni hierarhiji, vedno povzici nastavitve na višjih nivojih (bližje mapi `trunk/`).

Zgolj za lastnosti `tsvn`: lahko uporabite potrditveno polje **Rekurzivno**. S tem nastavite lastnost na vse podmape v hierarhiji, na datoteke pa ne.



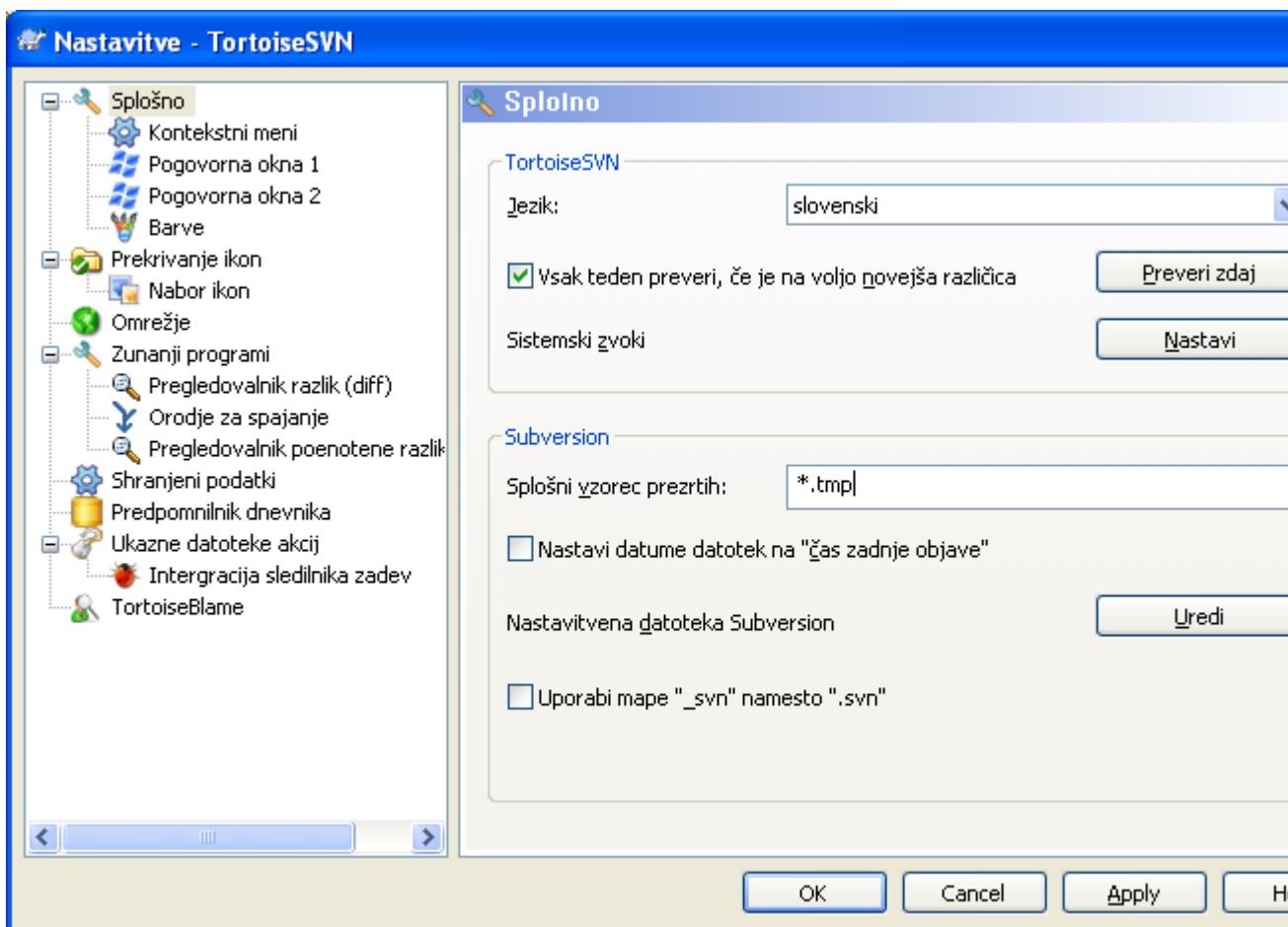
No Repo Viewer Links from Repository Browser

Because the repo viewer integration depends upon accessing subversion properties, you will only see the results when using a checked out working copy. Fetching properties remotely is a slow operation, so you will not see this feature in action from the repo browser.

4.30. Nastavitve TortoiseSVN

Da izveste, čemu služi določena nastavev, počakajte z miško nekaj trenutkov nad vnosnim/potrditvenim poljem... in prikazal se vam bo uporaben namig.

4.30.1. Splošne nastavitve



Slika 4.51. Okno za nastavitve, Splošno

To okno vam omogoča nastavev jezika in nastavev, specifičnih za Subversion.

Jezik

Izbere jezik za uporabniški vmesnik. Kaj drugega ste pa pričakovali?

Vsak teden preveri, če je na voljo novejša različica

Če je to polje potrjeno, bo TortoiseSVN tedensko na strežniku preveril, če je morda na voljo novejša različica programa. Uporabite gumb **Preveri zdaj**, če želite to storiti takoj.

Sistemske zvoki

TortoiseSVN ima tri lastne zvočne efekte, ki so vedno nameščeni.

- Napaka
- Obvestilo
- Opozorilo

Preko Nadzorne plošče sistema Windows lahko izberete zvoke (ali popolnoma izklopite privzete). Gumb **Nastavi** je bližnjica do Nadzorne plošče.

Splošni vzorec prezrtih

Global ignore patterns are used to prevent unversioned files from showing up e.g. in the commit dialog. Files matching the patterns are also ignored by an import. Ignore files or directories by typing in the names or extensions. Patterns are separated by spaces e.g. `bin obj *.bak *.~?? *.jar *. [Tt]mp`. These patterns should not include any path separators. Note also that there is no way to differentiate between files and directories. Read [Razdelek 4.13.1, "Iskanje vzorcev v seznamu prezrtih elementov"](#) for more information on the pattern-matching syntax.

Upoštevajte, da bodo vzorci prezrtih elementov, ki jih nastavite tukaj, vplivali tudi na druge odjemalce za Subversion, ki so nameščeni na vašem računalniku, vključno z odjemalcem za ukazno vrstico.



Opozorilo

Če uporabljate konfiguracijsko datoteko Subversion za nastavljanje splošnega vzorca prezrtih, bodo nastavitve, ki so nastavite tu, povožene. Do konfiguracijske datoteke Subversion dostopate z gumbom **Uredi**, kot je opisano spodaj.

Vzorec prezrtih elementov vpliva na vse vaše projekte. Ni pa pod nadzorom različic, tako da ne vpliva na druge uporabnike. Lahko pa uporabite tudi lastnost `svn:ignore`, ki je pod nadzorom. Z njo izključite datoteke ali mape iz nadzora različic. Za več informacij preberite [Razdelek 4.13, "Dodajanje datotek in map na seznam prezrtih elementov"](#).

Nastavi datume datotek na "čas zadnje objave"

Ta možnost pove sistemu TortoiseSVN, naj pri prevzemu ali posodobitvi popravi časovni žig datotek na čas objave. V nasprotnem primeru TortoiseSVN uporabi trenutni datum. Če razvijate programsko opremo, je najbolje, da uporabljate trenutni datum/čas, saj sistemi za gradjo običajno pogledajo datum in čas spremembe datoteke, da ugotovijo, katere je potrebno ponovno prevesti. Če uporabite "zadnji čas objave" in naredite povrnitev, se projekt morda ne bo prevedel tako, kot želite.

Nastavitvena datoteka Subversion

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/1.5/svn.advanced.confarea.html) [http://svnbook.red-bean.com/en/1.5/svn.advanced.confarea.html]. The section on [Automatic Property Setting](http://svnbook.red-bean.com/en/1.5/svn.advanced.props.html#svn.advanced.props.auto) [http://svnbook.red-bean.com/en/1.5/svn.advanced.props.html#svn.advanced.props.auto] 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/1.5/svn.advanced.confarea.html#svn.advanced.confarea.windows-registry) [http://svnbook.red-bean.com/en/1.5/svn.advanced.confarea.html#svn.advanced.confarea.windows-registry] to find out more.

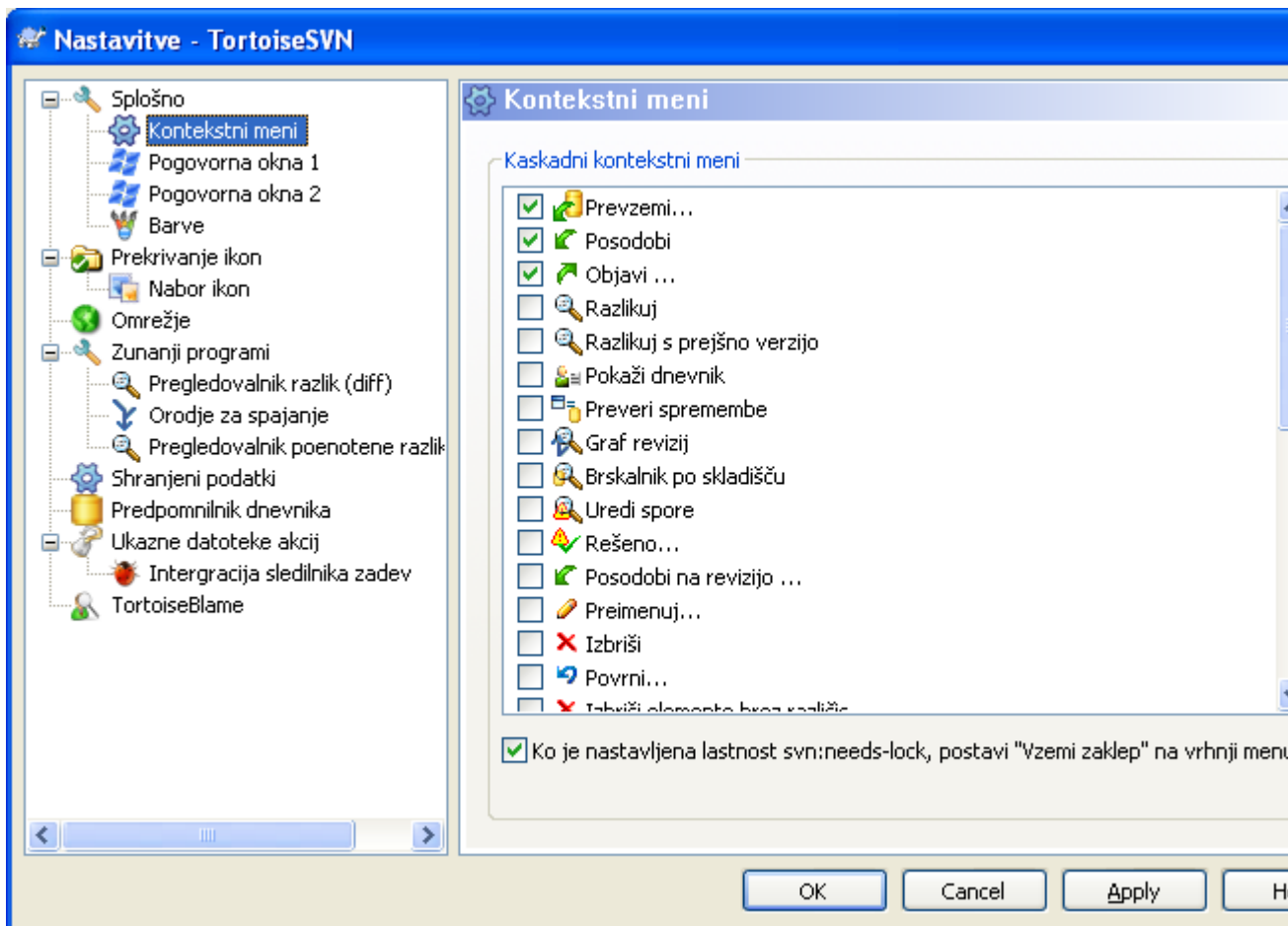
Uporabi mape `_svn` namesto `.svn`

VS.NET pri razvoju spletnih projektov ne zna delati z mapami `.svn`, ki jih Subversion uporablja za shranjevanje notranjih informacij. To ni napaka v sistemu Subversion, ampak v razvojnem okolju VS.NET in razširitvah frontpage, ki jih uporablja. Za več informacij o tem problemu preberite [Razdelek 4.30.11, "Delovne mape Subversion"](#).

Če želite spremeniti obnašanje programov Subversion in TortoiseSVN, lahko uporabite to potrditveno polje, da nastavite ustrezno okoljsko spremenljivko.

Vedeti morate, da sprememba te nastavitve ne vpliva na obstoječe delovne kopije. Te boste morali popraviti sami z uporabo ukazne datoteke (preberite Pogosto zastavljena vprašanja) ali pa enostavno narediti novo delovno kopijo.

4.30.1.1. Nastavitve kontekstnega menija



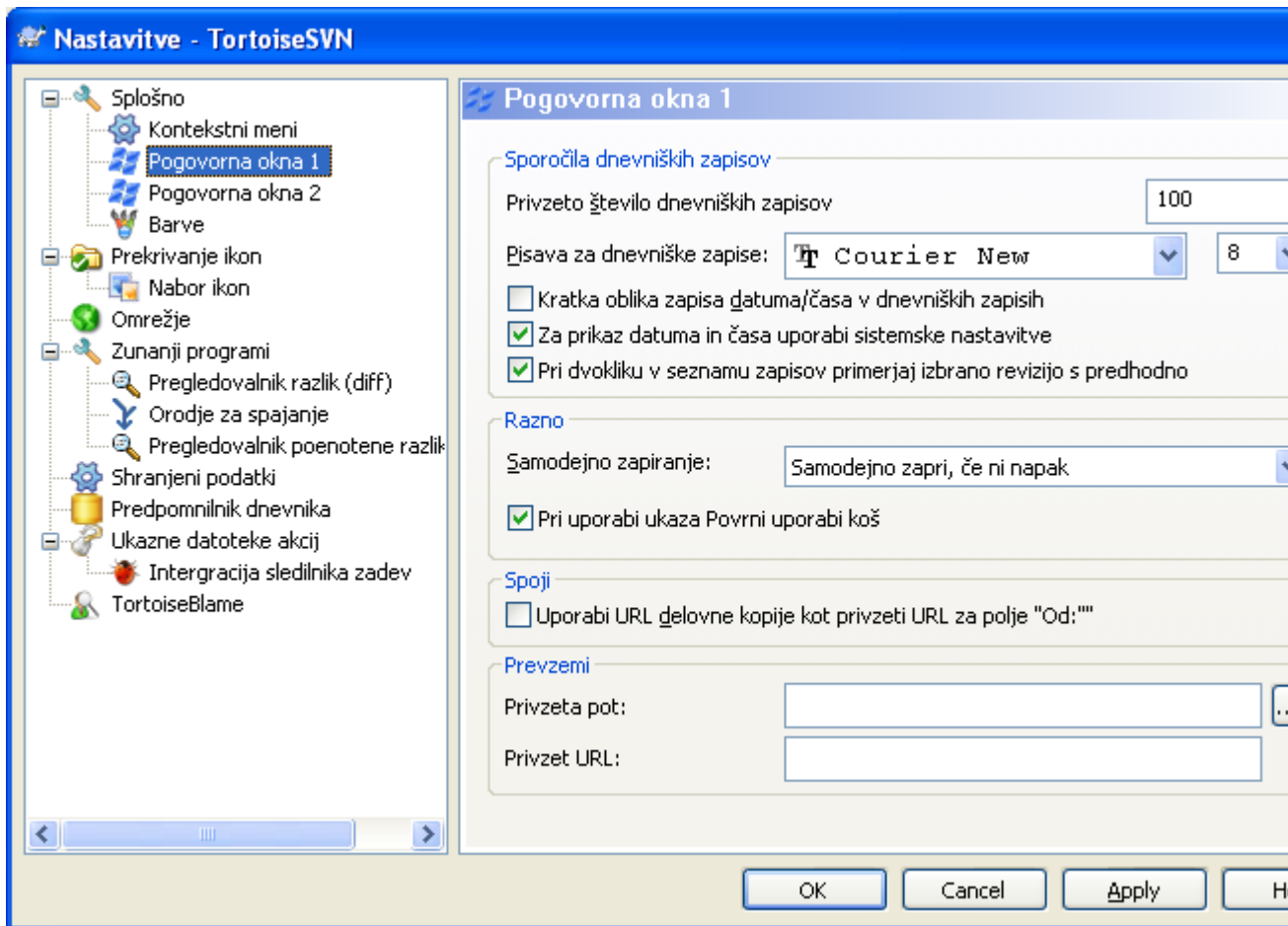
Slika 4.52. Okno nastavitvev, Kontekstni meni

Ta stran omogoča nastavitve vnosov kontekstnega menija TortoiseSVN, ki se pojavijo v glavnem kontekstnem meniju. Po privzetih nastavitvah večina vnosov ni izbranih in se pojavijo samo v podmeniju.

Ukaz **Dobi zaklep** je poseben primer. Seveda ga lahko postavite v glavni kontekstni meni, kot je opisano zgoraj. Ker pa večina datotek ne potrebuje zaklepanja, je to le nepotrebna obremenitev menija. Po drugi strani pa pri datotekah, ki imajo nastavljeno lastnost `svn:needs-lock`, potrebujete ta ukaz pri vsakem urejanju, zato je v tem primeru koristno, če se nahaja v glavnem kontekstnem meniju. Če potrdite potrditveno polje, se bo ukaz **Dobi zaklep** pojavil v glavnem meniju vsakokrat, ko izberete datoteko z nastavljeno lastnostjo `svn:needs-lock`.

If there are some paths on your computer where you just don't want TortoiseSVN's context menu to appear at all, you can list them in the box at the bottom.

4.30.1.2. Pogovorna okna 1



Slika 4.53. Okno nastavitvev, Pogovorna okna 1

To okno omogoča nastavitve nekaterih pogovornih oken programa TortoiseSVN po vaših željah.

Privzeto število dnevniških zapisov

Omeji število dnevniških zapisov, ki jih TortoiseSVN prenese, ko prvič izberete TortoiseSVN → Pokaži dnevnik. Uporabno za počasne povezave. Vedno lahko uporabite gumb Pokaži vse ali gumb Naslednjih 100, če želite videti več zapisov.

Pisava za dnevniški zapis

Izbere pisavo in velikost, ki se uporablja za izpis sporočila dnevniškega zapisa v srednjem delu dnevnika in pri sestavljanju dnevniških zapisov v oknu za objavo.

Kratka oblika datuma / ure v dnevniških zapisih

Če standardni dnevniški zapisi zavzamejo preveč prostora na zaslonu, uporabite kratek format.

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.

Okno napredka

TortoiseSVN lahko v primeru uspešne operacije samodejno zapre pogovorna okna. Ta nastavek vam omogoča nastaviti pogoje samodejnega zapiranja. Privzeta (priporočena) nastavek je Zapri

ročno, kar vam omogoča, da pregledate sporočila in preverite, kaj se je zgodilo. Lahko pa se odločite, da vas določene vrste sporočil ne zanimajo in želite, da se takšna okna samodejno zaprejo, če se ne pojavijo kritične napake.

Izbira **Samodejno zapri**, če ni **spajanj**, **dodajanj** ali **brisanj** pomeni, da se bo okno napredka zaprlo v primeru enostavnih posodobitev. Če pa pride do spajanja sprememb iz skladišča v delovno kopijo ali če so bile kakšne datoteke dodane ali izbrisane, bo pogovorno okno ostalo odprto. Odprto bo ostalo tudi, če se pri posodobitvi pojavijo spori ali napake.

Izbira **Samodejno zapri**, če pri **krajevnih operacijah** ni **spajanj**, **dodajanj** ali **brisanj** pomeni, da se bo okno napredka zaprlo kot pri uporabi možnosti **Samodejno zapri**, če ni **spajanj**, **dodajanj** ali **brisanj**, vendar le v primeru krajevnih operacij, kot sta dodajanje datotek ali povrnitev sprememb. Pri oddaljenih operacij ostane okno odprto.

Izbira **Samodejno zapri**, če ni **sporov** še bolj omili pogoje in zapre pogovorna okna tudi v primeru spajanja, dodajanja ali brisanja. Če pa se ob tem pojavijo spori ali napake, pa bo okno vseeno ostalo odprto.

Izbira **Samodejno zapri**, če ni **napak** pomeni, da se pogovorno okno vedno zapre, tudi v primeru sporov. Okno ostane odprto le v primeru, če Subversion ne more dokončati operacije. Primer: posodobitev je neuspešna, če strežnik ni dosegljiv, objava ni uspešna, če je delovna kopija zastarela.

Use recycle bin when reverting

When you revert local modifications, your changes are discarded. TortoiseSVN gives you an extra safety net by sending the modified file to the recycle bin before bringing back the pristine copy. If you prefer to skip the recycle bin, uncheck this option.

Uporabi URL delovne kopije kot privzeti URL za polje "Iz:"

Po privzetih nastavitvah si okno **Spoji** zapomni naslov URL, ki je nastavljen v polju **Iz:**. Nekateri uporabniki opravljajo spajanja iz precej različnih točk v svoji strukturi map in jim je lažje, če se v polju pojavi naslov URL trenutne delovne kopije. Naslov URL potem popravijo, da ustreza vzporedni poti na drugi veji.

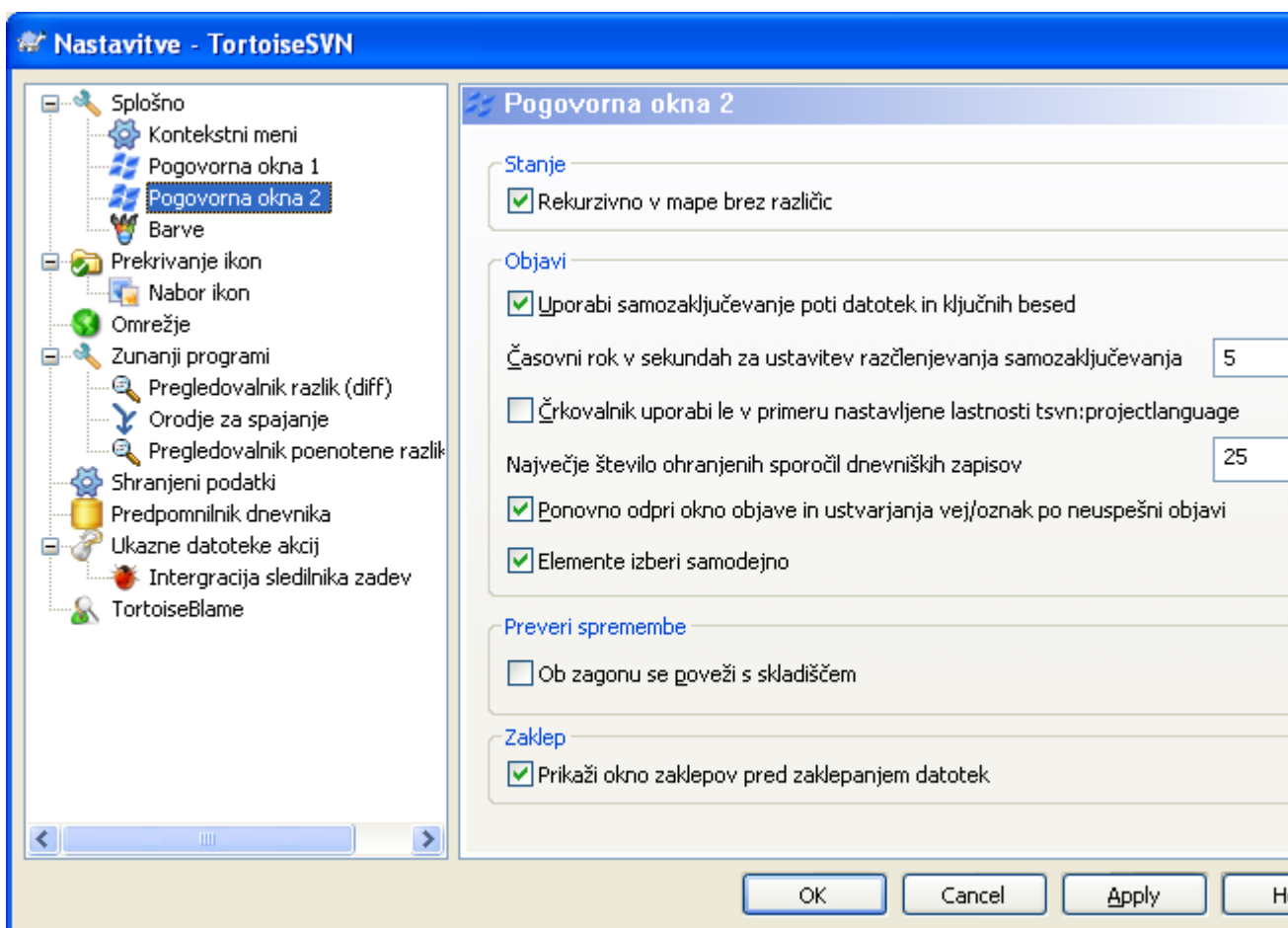
Privzeta pot za prevzem

Nastavite lahko privzeto pot za prevzeme. Če imate vse prevzeme na enem mestu, potem je uporabno, da sta pogon in mapa predizpolnjena, tako da dodate na koncu samo ime mape.

Privzeti URL za prevzem

Lahko pa nastavite privzeti naslov URL za prevzeme. Če pogosto prevzimate podprojekte nekega velike projekta, je uporabno, če imate polje naslov URL predizpolnjeno in mu na koncu dodate samo ime podprojekta.

4.30.1.3. Pogovorna okna 2



Slika 4.54. Okno nastavitvev, Pogovorna okna 2

Vstopi rekurzivno v mape brez različic.

Če je to polje potrjeno (privzeta nastavitve), potem se v oknih Dodaj, Objavi ali Preveri spremembe v primeru map brez različic pojavijo tudi podrejene mape in datoteke brez različic. Če to polje izklopite, potem je prikazana samo vrhnja mapa. Izklop tega polja pripomore k manjši obremenjenosti pogovornih oken. Če v takšnem primeru izberete za dodajanje mapo brez različic, se rekurzivno doda vsa njena vsebina.

Uporabi samodokončanje poti datotek in ključnih besed

Okno za objavo omogoča razčlenjevanje seznama datotek za objavo. Ko vtipkate prve tri črke elementa na seznamu, se pojavi okno za samodokončanje. Pritisnete lahko tipko ENTER in s tem vnesete predlagani niz. Za vklop te zmožnosti potrdite to polje.

Časovni rok v sekundah za ustavitev razčlenjevanja samodokončanja

Orodje za razčlenjevanje je lahko precej počasno, če dela z velikim številom datotek. Časovni rok prepreči, da bi okno za objavo čakalo predolgo časa. Če pri samodokončanju pogrešate določene informacije, podaljšajte časovni rok.

Uporabi črkvalnik le ko je nastavljena lastnost `tsvn:projectlanguage`

Če ne želite uporabiti črkvalnika za vse objave, potrdite to polje. Črkvalnik bo še vedno omogočen tam, kjer ga zahtevajo projektne lastnosti.

Največje število ohranjenih dnevniških zapisov

When you type in a log message in the commit dialog, TortoiseSVN stores it for possible re-use later. By default it will keep the last 25 log messages for each repository, but you can customize that

number here. If you have many different repositories, you may wish to reduce this to avoid filling your registry.

Note that this setting applies only to messages that you type in on this computer. It has nothing to do with the log cache.

Ponovno odpri okno za objave/oznake/veje, če je bila objava neuspešna

Ko objava zaradi določenega razloga ni uspešna (delovno kopijo je potrebno posodobiti, skripta akcije pred objavo prepreči objavo, napake v omrežju), lahko izberete to možnost, da okno objave ostane odprto in pripravljeno na ponoven poskus. Morate pa se zavedati, da to lahko povzroči težave. Če je vzrok za napako neposodobljena delovna kopija, jo morate posodobiti, to pa lahko povzroči spore. Le-te morate najprej rešiti.

Elemente izberi samodejno

Običajno obnašanje okna za objave je, da so vse spremenjene datoteke pod nadzorom izbrane samodejno. Če vam bolj ustreza, da ni izbrana nobena datoteka in boste elemente za objavo izbrali sami, izključite to možnost.

Poveži se s skladiščem ob zagonu

Okno Preveri spremembe po privzetih nastavitvah preveri le delovno kopijo, skladišče pa preverja le, če kliknete gumb **Preveri skladišče**. Če želite, da se skladišče vedno samodejno preverja, potrdite to možnost.

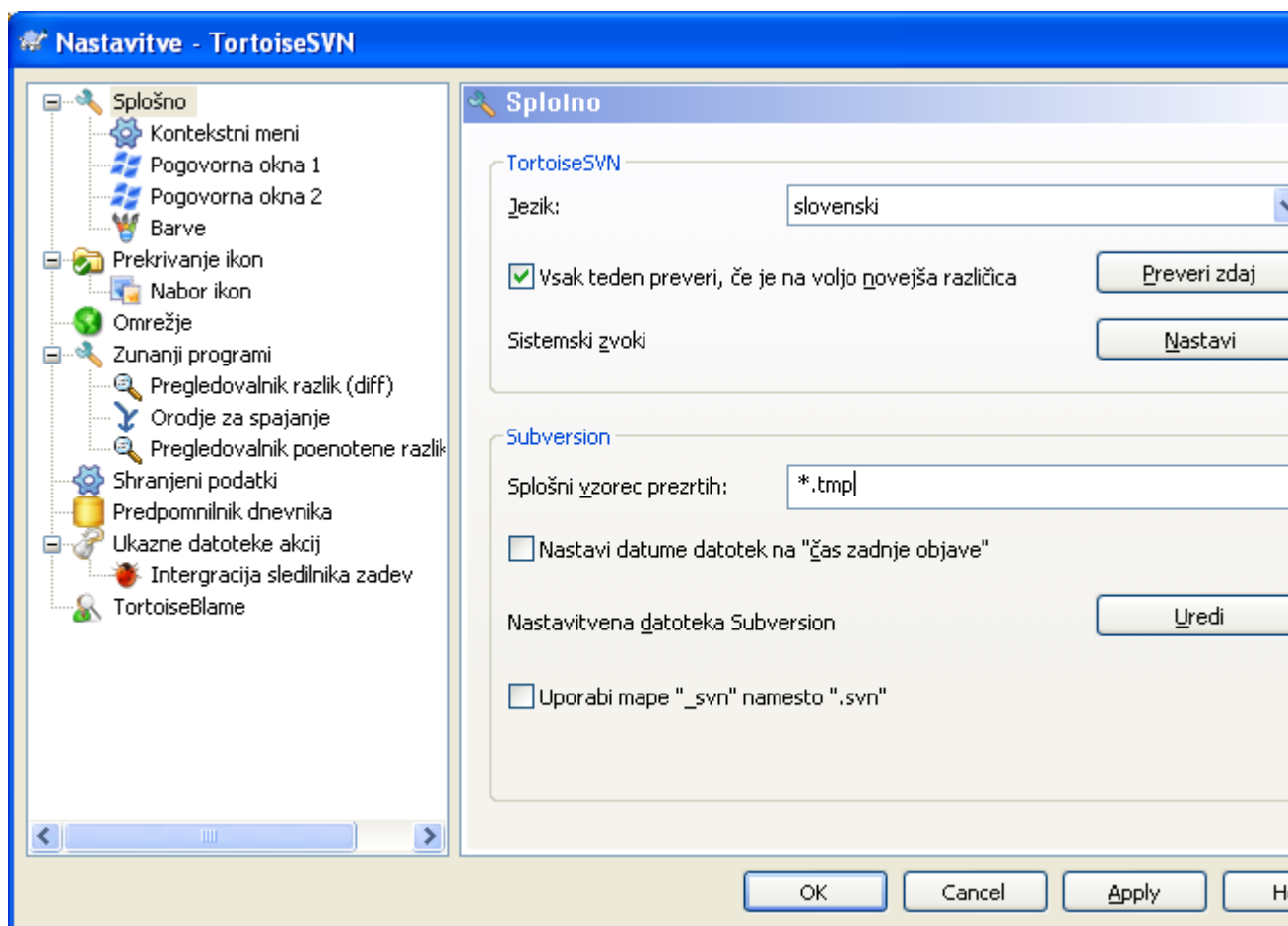
Prikaži okno zaklepov pred zaklepanjem datotek

Pri nekaterih projektih velja pravilo, da se pri zaklepanju ene ali več datotek z ukazom `TortoiseSVN` → `Lock` vpiše sporočilo zaklepa z vzrokom zaklepa. Če sporočil zaklepov ne uporabljate, lahko to možnost izključite. S tem preskočite pogovorno kono za vpis sporočila in nemudoma zaklenete datoteke.

Če uporabite zaklepanje na mapi, se vedno pojavi pogovorno okno za zaklepanje, ki vam ponudi možnost izbire datotek za zaklepanje.

Če vaš projekt uporablja lastnost `tsvn:lockmsgminsize`, se bo pogovorno okno zaklepov prikazalo ne glede na te nastavitve, ker projekt *zahteva* zapis zaklepa.

4.30.1.4. Nastavitev barv TortoiseSVN



Slika 4.55. Okno nastavitve, Barve

To pogovorno okno omogoča nastavljanje barve besedil v pogovornih oknih aplikacije TortoiseSVN.

Možen ali resničen spor / ovirano

Do spora je prišlo med posodobitvijo ali med spajanjem. Posodobitev ovira datoteka/mapa brez različic, ki nosi isto ime kot datoteka/mapa pod nadzorom.

Ta barva se uporablja tudi za sporočila o napakah v oknu napredka.

Dodane datoteke

Elementi, dodani v skladišče.

Manjkajoče / izbrisano / zamenjano

Elementi, izbrisani iz skladišča, ki v delovni kopiji manjkajo, ali element, izbrisan iz delovne kopije in zamenjan z drugo datoteko z istim imenom.

Spojeno

Spremembe iz skladišča so bile uspešno (brez sporov) spojene v delovno kopijo.

Spremenjeno / kopirano

Elementi, dodani z zgodovino, ali poti, kopirane v skladišču. Uporablja se tudi v Dnevniku za kopirane elemente.

Izbrisano vozlišče

Element, ki je bil izbrisan iz skladišča.

Dodano vozlišče

Element, ki je bil dodan v skladišče z uporabo ukazov dodaj, kopiraj ali premakni.

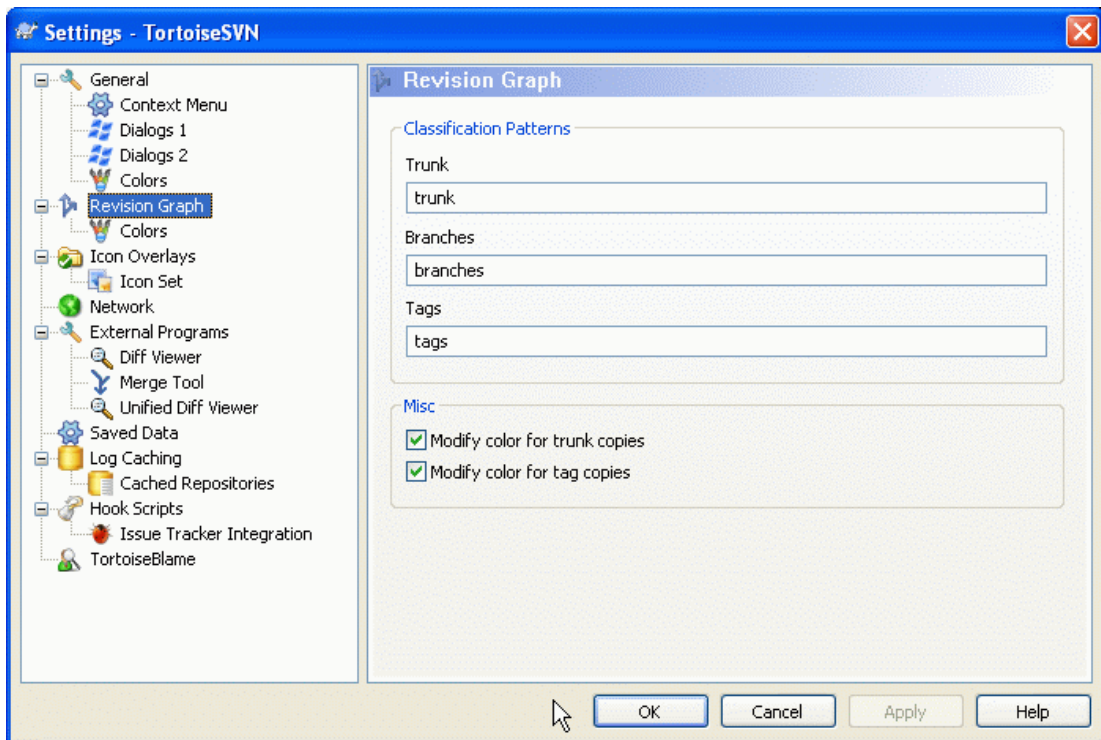
Preimenovano vozlišče

Element, ki je bil preimenovan znotraj skladišča.

Zamenjano vozlišče

Izvirni element je bil izbrisan. Nadomesti ga nov element z enakim imenom.

4.30.2. Revision Graph Settings



Slika 4.56. The Settings Dialog, Revision Graph Page

Classification Patterns

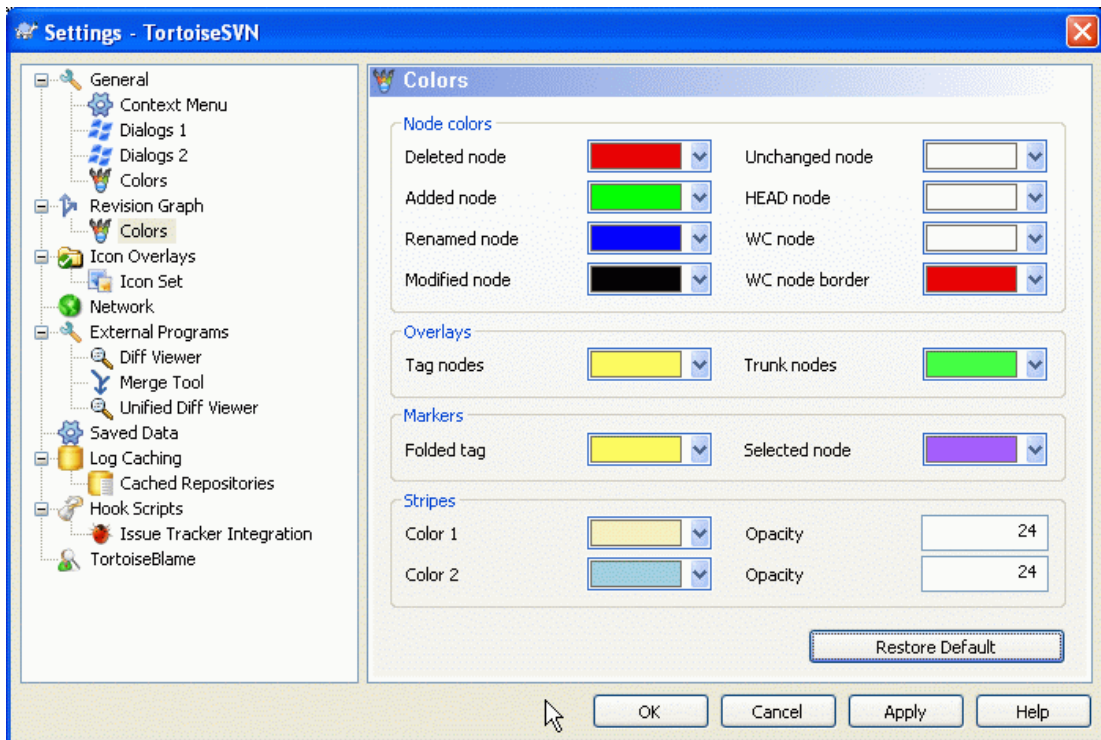
The revision graph attempts to show a clearer picture of your repository structure by distinguishing between trunk, branches and tags. As there is no such classification built into Subversion, this information is extracted from the path names. The default settings assume that you use the conventional English names as suggested in the Subversion documentation, but of course your usage may vary.

Specify the patterns used to recognise these paths in the three boxes provided. The patterns will be matched case-insensitively, but you must specify them in lower case. Wild cards * and ? will work as usual, and you can use ; to separate multiple patterns. Do not include any extra white space as it will be included in the matching specification.

Modify Colors

Colors are used in the revision graph to indicate the node type, i.e. whether a node is added, deleted, renamed. In order to help pick out node classifications, you can allow the revision graph to blend colors to give an indication of both node type and classification. If the box is checked, blending is used. If the box is unchecked, color is used to indicate node type only. Use the color selection dialog to allocate the specific colors used.

4.30.2.1. Revision Graph Colors



Slika 4.57. The Settings Dialog, Revision Graph Colors Page

This page allows you to configure the colors used. Note that the color specified here is the solid color. Most nodes are colored using a blend of the node type color, the background color and optionally the classification color.

Deleted Node

Items which have been deleted and not copied anywhere else in the same revision.

Added Node

Items newly added, or copied (add with history).

Renamed Node

Items deleted from one location and added in another in the same revision.

Modified Node

Simple modifications without any add or delete.

Unchanged Node

May be used to show the revision used as the source of a copy, even when no change (to the item being graphed) took place in that revision.

HEAD node

Current HEAD revision in the repository.

WC Node

If you opt to show an extra node for your modified working copy, attached to its last-commit revision on the graph, use this color.

WC Node Border

If you opt to show whether the working copy is modified, use this color border on the WC node when modifications are found.

Tag Nodes

Nodes classified as tags may be blended with this color.

Trunk Nodes

Nodes classified as trunk may be blended with this color.

Folded Tag Markers

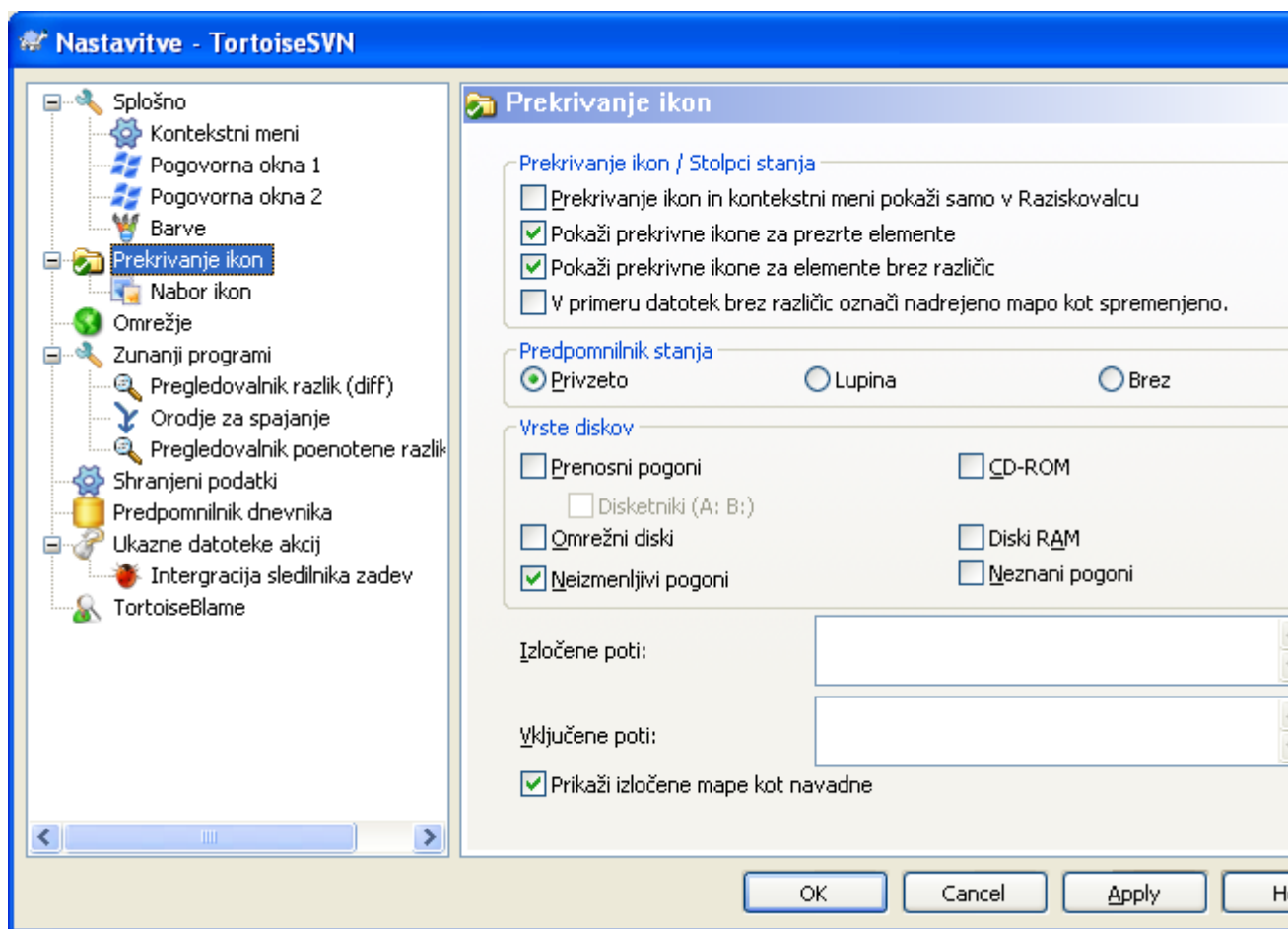
If you use tag folding to save space, tags are marked on the copy source using a block in this color.

Selected Node Markers

When you left click on a node to select it, the marker used to indicate selection is a block in this color.

Stripes

These colors are used when the graph is split into sub-trees and the background is colored in alternating stripes to help pick out the separate trees.

4.30.3. Nastavitve prekrivnih ikon

Slika 4.58. Okno nastavitve, Izbor ikon

Ta stran omogoča izbiro elementov, za katere TortoiseSVN prikaže prekrivne ikone.

Po privzetih nastavitvah se prekrivne ikone in kontekstni meni pojavijo v vseh oknih Open/Save (Odpri/Shrani) in v Raziskovalcu. Če želite, da se pojavijo *samo* v Raziskovalcu, potrdite polje **Pokaži prekrivanje ikon in kontekstni meni samo v raziskovalcu**.

Prezrti elementi in elementi brez različic so običajno brez prekrivnih ikon. Če na jih teh elementih vseeno želite, potrdite ustrezna polja.

Če izberete privzeto možnost, lahko izberete tudi možnost označevanja map kot spremenjenih, če vsebujejo elemente brez različic. To je uporabno, saj vas opozarja, da ste ustvarili nove datoteke, ki še niso dodane v skladišče. Ta možnost je na voljo le, če uporabljate *privzet* način predpomnilnika stanja (spodaj).

Ker pridobivanje informacij o stanju delovne kopije traja kar nekaj časa, TortoiseSVN uporablja predpomnilnik, tako da prikazovanje prekrivnih ikon poteka bolj gladko. Glede na vaš sistem in velikost delovne kopije izberite ustrezen tip predpomnilnika:

Prizeto

Vse informacije o stanju pridobiva poseben proces (`TSVNCache.exe`). Ta preverja spremembe na vseh pogonih in ponovno pridobi informacijo o stanju, če se datoteke znotraj delovne kopije spremenijo. Proces teče z najmanjšo možno prioriteto izvajanja, tako da ne upočasni ostalih aplikacij. To pa pomeni tudi, da se informacije o procesu ne pridobivajo v *realnem času*, zato se lahko sprememba prekrivne ikone zgodi šele čez nekaj sekund.

Prednost: prekrivne ikone prikazujejo status rekurzivno: če se spremeni datoteka nekje globoko v hierarhiji map, se prekrivne ikone spremenijo vsem mapam do korenske mape delovne kopije. Ker lahko proces pošilja obvestila o spremembah ukazni lupini, se običajno spremenijo tudi prekrivne ikone map na levi strani Raziskovalca (drevo strukture map).

Slabost: proces teče neprestano, tudi če ne delate na projektih. Uporablja pa tudi 10 do 50 MB pomnilnika (RAM), odvisno od števila in velikosti delovnih kopij.

Lupina

Polnjenje predpomnilnika se dogaja znotraj knjižnice za razširitev lupine, vendar zgolj za trenutno vidno mapo. Vsakič, ko se premaknete na drugo mapo, se informacije o stanju ponovno pridobijo.

Prednost: potrebuje zelo malo pomnilnika (približno 1MB), stanje pa lahko prikazuje v *realnem času*.

Slabost: Ker se v predpomnilnik shranijo le informacije o eni mapi, se prekrivne ikone ne prikazujejo rekurzivno. Pri velikih delovnih kopijah lahko prikaz mape traja dalj časa kot pri privzetih nastavitvah. Stolpec, ki prikazuje tip MIME, ni na voljo.

Brez predpomnilnika

Pri tej nastavitvi TortoiseSVN ne pridobiva stanja delovne kopije v Raziskovalcu. Datoteke nimajo prekrivnih ikon, mape pa imajo prekrivno ikono 'običajno', če so pod nadzorom različic. Druge prekrivne ikone se ne prikazujejo, prav tako niso na voljo dodatni stolpci z informacijami.

Prednost: ne uporablja nobenega dodatnega spomina in ne upočasni delovanja Raziskovalca med brskanjem.

Slabost: informacija o stanju datotek in map ni prikazana v Raziskovalcu. Če želite preveriti, ali je vaša delovna kopija spremenjena, morate uporabiti okno "Preveri spremembe".

Naslednja skupina omogoča nastavitve vrste elementov, za katere se prikažejo prekrivne ikone. Po privzetih nastavitvah so izbrani le trdi diski. Seveda lahko prikaz prekrivnih ikon popolnoma izključite, toda to sploh ni zabavno, kaj ne?

Omrežni pogoni so zelo počani, zato po privzetih nastavitvah prekrivne ikone delovnih kopij na omrežnih pogonih niso prikazane.

Pogoni USB so poseben primer, saj je njihov tip definiran s pogonom samim. Tako se nekateri pojavijo kot stalni pogoni, nekateri pa kot odstanjivi.

Za poti, nastavljene v **Izloči poti**, TortoiseSVN *ne* prikazuje prekrivnih ikon in stolpcev stanja. To je uporabno, če imate velike delovne kopije, ki vsebujejo le knjižnice, ki jih ne boste spreminjali in zato ne potrebujete informacij preko prekrivnih ikon. Primer:

Nastavitev `f:\development\SVN\Subversion` onemogoči prikaz prekrivnih ikon *le* na točno tej mapi. Prekrivne ikone še vedno vidite na vseh datotekah in mapah znotraj te mape.

Nastavitev `f:\development\SVN\Subversion*` onemogoči prikaz prekrivnih ikon na *vseh* datotekah in mapah, katerih pot se začne z `f:\development\SVN\Subversion`. To pomeni, da na datotekah in mapah pod to potjo ne boste videli prekrivnih ikon.

Enako velja tudi za nastavitev **Vključi poti**. Za te poti se prekrivne ikone prikazujejo vedno, tudi če so sicer izklopljene za ta tip pogona ali preko zgornjih nastavitev.

Uporabniki včasih vprašajo, kako te nastavitve sodelujejo. Odgovor je naslednji:

```
če je pot na seznamu vključenih poti
  prikaži prekrivne ikone
če je pot na dovoljeni vrsti pogona IN ni na seznamu izključenih poti
  prikaži prekrivne ikone
```

Če je pot na seznamu vključenih poti, so prekrivne ikone *vedno* prikazane. V nasprotnem primeru se prekrivne ikone prikažejo za vse označene tipe pogonov, *razen* če je pot na seznamu izključenih.

Te poti uporablja tudi `TSVNCache.exe`, da si omeji pregledovanje. Če želite, da pregleda samo določene mape, onemogočite vse tipe pogonov in vključite le mape, ki jih želite pregledati.



Izločite pogone SUBST

Pogosto je uporaben način dostopa do delovne kopije z uporabo ukaza `SUBST`, n. pr.

```
subst T: C:\TortoiseSVN\trunk\doc
```

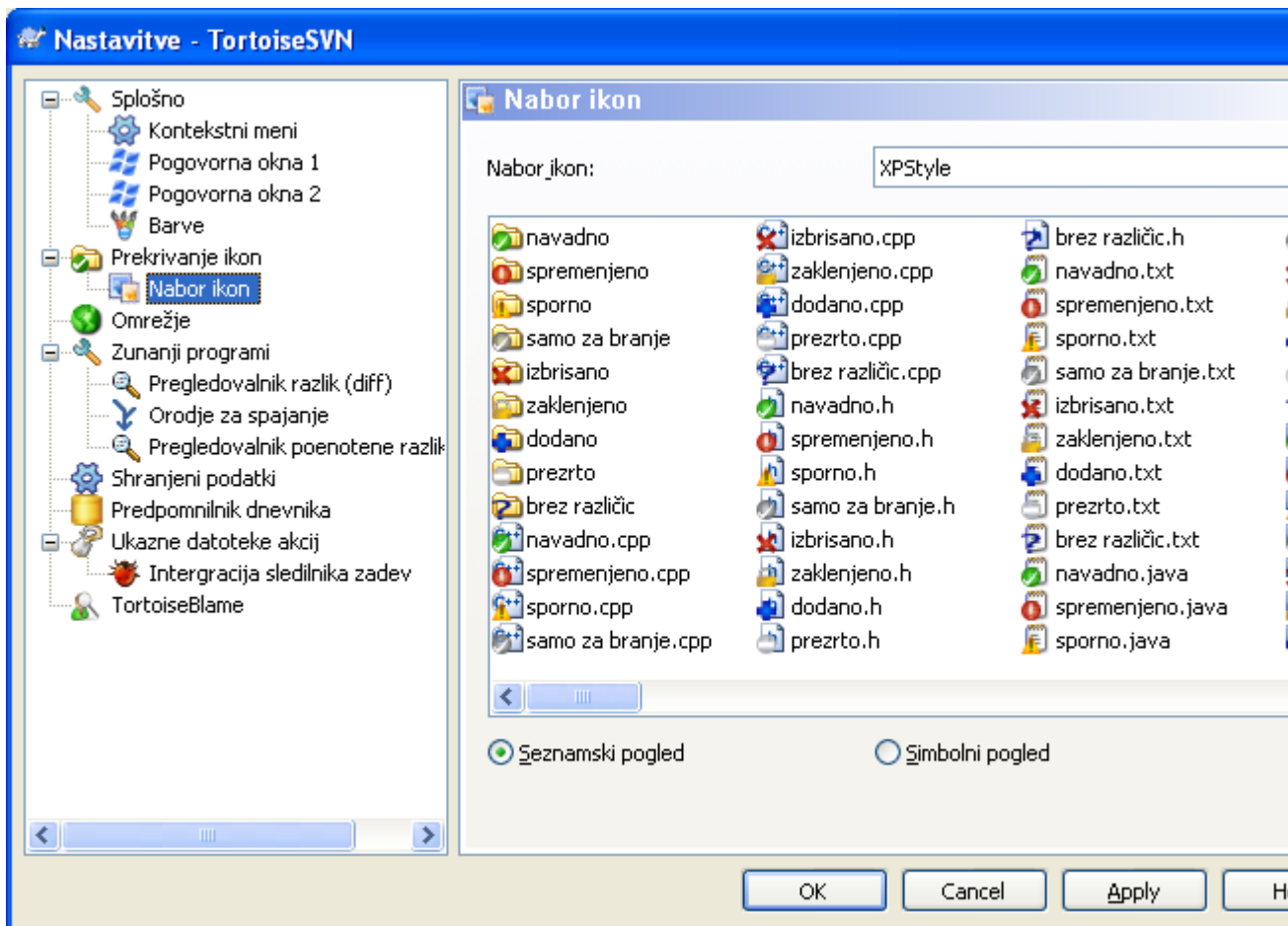
Vendar se lahko pojavijo težave pri posodabljanju prekrivnih ikon. Proces `TSVNCache` namreč dobi samo eno sporočilo, ko se spremeni datoteka; običajno je to izvirna pot. Tako se lahko zgodi, da se prekrivne ikone na poti, ki vsebuje ime pogona, narejenega z ukazom `subst`, ne bodo posodabljale.

Da se izognemo temu problemu, izvirno pot dodamo na seznam poti, ki ji `TortoiseSVN` prezre pri posodabljanju prekrivnih ikon. Na ta način se bodo posodabljale ikone na pogonu, narejenem z ukazom `subst`.

Včasih izključite območja, ki vsebujejo delovne kopije, kar povzroči, da `TSVNCache` teh področij ne preverja. Vseeno pa želite imeti takšne mape označene, da so pod nadzorom različic. Polje **Prikaži izločene mape kot 'običajne'** omogoča prav to. S potrditvijo tega polja dosežete, da se datoteke v izključenih območjih (nevključeni pogoni, izključeni pogoni) prikažejo kot običajne in sodobne, z zeleno kljukico. To vam pove, da gledate delovno kopijo, čeprav stanje, ki ga prekrivne ikone prikazujejo, morda ni pravilno. Datoteke prekrivnih ikon nimajo, vendar kontekstni meni kljub temu deluje.

Izjema sta pogona `A:` in `B:`, ki se nikoli ne upoštevata pri možnosti **Prikaži izločene mape kot 'navadne'**. V takem primeru bi namreč sistem `Windows` pregledal omenjena pogona, tudi če ju dejansko nimate, kar bi za nekaj sekund upočasnilo zaganjanje `Raziskovalca`.

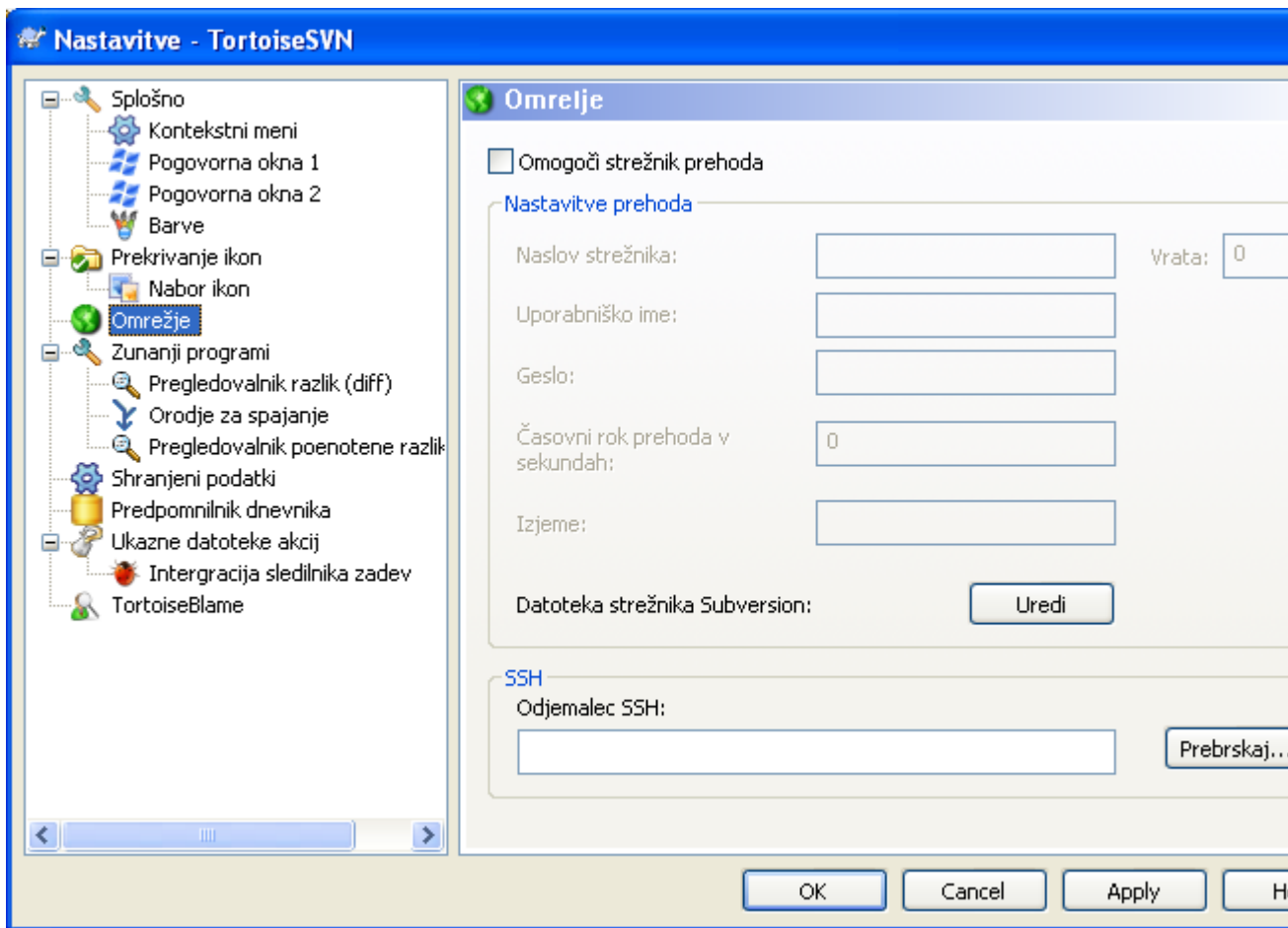
4.30.3.1. Izbira nabora ikon



Slika 4.59. Okno nastavitvev, Izbor ikon

Izberete si lahko tak nabor ikon, ki vam najbolj ustreza. Da bi uveljavili spremembe, boste morali mogoče ponovno zagnati računalnik.

4.30.4. Nastavitve omrežja



Slika 4.60. Okno nastavitvev, Omrežje

Tukaj lahko nastavite posredniški strežnik (proxy). Mogoče ga potrebujete, da lahko delate preko požarnega zidu v podjetju.

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/1.5/svn.advanced.confarea.html) [http://svnbook.red-bean.com/en/1.5/svn.advanced.confarea.html] for details on how to use this file.

Določite lahko aplikacijo, ki naj jo TortoiseSVN uporabi za vzpostavitev varne povezave do skladišča svn +ssh. Priporočamo uporabo programa TortoisePlink.exe. To je posebna različica priljubljenega programa Plink in je del paketa TortoiseSVN. Prevedena je kot aplikacija brez okna, tako da se ob avtentikaciji ne pojavi okno DOS.

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"
```

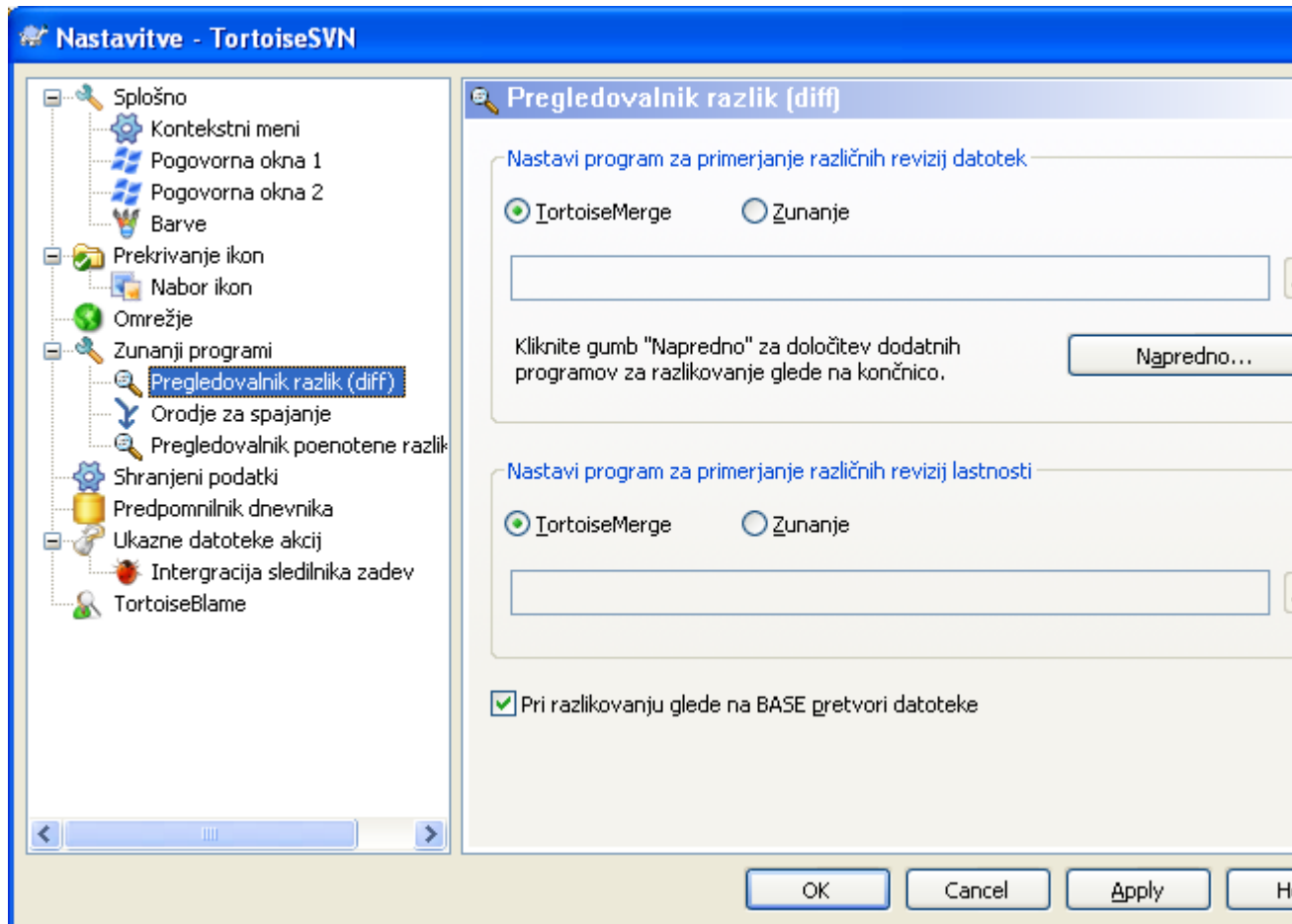
Stranski učinek aplikacije brez okna je, da ta ne more izpisati sporočila o napaki. Če avtentikacija ne uspe, dobite sporočilo tipa "Ne morem pisati na standardni izhod". Zato priporočamo, da najprej uporabite standardno različico programa Plink. Ko vse deluje, pa jo zamenjate s programom TortoisePlink. Uporabite popolnoma enake parametre.

TortoisePlink nima lastne dokumentacije, ker se le malo razlikuje od programa Plink. Parametrd ukazne vrstice lahko najdete na naslovu [PuTTY](http://www.chiark.greenend.org.uk/~sgtatham/putty/) [http://www.chiark.greenend.org.uk/~sgtatham/putty/]

Da vam ne bi bilo potrebno vsakokrat vpisovati uporabniškega imena in gesla, lahko uporabite orodje za shranjevanje gesel, naprimer Pageant, ki je na voljo na domači strani PuTTY.

Nastavljanje SSH na strežniku in odjemalcih je malce bolj obsežen postopek, ki ne sodi v okvir te pomoči. Navodila lahko najdete med pogosto zastavljeni vprašanji TortoiseSVN: [Subversion/TortoiseSVN SSH How-To](http://tortoisesvn.net/ssh_howto) [http://tortoisesvn.net/ssh_howto].

4.30.5. Nastavitve zunanjih programov



Slika 4.61. Okno nastavitve, ogledovalnik razlik

Tukaj lahko nastavite programe za razlikovanje/spajanje, ki naj jih TortoiseSVN uporabi. Privzeta nastavitve je uporaba programa TortoiseMerge, ki se namesti skupaj s TortoiseSVN.

Preberite [Razdelek 4.10.5, "Zunanja orodja za razlikovanje/spajanje"](#), kjer so navedena nekatera zunanja orodja za razlikovanje/spajanje, ki jih uporabniki uporabljajo skupaj s sistemom TortoiseSVN.

4.30.5.1. Pregledovalnik razlik

Za primerjanje razlik med različnimi revizijami datotek lahko uporabljate zunanji program za razlikovanje. Zunanji program mora v ukazni vrstici dobiti imena datotek in parametre ukazne vrstice. TortoiseSVN uporablja parametre za zamenjavo s predpono %. Ko naleti na parameter, ga zamenja z ustrežno vrednostjo. Vrstni red parametrov je odvisen od programa za razlikovanje, ki ga uporabljate.

%base

Izvirna datoteka brez vaših sprememb

%bname

Naslov okna osnovne datoteke

`%mine`

Vaša lastna datoteka z vašimi spremembami

`%yname`

Naslov okna za vašo datoteko

Naslovi oken niso zgolj imena datotek. TortoiseSVN jih obravnava kot nazive za prikaz in ustvari ustrezna imena. N. pr. če razlikujete datoteko revizije 123 z datoteko v delovni kopiji, bosta imeni za prikaz imedatoteke : revizija 123 in ime datoteke : delovna kopija.

Naprimmer: z ExamDiff Pro:

```
C:\Pot-do\ExamDiff.exe %base %mine
```

ali s KDiff3:

```
C:\Pot-do\kdiff3.exe %base %mine --L1 %bname --L2 %yname
```

ali z WinMerge:

```
C:\Pot-do\WinMerge.exe -e -ub -dl %bname -dr %yname %base %mine
```

ali z Araxis:

```
C:\Pot-do\compare.exe /max /wait /title1:%bname /title2:%yname
%base %mine
```

Če uporabljate lastnost `svn:keywords` za razširitev ključnih besed, posebej če uporabite *revizijo* datoteke, se med datotekami pojavi razlike zgolj zaradi različne vrednosti ključnih besed. Podobno je pri uporabi lastnosti `svn:eol-style = native`: osnovna datoteka (BASE) bo uporabljala LF za zaključek vrstice, vaša datoteka pa CR-LF. TortoiseSVN običajno te razlike skriva, tako da pred primerjanjem tudi v osnovni datoteki razširi ključne besede in spremeni zaključke vrstic. To pa lahko traja precej časa, če delamo z dolgimi datotekami. Če je polje **Pri razlikovanju glede na BASE pretvori datoteke** izklopljeno, bo TortoiseSVN preskočil predprocesiranje datotek.

Lahko pa določite tudi drugo orodje za razlikovanje lastnosti v sistemu Subversion. Ker so lastnosti ponavadi enostavna kratka besedila, vam bo mogoče bolj ustrezal enostavnejši pregledovalnik.

Če ste nastavili dodatna orodja za razlikovanje, lahko iz kontekstnega menija izbere TortoiseMerge *in* ostala orodja. **Kontekstni meni** → **Razlikuj** uporabi osnovno orodje za razlikovanje, **Shift+Kontekstni meni** → **Razlikuj** pa dodatnega.

4.30.5.2. Orodje za spajanje

Zunanji program za spajanje, uporabljen za reševanje spornih datotek. Zamenjava parametrov se uporablja prav tako kot pri programu za razlikovanje.

`%base`

izvirna datoteka brez vaših sprememb ali sprememb ostalih

`%bname`

Naslov okna osnovne datoteke

`%mine`

vaša datoteka z vašimi spremembami

`%yname`
Naslov okna za vašo datoteko

`%theirs`
datoteka, kot se nahaja v skladišču

`%tname`
naslov okna za datoteko v skladišču

`%merged`
sporna datoteka, rezultat spajanja

`%mname`
naslov okna za spojeno datoteko

Naprimer: s Perforce Merge:

```
C:\Path-To\P4Merge.exe %base %theirs %mine %merged
```

ali s KDiff3:

```
C:\Path-To\kdiff3.exe %base %mine %theirs -o %merged  
--L1 %bname --L2 %yname --L3 %tname
```

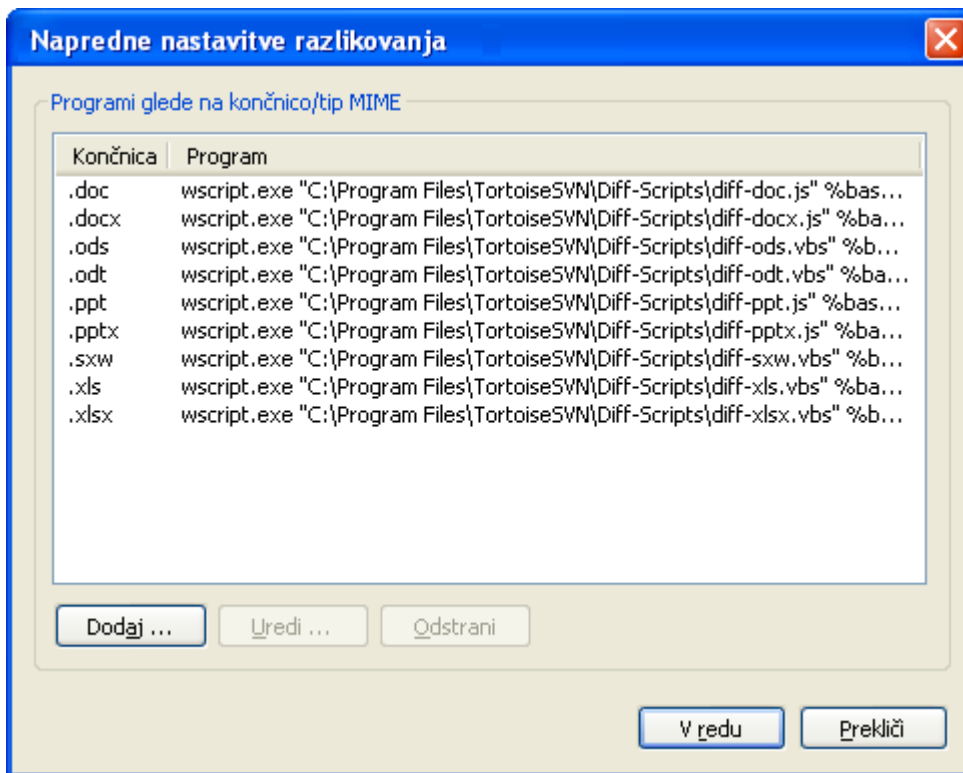
ali z Araxis:

```
C:\Path-To\compare.exe /max /wait /3 /title1:%tname /title2:%bname  
/title3:%yname %theirs %base %mine %merged /a2
```

ali z WinMerge (2.8 ali novejši):

```
C:\Path-To\WinMerge.exe %merged
```

4.30.5.3. Napredne nastavitve za razlikovanje/spajanje



Slika 4.62. Okno nastavitve, napredne nastavitve razlikovanja/spajanja

V naprednih nastavitvah lahko nastavite različna orodja za razlikovanje in spajanje glede na končnico datoteke. Primer: Photoshop nastavite kot program za "razlikovanje" datotek `.jpg` :-). Tudi lastnost `svn:mime-type` lahko asociirate s programom za razlikovanje ali spajanje.

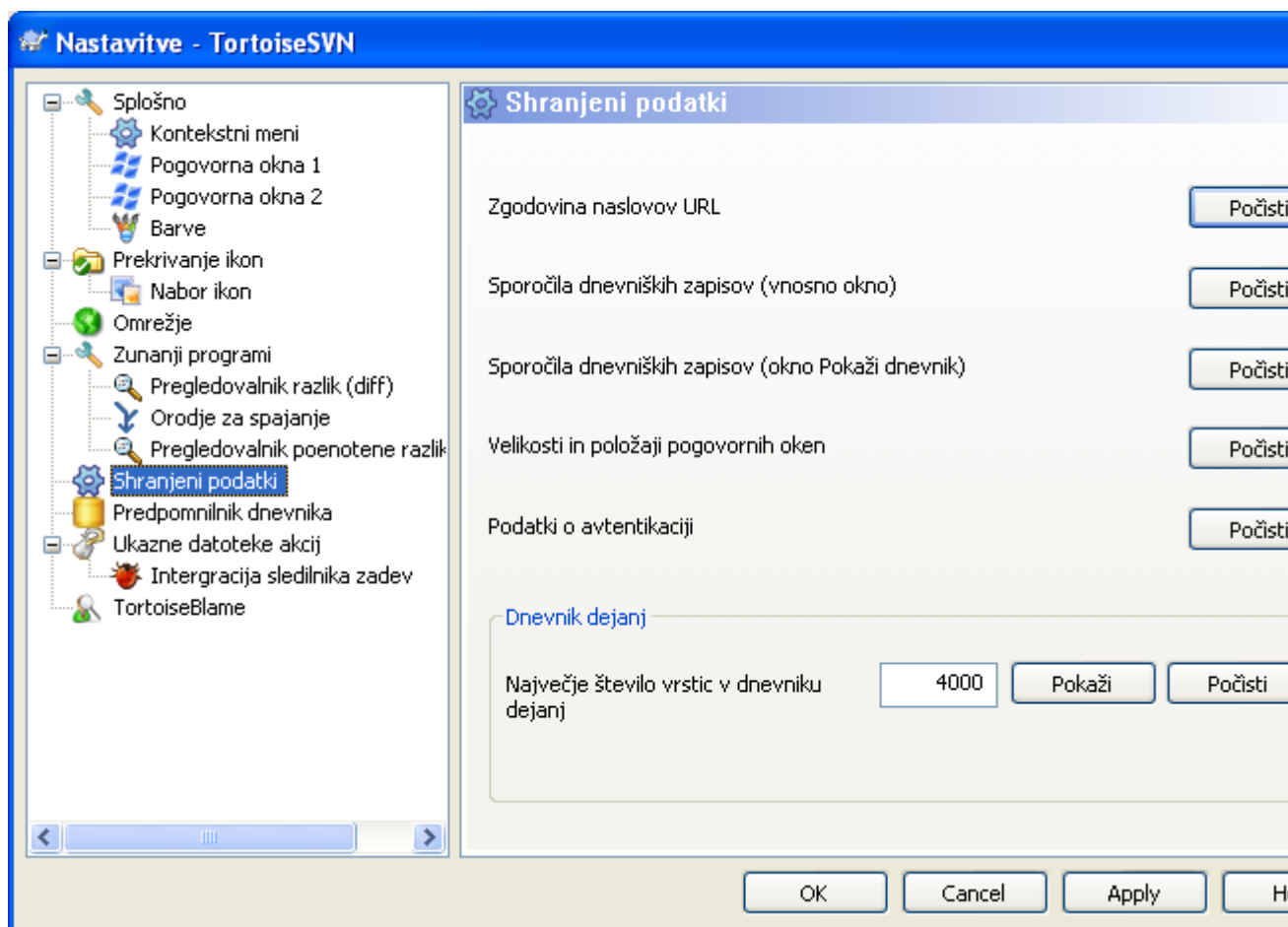
Za povezavo orodij s končnico datotek morate navesti končnico. Uporabite `.bmp` za datoteke bitmap sistema Windows. Za povezavo orodij z lastnostjo `svn:mime-type` nastavite tip MIME, vključno s poševnico, n.pr. `text/xml`.

4.30.5.4. Pregledovalnik za poenotene razlike

Pregledovalnik za poenotene razlike (datoteke popravkov). Noben parameter ni obvezen. Možnost **Default** pove, da je potrebno poiskati program, ki je povezan s končnico `.diff` in s končnico `.txt`. Če pregledovalnika za datoteke `.diff` nimate, vam bo po vsej verjetnosti ponujena Beležka.

Originalna Beležka sistema Windows ne deluje najbolje pri datotekah, ki imajo nestandardne zaključke vrstic CR-LF. Ker imajo datoteke poenotene razlike večinoma le zaključke LF, se jih v Beležki ne da dobro pregledovati. Lahko pa si prenesete brezplačno zamenjavo za Beležko. *Notepad2* [<http://www.floss-freeware.ch/notepad2.html>], poleg tega, da pravilno prikaže zaključke datotek, uporablja tudi barvno kodiranje za dodane in odstranjene vrstice.

4.30.6. Shranjeni podatki



Slika 4.63. Okno nastavitvev, Shranjeni podatki

Za večjo uporabnost TortoiseSVN shrani veliko nastavitvev, ki jih uporabljate, in si zapomni vaše zadnje operacije. Če želite, lahko tu te podatke izbrišete.

Zgodovina naslovov URL

Ko naredite prevzem delovne kopije, spojite spremembe ali uporabite brskalnik po skladišču, vam TortoiseSVN ponudi shranjene zadnje uporabljene naslove URL v spustnem meniju. Včasih je seznam poln starih naslovov in ga je zato koristno počistiti.

Če želite iz spustnega polja izbrisati posamezne elemente, lahko to storite kar na polju samem. Kliknite na puščico za spust spustnega polja, premaknite miško na element, ki ga želite odstraniti in stisnite **Shift+Del**.

Dnevniška sporočila (vnosno okno)

TortoiseSVN shrani zadnjih nekaj sporočil dnevniških zapisov, ki jih vnesete in sicer za vsako skladišče posebej. Če uporabljate veliko skladišč, postane ta seznam precej dolg.

Dnevniška sporočila (okno Prikaži dnevnik)

TortoiseSVN si ob prikazu okna sporočila dnevniških zapisov shrani v predpomnilnik, tako da naslednjič, ko jih mora prikazati, prihrani čas. Če nekdo dnevniški zapis popravi, vi pa ta zapis že imate v predpomnilniku, te spremembe ne boste videli, dokler ne izbrišete predpomnilnika. Predpomnenje dnevniških zapisov se vključi v zavihku **Predpomnilnik dnevnika**.

Velikosti in položaji pogovornih oken

Veliko oken si zapomni velikost in položaj na zaslonu ob zadnji uporabi.

Podatki avtentikacije

Ko uporabljate avtentikacijo s strežnikom Subversion, se uporabniško ime in geslo shranita v krajevni datoteki in vam ju tako ni potrebno vedno znova vnašati. Te podatke boste morda želeli izbrisati - zaradi varnostnih razlogov ali pa zato, ker želite do skladišča dostopati z drugim uporabniškim imenom ... ali Janez ve, da uporabljate njegov PC?

Če želite izbrisati podatke o avtentikaciji samo za nek točno določen strežnik, preberite [Razdelek 4.1.5, "Avtentikacija"](#). Tu je opisano, kje najdete podatke predpomnilnika.

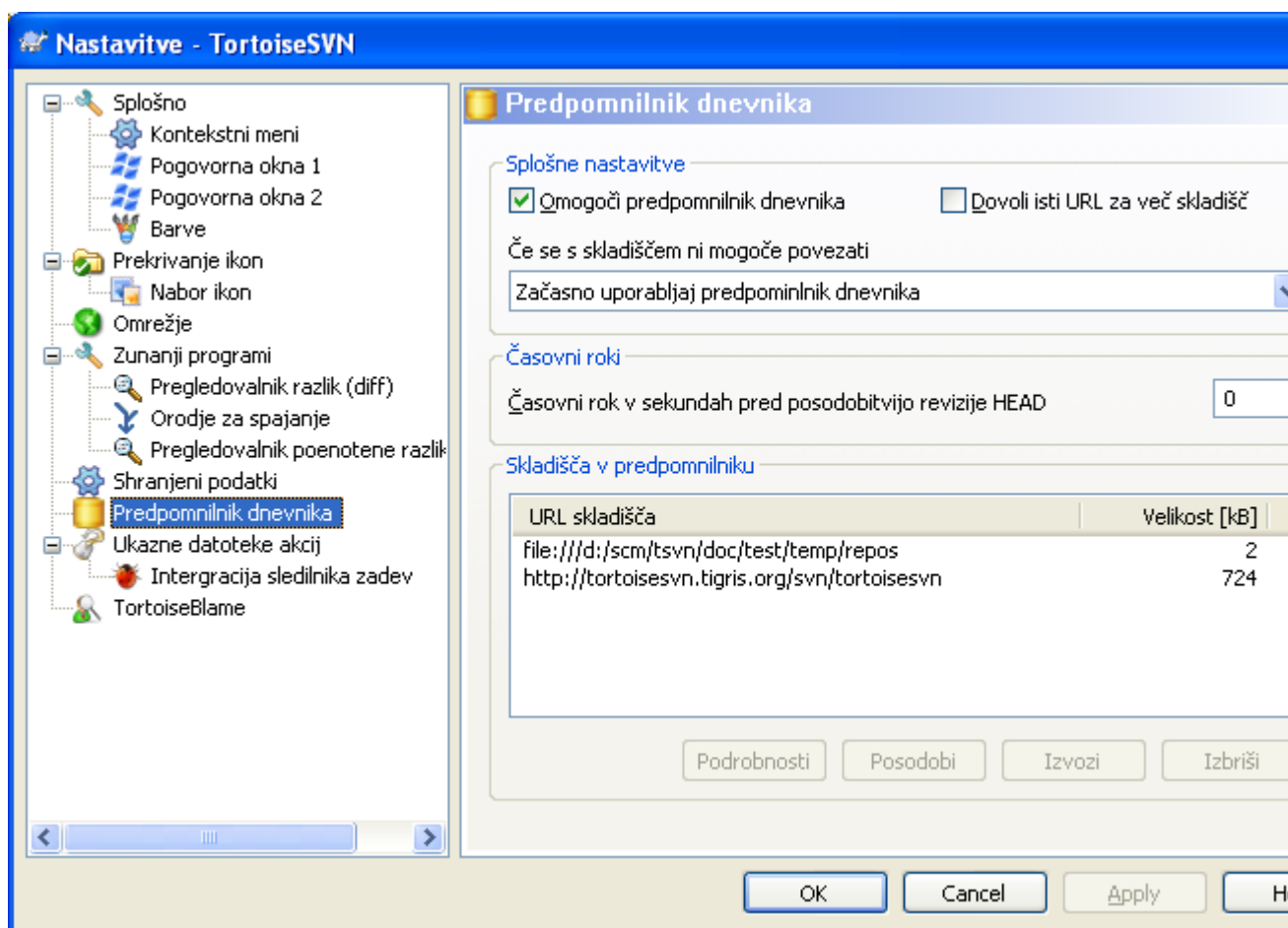
Zapisi dejanj

TortoiseSVN hrani zapise vsega, kaj se izpisuje v pogovorna okna napredka. To je uporabno, če želite, na primer, preveriti, kaj se je zgodilo pri zadnji posodobitvi delovne kopije.

Datoteka zapisa je omejena na določeno dolžino. Ko to dolžino preseže, se najstarejši zapisi odstranijo. Po privzetih nastavitvah se ohrani 4000 vrstic, vendar lahko to številko spremenite.

Tukaj lahko pogledate vsebino zapisov, prav tako pa lahko vsebino počistite.

4.30.7. Predpomnenje dnevnika



Slika 4.64. Okno nastavitvev, Predpomnilnik dnevnika

To okno omogoča nastavitvev predpomnenja. Predpomnenje shrani krajevno kopijo dnevnških zapisov in sprememb poti, tako da se prihrani dolgotrajno prenašanje teh podatkov s strežnika. Uporaba predpomnenja drastično skrajša čas, potreben za prikaz okna dnevnških zapisov in grafa revizij. Dodatna prednost je, da so ti podatki na voljo, ko strežnik ni dosegljiv.

Omogoči predpomnilnik dnevnika

Omogoči predpomnenje dnevnika ob vsakem zahtevku. Če je možnost omogočena, se podatki - če so na voljo - preberejo iz predpomnilnika. Če podatkov ni, se preberejo iz strežnika in dodajo v predpomnilnik.

Če je predpomnenje onemogočeno, se podatki vedno prenesejo neposredno s strežnika in se ne shranjujejo na krajevnem pogonu.

Allow ambiguous URLs

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, leave it unchecked to improve performance.

Allow ambiguous UUIDs

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, leave it unchecked to improve performance.

Če se s skladiščem ni mogoče povezati

Če delate nepovezano ali če je strežnik skladišča nedostopen, lahko predpomnilnik uporabljate za pregled dnevniških zapisov, ki se že nahajajo na krajevnem pogonu. Seveda je predpomnilnik lahko zastarel, zato lahko izbirate, ali boste to možnost sploh uporabljali.

Če so podatki o dnevniku pridobljeni brez dostopa do strežnika, je to stanje prikazano v naslovni vrstici pogovornega okna.

Časovni rok pred posodabljanjem revizije HEAD

Ko odprete pogovorno okno dnevnika, ponavadi želite na strežniku preveriti, ali obstajajo novejši zapisi. Če je ta vrednost različna od nič, se bo preverjanje zgodilo le v primeru, da je od zadnjega preverjanja pretekel nastavljeni čas. S tem zmanjšate število dostopov do strežnika, vendar podatki morda ne bodo najsodobnejši. Če želite uporabljati to zmožnost, priporočamo uporabo vrednosti 300 (5 minut) kot dober kompromis.

Days of inactivity until small caches get removed

If you browse around a lot of repositories you will accumulate a lot of log caches. If you're not actively using them, the cache will not grow very big, so TortoiseSVN purges them after a set time by default. Use this item to control cache purging.

Maximum size of removed inactive caches

Larger caches are more expensive to reacquire, so TortoiseSVN only purges small caches. Fine tune the threshold with this value.

Maximum number of tool failures before cache removal

Occasionally something goes wrong with the caching and causes a crash. If this happens the cache is normally deleted automatically to prevent a recurrence of the problem. If you use the less stable nightly build you may opt to keep the cache anyway.

4.30.7.1. Cached Repositories

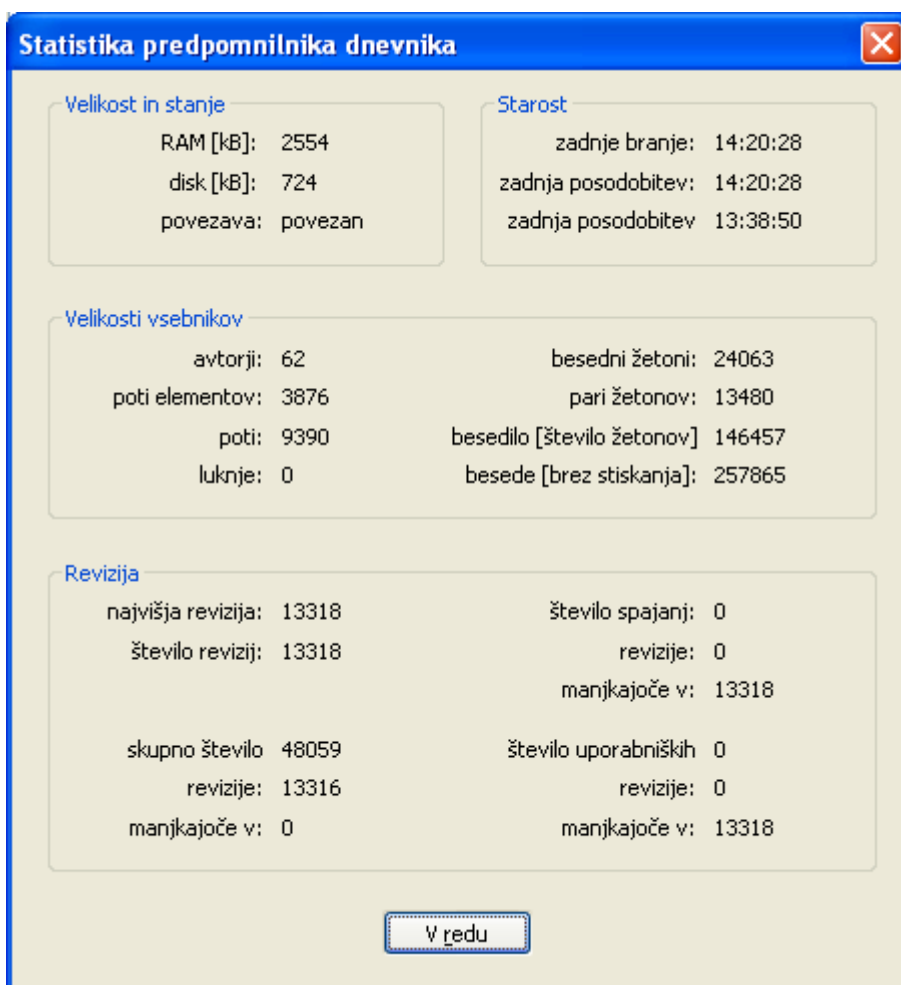
On this page you can see a list of the repositories that are cached locally, and the space used for the cache. If you select one of the repositories you can then use the buttons underneath.

Za popolno osvežitev predpomnilnika in krpanje lukenj kliknite na gumb **Posodobi**. Pri velikih skladiščih bo posodobitev trajala precej časa. Postopek je uporaben, če nameravate delati nepovezano in želite imeti najboljši možni predpomnilnik.

Kliknite na gumb **Izvozi**, če želite izvoziti celoten predpomnilnik v datoteko CSV. To je uporabno, če želite predelati podatke z uporabo zunanjega orodja. V glavnem pa ta ukaz uporabljajo razvijalci sistema.

Če želite izbrisati celoten predpomnilnik za izbrano skladišče, kliknite na gumb Izbriši. S tem ne onemogočite predpomnenja, tako da se bo ob naslednjih zahtevkih predpomnilnik ponovno pričel polniti.

4.30.7.2. Statistika predpomnilnika dnevnika



Slika 4.65. Okno nastavitv, Statistika predpomnilnika dnevnika

Za podrobno statistiko posameznega predpomnilnika kliknite na gumb Podrobnosti. Veliko polj je zanimivih zgolj za razvijalce sistema TortoiseSVN, zato le-ta niso podrobneje razložena.

RAM

Spomin, potreben za uporabo predpomnilnika

Disk

Velikost predpomnilnika na disku. Podatki so stisnjeni, tako da je količina porabljenega prostora dokaj skromna.

Povezava

Pove, ali je bilo skladišče dostopno ob zadnji uporabi predpomnilnika.

Zadnja posodobitev

Čas zadnje spremembe vsebine predpomnilnika.

Zadnja posodobitev revizije HEAD

Čas zadnjega zahtevka revizije HEAD s strežnika

Avtorji

Število avtorjev dnevniških zapisov v predpomnilniku.

Poti

Število poti, kot jih prikaže ukaz `svn log -v`.

Število lukenj predpomnilnika

Število območij revizij, ki niso bila prenesena zato, ker niso bila zahtevana. To je merilo za luknjavost predpomnilnika.

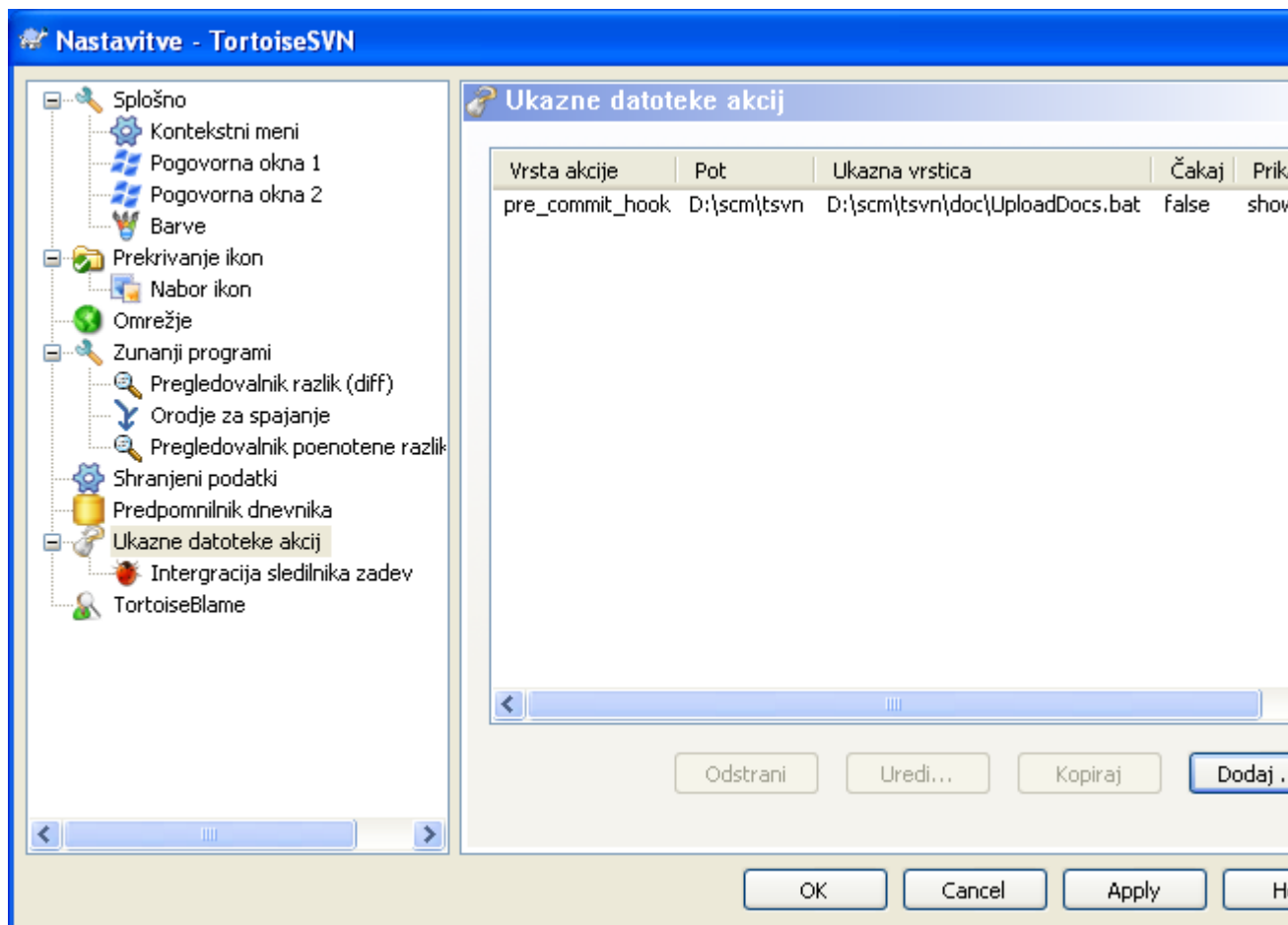
Najvišja revizija

Najvišja revizija v predpomnilniku

Število revizij

Število revizij v predpomnilniku. To je še eno merilo popolnosti predpomnilnika.

4.30.8. Ukazne datoteke akcij na strani odjemalca

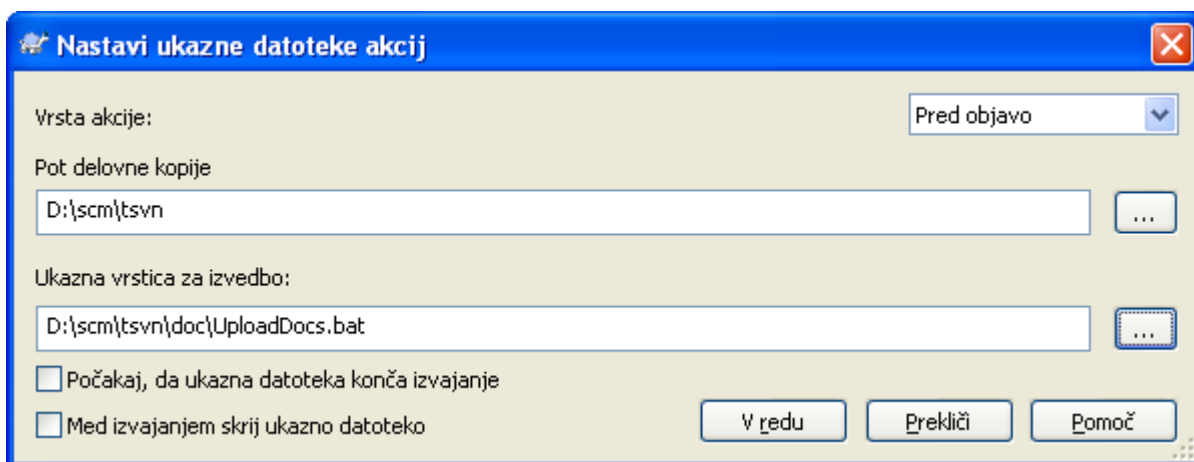


Slika 4.66. Okno nastavitvev, Ukazne datoteke akcij

To okno vam omogoča nastavitve ukaznih datotek akcij, ki se izvedejo samodejno ob določeni operaciji sistema Subversion. V nasprotju s ukaznimi datotekami akcij, ki so razložene v [Razdelek 3.3, "Server side hook scripts"](#), se te datoteke izvedejo krajevno na strani odjemalca.

Ena možnost uporabe skripte akcije je klicanje programa `SubWCRev.exe`, ki zamenja številke revizij po objavi, ali pa sproži ponovno gradnjo projekta.

Zaradi različnih varnostnih in izvedbenih razlogov so ukazne datoteke akcij definirane krajevno na računalniku namesto v projektnih lastnostih. Tako sami definirate, kaj se bo zgodilo, ne glede na to, kaj nek drug uporabnik objavi v skladišču. Seveda pa lahko ukazno datoteko akcije nastavite tako, da kliče datoteko, ki je sama pod nadzorom različic.



Slika 4.67. Okno nastavitvev, nastavitvev ukaznih datotek akcij

Za dodajanje nove ukazne datoteke akcije preprosto kliknite na gumb Dodaj in izpolnite podrobnosti.

Zaenkrat je na razpolago šest tipov ukaznih datotek akcij

Akcija pred prikazom okna objave

Called before the commit dialog is shown. You might want to use this if the hook modifies a versioned file and affects the list of files that need to be committed and/or commit message. However you should note that because the hook is called at an early stage, the full list of objects selected for commit is not available.

Akcija pred objavo

Called after the user clicks OK in the commit dialog, and before the actual commit begins. This hook has a list of exactly what will be committed.

Akcija po objavi

Pokliče se, ko se objava zaključi (uspešno ali neuspešno).

Akcija pred prikazom okna posodobitve

Pokliče se, preden se prikaže pogovorno okno "Posodobi na revizijo".

Akcija pred posodobitvijo

Pokliče se, preden se posodobitev začne.

Akcija po posodobitvi

Pokliče se, ko se posodobitev zaključi (uspešno ali neuspešno).

Ukazna datoteka akcije se definira za določeno pot v delovni kopiji. Nastaviti jo morate le za najvišjo mapo. Če operacijo izvedete na neki podmapji, bo TortoiseSVN samodejno iskal navzgor po hierarhiji.

Nato morate določiti ukazno vrstico, ki naj se izvede, začenši s potjo do ukazne ali izvršne datoteke. Določite lahko paketno datoteko, izvršno datoteko ali katerokoli datoteko, ki ima veljavno asociacijo s programom v sistemu Windows, n. pr. ukazno datoteko v jeziku perl.

The command line includes several parameters which get filled in by TortoiseSVN. The parameters passed depend upon which hook is called. Each hook has its own parameters which are passed in the following order:

Akcija pred prikazom okna objave
PATHMESSAGEFILECWD

Akcija pred objavo
PATHDEPTHMESSAGEFILECWD

Akcija po objavi
PATHDEPTHMESSAGEFILEREVISIONERRORCWD

Akcija pred prikazom okna posodobitve
PATHCWD

Akcija pred posodobitvijo
PATHDEPTHREVISIONCWD

Akcija po posodobitvi
PATHDEPTHREVISIONERRORCWD

Pomen vsakega izmed teh parametrov je podan tukaj:

PATH

Pot do začasne datoteke, ki vsebuje vse poti, za katere je bila izvedena operacija. Vsaka pot je v svoji vrstici začasne datoteke.

DEPTH

Globina, do katere se izvede objava/posodobitev.

Možne vrednosti spremenljivk:

-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

Pot do datoteke, ki vsebuje dnevniški zapis za objavo. Datoteka vsebuje besedilo v obliki UTF8. Po uspešni izvršbi se besedilo dnevniškega zapisa prebere, tako da ga akcijska ukazna datoteka lahko spremeni.

REVISION

Revizija skladišča, na katero je potrebno posodobiti delovno kopijo, ali revizija po uspešni objavi.

ERROR

Pot do datoteke, ki vsebuje sporočilo o napaki. Če napake ni, bo datoteka prazna.

CWD

Trenutna delovna mapa, na kateri se izvede ukazna datoteka. Nastavljena na skupno korensko mapo vseh uporabljenih poti.

Note that although we have given these parameters names for convenience, you do not have to refer to those names in the hook settings. All parameters listed for a particular hook are always passed, whether you want them or not ;-)

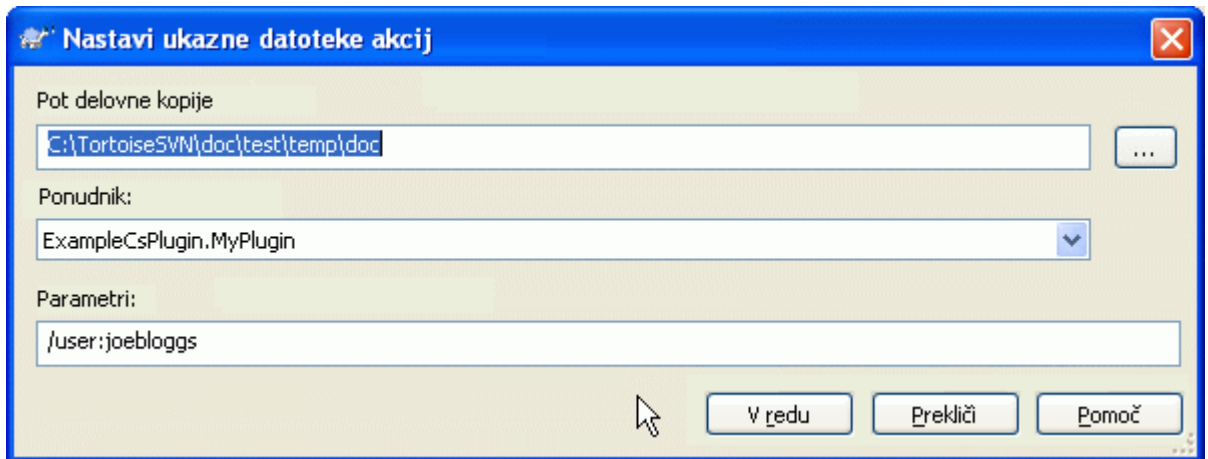
Če želite, da operacija Subversion čaka, dokler se ukazna datoteka akcije ne zaključi, potrdite polje Počakaj, da ukazna datoteka konča izvajanje.

Običajno ne želite videti nepriljubljenih oken sistema DOS, ko poženet ukazno datoteko, zato je možnost Med izvajanjem skrij ukazno datoteko po privzetih nastavitvah vključena.

Sample client hook scripts can be found in the `contrib` folder in the *TortoiseSVN repository* [<http://tortoisesvn.googlecode.com/svn/trunk/contrib/hook-scripts>]. (**Razdelek 3, "TortoiseSVN je brezplačen!"** explains how to access the repository).

4.30.8.1. Integracija sledilnika zadev

TortoiseSVN lahko uporabi vtičnik COM za pregledovanje sledilnika zadev v pogovornem oknu za objave. Uporaba vtičnikov je opisana v **Razdelek 4.28.2, "Pridobivanje informacij iz sledilnika zadev"**. Če vam je skrbnik sistema dodelil vtičnik, ki ste ga že namestili in registrirali, sedaj tu določite, kako se vključi v delovno kopijo.



Slika 4.68. Okno za nastavitve, Okno za integracijo sledilnika zadev

Za uporabo vtičnika na določeni delovni kopiji kliknite na gumb Dodaj... Tu določite pot delovne kopije, iz spustnega seznama registriranih vtičnikov izberete vtičnik, ki ga želite uporabiti, in navedete parametre. Parametri so odvisni od vtičnika, pametno pa je vključiti uporabniško ime sledilnika zadev, tako da vtičnik lahko poišče zadeve, ki so določene vam.

If you want all users to use the same COM plugin for your project, you can specify the plugin also with the properties `bugtraq:provideruuid` and `bugtraq:providerparams`.

`bugtraq:provideruuid`

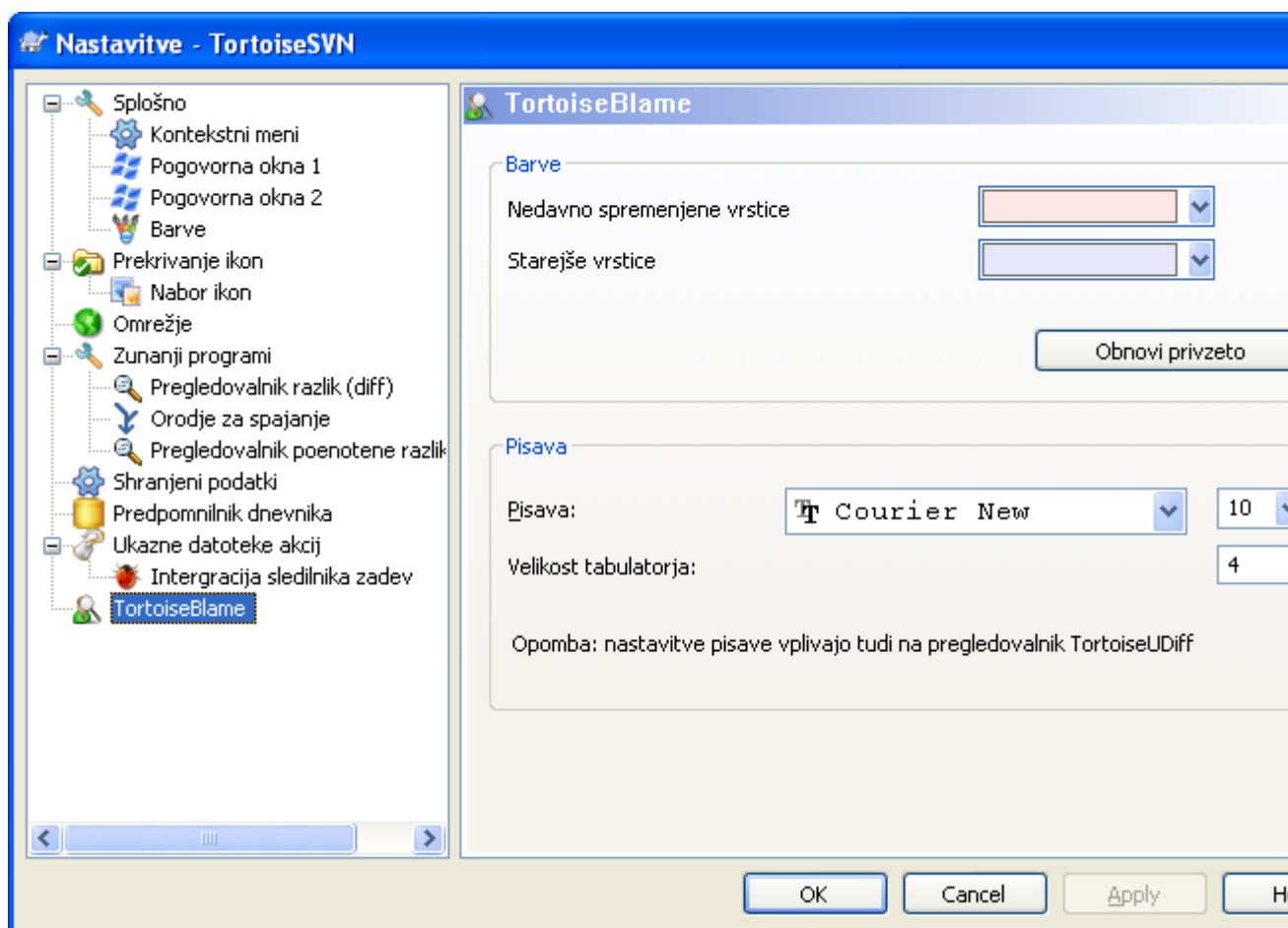
This property specifies the COM UUID of the `IBugtraqProvider`, for example `{91974081-2DC7-4FB1-B3BE-0DE1C8D6CE4E}`. (this example is the UUID of the *Gurtle bugtraq provider* [<http://code.google.com/p/gurtle/>], which is a provider for the *Google Code* [<http://code.google.com/hosting/>] issue tracker).

`bugtraq:providerparams`

This property specifies the parameters passed to the `IBugtraqProvider`.

Please check the documentation of your `IBugtraqProvider` plugin to find out what to specify in these two properties.

4.30.9. Nastavitve TortoiseBlame



Slika 4.69. Okno za nastavitve, TortoiseBlame

Nastavitve, ki jih uporablja TortoiseBlame, se nastavljejo v glavnem kontekstnem meniju in ne neposredno v programu TortoiseBlame.

Barve

TortoiseBlame lahko uporablja različne barve ozadja za označevanje starosti vrstic v datoteki. Nastavite začetno in končno barvo za najnovejšo in najstarejšo revizijo, TortoiseBlame pa uporabi linearno interpolacijo za določitev barve posameznih vrstic glede na številko revizije.

Pisava

Izberete lahko tip in velikost pisave za prikazano besedilo. Nastavitev velja tako za vsebino datoteke kot za informacije na levi strani (avtor, revizija).

Tabulatorji

Definira, koliko presledkov nadomesti tabulatorje, ki jih najde v datoteki.

4.30.10. Nastavitve registra

A few infrequently used settings are available only by editing the registry directly. It goes without saying that you should only edit registry values if you know what you are doing.

Konfiguracija

Konfiguracijsko datoteko za Subversion lahko postavite na drugo lokacijo z uporabo naslednje lokacije v registru: `HKCU\Software\TortoiseSVN\ConfigDir`. Sprememba ima vpliv na vse operacije TortoiseSVN.

Ikona predpomnilnika v sistemskem traku

Če želite dodati za program TSVNCache ikono v sistemski trak, ustvarite ključ `DWORD` z vrednostjo na lokaciji `HKCU\Software\TortoiseSVN\CacheTrayIcon`. V resnici je to uporabno le za razvijalce, ki želijo pravilno zaustaviti program.

Debug

To show the command line parameters passed from the shell extension to `TortoiseProc.exe` create a `DWORD` key with a value of `1` at `HKCU\Software\TortoiseSVN\Debug`.

Context Menu Icons

This can be useful if you use something other than the windows explorer or if you get problems with the context menu displaying correctly. create a `DWORD` key with a value of `0` at `HKCU\Software\TortoiseSVN>ShowContextMenuIcons` if you don't want TortoiseSVN to not show icons for the shell context menu items. Set this value to `1` to show the icons again.

Block Overlay Status

If you don't want the explorer to update the status overlays while another TortoiseSVN command is running (e.g. Update, Commit, ...) then create a `DWORD` key with a value of `1` at `HKCU\Software\TortoiseSVN\BlockStatus`.

Update Check URL

`HKCU\Software\TortoiseSVN\UpdateCheckURL` contains the URL from which TortoiseSVN tries to download a text file to find out if there are updates available. You can also set this under `HKLM` instead of `HKCU` if you want, but `HKCU` overwrites the setting in `HKLM`. This might be useful for company admins who don't want their users to update TortoiseSVN until they approve it.

Imena datotek brez končnic v seznamu za samodokončanje

Seznam za samodokončanje, ki se prikaže v urejevalniku sporočil v oknu za objave, prikaže datoteke, ki so predvidene za objavo. Če želite, da se prikažejo tudi imena brez končnic, na naslovu `HKCU\Software\TortoiseSVN\AutocompleteRemovesExtensions` ustvarite ključ `DWORD` z vrednostjo `1`

Stolpci v Raziskovalcu vsepovsod

Dodatni stolpci, ki jih TortoiseSVN doda podrobnemu pogledu v Raziskovalcu, so običajno vidni le v delovni kopiji. Če želite, da so dostopni tudi drugje, na naslovu `HKCU\Software\TortoiseSVN\ColumnsEveryWhere` ustvarite ključ `DWORD` z vrednostjo `1`.

Ločilnik dnevniškega zapisa spajanja

Pri spajanju revizij z druge veje in če so informacije o sledenju spajnja na voljo, se sporočila dnevniških zapisov zberejo in tvorijo novo sporočilo objave. Za ločitev posameznih sporočil spojenih revizij se uporabi prednastavljeni niz. Če želite uporabiti svoj niz, ustvarite na naslovu `HKCU\Software\TortoiseSVN\MergeLogSeparator` ključ `SZ`, kot vrednost pa vpišite izbrani niz.

Vedno pokaži krivdne informacije z uporabo programa TortoiseMerge

TortoiseSVN omogoča uporabo zunanjega pregledovalnika razlik. Večina pregledovalnikov pa ni primerna za prikaz krivdnih informacij (**Razdelek 4.23.2, "Okriži spremembe"**), zato v tem primeru raje uporabite TortoiseMerge. Da bi to storili, ustvarite ključ `DWORD` z vrednostjo `1` na lokaciji `HKCU\Software\TortoiseSVN\DiffBlamesWithTortoiseMerge`.

Current revision highlighting for folders in log dialog

The log dialog highlights the current working copy revision when the log is shown for a file. To do the same thing for a folder requires a working copy crawl, which is the default action, but it can be a slow operation for large working copies. If you want to change the operation of this feature you must create a `DWORD` registry key at `HKCU\Software\TortoiseSVN\RecursiveLogRev`. A value of `0` disables the feature (no highlighting for folders), a value of `1` (default) will fetch the status recursively (find the highest revision in the working copy tree), and a value of `2` will check the revision of the selected folder itself, but will not check any child items.

Make checkout fail if an item of the same name exists

By default, if you checkout a working copy over an existing unversioned folder structure, as you might do after import, then any existing which differ from the repository content will be left unchanged and marked as modified. When you come to commit, it is your local copy which will then be sent back to the repository. Some people would prefer the checkout to fail if the existing content differs, so that if two people add the same file the second person's version does not overwrite the original version by mistake. If you want to force checkouts to fail in this instance you must create a DWORD registry key with value 0 at HKCU\Software\TortoiseSVN\AllowUnversionedObstruction.

4.30.11. Delovne mape Subversion

Ko VS.NET 2003 uporabljate za spletne projekte, ne more delati z mapami `.svn`, ki jih Subversion uporablja za shranjevanje notranjih informacij. To ni napaka sistema Subversion. To je napaka razvojnega okolja VS.NET 2003 in dodatkov za FrontPage, ki jih uporablja.

Napaka je odpravljena v VS2005 in novejših različicah.

Od različice 1.3.0 sistemov Subversion in TortoiseSVN naprej lahko nastavite okoljsko spremenljivko `SVN_ASP_DOT_NET_HACK`. Če je le-ta nastavljena, bo Subversion namesto map `.svn` uporabljal mape `_svn`. Da bi se sprememba okoljske spremenljivke upoštevala, morate ponovno zagnati lupino. Običajno to pomeni ponoven zagon računalnika. Sedaj lahko to nastavitve opravite v oknu za nastavitve z uporabo preprostega potrditvenega polja - poglejte na [Razdelek 4.30.1, "Splošne nastavitve"](#).

Za več informacij in ostale načine, kako se tem težavam izogniti, preberite članek o tem v [pogosto zastavljenih vprašanjih](#) [<http://tortoisesvn.net/aspdotnethack>].

4.31. Zadnji korak

Darujte!

Čeprav sta programa TortoiseSVN in TortoiseMerge brezplačna, lahko pomagate razvijalcem s pošiljanjem popravkov in igranjem aktivne vloge pri razvoju. Lahko nas tudi pomagate razvedriti med nešteti urami, ki jih prebijemo pred računalnikom.

Med delom na projektu TortoiseSVN radi poslušamo glasbo. In ker delamo na projektu veliko časa, potrebujemo *veliko* glasbe. Zato smo sestavili seznam želja z našimi priljubljenimi CDji in DVDji: <http://tortoisesvn.tigris.org/donate.html>. Prosimo, poglejte tudi seznam ljudi, ki so prispevali k projektu s popravki ali prevodi.

Poglavje 5. Program SubWCRev

SubWCRev je konzolna aplikacija, ki se uporablja za pridobivanje informacij o stanju delovne kopije z možnostjo zamenjave ključnih besed v predlogi. Postopek se pogosto uporablja kot del procesa gradnje projekta. S tem se v elemente projekta, ki se gradi, vnese informacije o delovni kopiji. Tipičen primer uporabe je prikaz številke revizije v pogovornem oknu "O programu".

5.1. Program SubWCRev za ukazno vrstico

SubWCRev prebere stanje vseh datotek v delovni kopiji, razen datotek zunanjih projektov (po privzetih nastavitvah). Zapomni si najvišjo najdeno številko revizije in časovni žig te revizije. Prav tako si zapomni, ali so v delovni kopiji narejene krajevne spremembe ali datoteke mešanih revizij. Na standardnem izhodu (stdout) se izpiše številka revizije, območje posodobitve revizij in stanje sprememb.

SubWCRev.exe kličemo iz ukazne vrstice ali iz ukazne datoteke in nastavljam s pomočjo parametrov ukazne vrstice.

```
SubWCRev PotDelovneKopije [IzvirnaDatoteka CiljnaDatoteka] [-nmdfe]
```

PotDelovneKopije je pot do delovne kopije, ki se preverja. SubWCRev lahko uporabljate le na delovni kopiji, ne pa tudi neposredno na skladišču. Pot je lahko absolutna ali relativna na trenutno mapo.

Če želite, da SubWCRev opravi zamenjavo ključnih besed, tako da so polja revizija in naslov URL shranjena v besedilni datoteki, morate podati predlogo IzvirnaDatoteka in izhodno datoteko CiljnaDatoteka, ki vsebuje zamenjano verzijo predloge.

Obstaja več opsijskih stikal, ki vplivajo na to, kako SubWCRev deluje. Če uporabite več kot eno, jih morate navesti kot eno skupino, n. pr. -nm, in ne -n -m.

Preklop	Opis
-n	Ob uporabi tega stikala se SubWCRev zaključi s številko napake <code>ERRORLEVEL 7</code> , če delovna kopija vsebuje krajevne spremembe. To stikalo se uporablja, da se prepreči gradnja projekta, ki vsebuje neobjavljene spremembe.
-m	Ob uporabi tega stikala se SubWCRev zaključi s številko napake <code>ERRORLEVEL 8</code> , če delovna kopija vsebuje mešane revizije. To stikalo se uporablja, da se prepreči gradnja projekta, ki vsebuje delno posodobljeno delovno kopijo.
-d	Ob uporabi tega stikala se SubWCRev zaključi s številko napake <code>ERRORLEVEL 9</code> , če ciljna datoteka že obstaja.
-f	Ob uporabi tega stikala SubWCRev pri iskanju najvišje revizije vključi tudi mape. Po privzetih nastavitvah se išče najvišja revizija le na datotekah.
-e	Ob uporabi tega stikala SubWCRev pregleda mape, ki so označene z lastnostjo <code>svn:externals</code> , vendar le, če se nahajajo v istem skladišču. Privzeta nastavitve ne upošteva zunanjih datotek.
-x	Ob uporabi tega stikala SubWCRev izpiše številke revizij v šestnajstiški obliki
-X	Ob uporabi tega stikala SubWCRev izpiše številke revizij v šestnajstiški obliki s predpono '0X'.

Tabela 5.1. Seznam stikal ukazne vrstice, ki so na voljo

5.2. Zamenjava ključnih besed

Če podamo začetno in ciljno datoteko, SubWCRev prekopira začetno datoteko v ciljno, pri tem pa izvede zamenjavo ključnih besed:

Ključna beseda	Opis
\$WCREV\$	Zamenjano z najvišjo revizijo v delovni kopiji.
\$WCDATES\$	Zamenjano z datumom/uro objave najvišje revizije. Po privzetih nastavitvah je uporabljena mednarodna oblika: yyyy-mm-dd hh:mm:ss. Lahko pa uporabite lastno obliko s <code>strftime()</code> , n. pr. : <code>\$WCDATE=%a %b %d %I:%M:%S %p\$</code> . Za seznam znakov za oblikovanje pogledajte na spletne navodila [http://www.cppreference.com/stddate/strftime.html].
\$WCNOW\$	Replaced with the current system date/time. This can be used to indicate the build time. Time formatting can be used as described for <code>\$WCDATE\$</code> .
\$WCRANGES\$	Zamenjano z območjem revizij delovne kopije. Če je delovna kopija v konsistentnem stanju, bo to le ena številka. Če delovna kopija vsebuje mešane revizije - ker je zastarela ali zaradi namerne uporabe ukaza posodobi-na-revizijo - se območje prikaže v obliki 100:200.
\$WCMIXED\$	<code>\$WCMIXED?TText:FText\$</code> se zamenja z vrednostjo <code>TText</code> , če obstajajo mešane revizije, in z vrednostjo <code>FText</code> , če ne obstajajo.
\$WCMODS\$	<code>\$WCMODS?TText:FText\$</code> se zamenja z vrednostjo <code>TText</code> , če obstajajo krajevne spremembe, in z vrednostjo <code>FText</code> , če krajevnih sprememb ni.
\$WCURL\$	Zamenjano z naslovom URL delovne kopije, ki je bil podan programu SubWCRev.
\$WCINSVN\$	<code>\$WCINSVN?TText:FText\$</code> se zamenja z vrednostjo <code>TText</code> , če je element pod nadzorom, in z vrednostjo <code>FText</code> , če ni.
\$WCNEEDSLOCKS\$	<code>\$WCNEEDSLOCK?TText:FText\$</code> se zamenja z vrednostjo <code>TText</code> , če ima element nastavljeno lastnost <code>svn:needs-lock</code> , in z vrednostjo <code>FText</code> , če je nima.
\$WCISLOCKED\$	<code>\$WCISLOCKED?TText:FText\$</code> se zamenja z vrednostjo <code>TText</code> , če je element zaklenjen, in z vrednostjo <code>FText</code> , če ni.
\$WCLOCKDATES\$	Replaced with the lock date. Time formatting can be used as described for <code>\$WCDATE\$</code> .
\$WCLOCKOWNER\$	Replaced with the name of the lock owner.
\$WCLOCKCOMMENTS\$	Replaced with the comment of the lock.

Tabela 5.2. Seznam stikal ukazne vrstice, ki so na voljo



Namig

Some of these keywords apply to single files rather than to an entire working copy, so it only makes sense to use these when SubWCRev is called to scan a single file. This applies to `$WCINSVN$`, `$WCNEEDSLOCK$`, `$WCISLOCKED$`, `$WCLOCKDATE$`, `$WCLOCKOWNER$` and `$WCLOCKCOMMENT$`.

5.3. Primer uprabe ključne besede

Spodnji primer prikazuje, kako se ključne besede v predlogi zamenjajo v izhodni datoteki.

```
// Preizkusna datoteka za SubWCRev: testfile.tmpl

char *Revision = "$WCREV$";
```



```
char *Modified = "$WCMODS?Spremenjeno:Nespremenjeno$";
char *Date     = "$WCDATE$";
char *RevRange = "$WCRANGE$";
char *Mixed    = "$WCMIXED?Mešana delovna kopija:Nemešana$";
char *URL      = "$WCURL$";
```

```
#if $WCMODS?1:0$
#error Vir spremenjen
#endif
```

```
// EndOfFile
```

Po zagonu programa SubWCRev.exe pot\do\delovnekopije testfile.tmp1 testfile.txt bo izhodna datoteka testfile.txt izgledala takole:

```
// Test file for SubWCRev: testfile.txt
```

```
char *Revision = "3701";
char *Modified = "Modified";
char *Date     = "2005/06/15 11:15:12";
char *Range    = "3699:3701";
char *Mixed    = "Mixed revision WC";
char *URL      = "http://project.domain.org/svn/trunk/src";
```

```
#if 1
#error Source is modified
#endif
```

```
// End of file
```



Namig

Takšna datoteka je vključena v proces gradnje, zato bi pričakovali, da bo pod nadzorom. Pod nadzor dajte predlogo datoteke in ne samodejno ustvarjene datoteke, v nasprotnem primeru boste morali ob vsakem usvarjanju datoteke narediti objavo, kar pa bi pomenilo, da je potrebno datoteko spet posodobiti.

5.4. Vmesnik COM

Če morate do informacij o reviziji dostopati iz drugih programov, lahko uporabite vmesnik COM programa SubWCRev. Ustvariti morate objekt `SubWCRev.object`, na katerem so podprte naslednje metode:

Način	Opis
.GetWCInfo	Metoda pregleda delovno kopijo in pridobi informacije o revizijah. Klicati jo morate pred metodami, ki so opisane v nadaljevanju. Prvi parameter je pot. Drugi parameter mora imeti vrednost true, če želite upoštevati tudi revizije map. Ta parameter je ekvivalenten stikalu <code>-f</code> v ukazni vrstici. Tretji parameter mora biti true, če želite vključiti tudi zunanje elemente. Ta parameter je ekvivalenten stikalu <code>-e</code> v ukazni vrstici.
.Revision	Najvišja objavljena revizija v delovni kopiji. Isto kot spremenljivka <code>\$WCREV\$</code> .
.Date	Datum/čas objave najvišje revizije. Isto kot spremenljivka <code>\$WCDATE\$</code> .

Način	Opis
.Author	Avtor najvišje objavljene revizije, to je zadnji avtor, ki je objavil spremembe delovne kopije.
.MinRev	Najnižja revizija posodobitve, kot je prikazana z \$WCRANGE\$
.MaxRev	Najvišja revizija posodobitve, kot je prikazana z \$WCRANGE\$
.HasModifications	'True', če obstajajo krajevne spremembe
.Url	Zamenjano z naslovom URL skladišča poti delovne kopije, uporabljene pri GetWCInfo. Isto kot spremenljivka \$WCURL\$
.IsSvnItem	True if the item is versioned.
.NeedsLocking	True if the item has the svn:needs-lock property set.
.IsLocked	True if the item is locked.
.LockCreationDate	String representing the date when the lock was created, or an empty string if the item is not locked.
.LockOwner	String representing the lock owner, or an empty string if the item is not locked.
.LockComment	The message entered when the lock was created.

Tabela 5.3. Podprte COM/avtomatizacijske metode

Naslednji primer prikazuje uporabo vmesnika.

```
// testCOM.js - javascript file
// test script for the SubWCRev COM/Automation-object

filesystem = new ActiveXObject("Scripting.FileSystemObject");

revObject1 = new ActiveXObject("SubWCRev.object");
revObject2 = new ActiveXObject("SubWCRev.object");
revObject3 = new ActiveXObject("SubWCRev.object");
revObject4 = new ActiveXObject("SubWCRev.object");

revObject1.GetWCInfo(
    filesystem.GetAbsolutePathName("."), 1, 1);
revObject2.GetWCInfo(
    filesystem.GetAbsolutePathName(".."), 1, 1);
revObject3.GetWCInfo(
    filesystem.GetAbsolutePathName("SubWCRev.cpp"), 1, 1);
revObject4.GetWCInfo(
    filesystem.GetAbsolutePathName("../.."), 1, 1);

wcInfoString1 = "Revision = " + revObject1.Revision +
    "\nMin Revision = " + revObject1.MinRev +
    "\nMax Revision = " + revObject1.MaxRev +
    "\nDate = " + revObject1.Date +
    "\nURL = " + revObject1.Url + "\nAuthor = " +
    revObject1.Author + "\nHasMods = " +
    revObject1.HasModifications + "\nIsSvnItem = " +
    revObject1.IsSvnItem + "\nNeedsLocking = " +
    revObject1.NeedsLocking + "\nIsLocked = " +
    revObject1.IsLocked + "\nLockCreationDate = " +
    revObject1.LockCreationDate + "\nLockOwner = " +
    revObject1.LockOwner + "\nLockComment = " +
```

```
    revObject1.LockComment;
wcInfoString2 = "Revision = " + revObject2.Revision +
  "\nMin Revision = " + revObject2.MinRev +
  "\nMax Revision = " + revObject2.MaxRev +
  "\nDate = " + revObject2.Date +
  "\nURL = " + revObject2.Url + "\nAuthor = " +
  revObject2.Author + "\nHasMods = " +
  revObject2.HasModifications + "\nIsSvnItem = " +
  revObject2.IsSvnItem + "\nNeedsLocking = " +
  revObject2.NeedsLocking + "\nIsLocked = " +
  revObject2.IsLocked + "\nLockCreationDate = " +
  revObject2.LockCreationDate + "\nLockOwner = " +
  revObject2.LockOwner + "\nLockComment = " +
  revObject2.LockComment;
wcInfoString3 = "Revision = " + revObject3.Revision +
  "\nMin Revision = " + revObject3.MinRev +
  "\nMax Revision = " + revObject3.MaxRev +
  "\nDate = " + revObject3.Date +
  "\nURL = " + revObject3.Url + "\nAuthor = " +
  revObject3.Author + "\nHasMods = " +
  revObject3.HasModifications + "\nIsSvnItem = " +
  revObject3.IsSvnItem + "\nNeedsLocking = " +
  revObject3.NeedsLocking + "\nIsLocked = " +
  revObject3.IsLocked + "\nLockCreationDate = " +
  revObject3.LockCreationDate + "\nLockOwner = " +
  revObject3.LockOwner + "\nLockComment = " +
  revObject3.LockComment;
wcInfoString4 = "Revision = " + revObject4.Revision +
  "\nMin Revision = " + revObject4.MinRev +
  "\nMax Revision = " + revObject4.MaxRev +
  "\nDate = " + revObject4.Date +
  "\nURL = " + revObject4.Url + "\nAuthor = " +
  revObject4.Author + "\nHasMods = " +
  revObject4.HasModifications + "\nIsSvnItem = " +
  revObject4.IsSvnItem + "\nNeedsLocking = " +
  revObject4.NeedsLocking + "\nIsLocked = " +
  revObject4.IsLocked + "\nLockCreationDate = " +
  revObject4.LockCreationDate + "\nLockOwner = " +
  revObject4.LockOwner + "\nLockComment = " +
  revObject4.LockComment;

WScript.Echo(wcInfoString1);
WScript.Echo(wcInfoString2);
WScript.Echo(wcInfoString3);
WScript.Echo(wcInfoString4);
```

Poglavje 6. IBugtraqProvider interface

To get a tighter integration with issue trackers than by simply using the `bugtraq:` properties, TortoiseSVN can make use of COM plugins. With such plugins it is possible to fetch information directly from the issue tracker, interact with the user and provide information back to TortoiseSVN about open issues, verify log messages entered by the user and even run actions after a successful commit to e.g. close an 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. ([Razdelek 3, "TortoiseSVN je brezplačen!"](#) explains how to access the repository).

6.1. The IBugtraqProvider interface

TortoiseSVN 1.5 can use plugins which implement the IBugtraqProvider interface. The interface provides a few methods which plugins can use to interact with the issue tracker.

```
HRESULT ValidateParameters (
    // Parent window for any UI that needs to be
    // displayed during validation.
    [in] HWND hParentWnd,

    // The parameter string that needs to be validated.
    [in] BSTR parameters,

    // Is the string valid?
    [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 (
    // Parent window for any (error) UI that needs to be displayed.
    [in] HWND hParentWnd,

    // The parameter string, just in case you need to talk to your
    // web service (e.g.) to find out what the correct text is.
    [in] BSTR parameters,

    // What text do you want to display?
    // Use the current 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 (
    // Parent window for your provider's UI.
    [in] HWND hParentWnd,

    // Parameters for your provider.
    [in] BSTR parameters,
    [in] BSTR commonRoot,
    [in] SAFEARRAY(BSTR) pathList,

    // The text already present in the commit message.
    // Your provider should include this text in the new message,
    // where appropriate.
    [in] BSTR originalMessage,

    // The new text for the commit message.
    // This replaces the original message.
    [out, retval] BSTR *newMessage
);

```

This is the main method of the plugin. This method is called from the TortoiseSVN commit dialog when the user clicks on the plugin button. The `parameters` string is the string the user has to enter in the settings dialog when he configures the plugin. Usually a plugin would use this to find the URL of the issue tracker and/or login information or more. The `commonRoot` string contains the parent path of all items selected to bring up the commit dialog. Note that this is *not* the root path of all items which the user has selected in the commit dialog. The `pathList` parameter contains an array of paths (as strings) which the user has selected for the commit. The `originalMessage` parameter contains the text entered in the log message box in the commit dialog. If the user has not yet entered any text, this string will be empty. The `newMessage` return string is copied into the log message edit box in the commit dialog, replacing whatever is already there. If a plugin does not modify the `originalMessage` string, it must return the same string again here, otherwise any text the user has entered will be lost.

6.2. The IBugtraqProvider2 interface

In TortoiseSVN 1.6 a new interface was added which provides more functionality for plugins. This `IBugtraqProvider2` interface inherits from `IBugtraqProvider`.

```

HRESULT GetCommitMessage2 (
    // Parent window for your provider's UI.
    [in] HWND hParentWnd,

    // Parameters for your provider.
    [in] BSTR parameters,
    // The common URL of the commit
    [in] BSTR commonURL,
    [in] BSTR commonRoot,
    [in] SAFEARRAY(BSTR) pathList,

    // The text already present in the commit message.
    // Your provider should include this text in the new message,
    // where appropriate.
    [in] BSTR originalMessage,

    // You can assign custom revision properties to a commit
    // by setting the next two params.

```

```

// note: Both safearrays must be of the same length.
//       For every property name there must be a property value!

// The content of the bugID field (if shown)
[in] BSTR bugID,

// Modified content of the bugID field
[out] BSTR * bugIDOut,

// The list of revision property names.
[out] SAFEARRAY(BSTR) * revPropNames,

// The list of revision property values.
[out] SAFEARRAY(BSTR) * revPropValues,

// The new text for the commit message.
// This replaces the original message
[out, retval] BSTR * newMessage
);

```

This method is called from the TortoiseSVN commit dialog when the user clicks on the plugin button. This method is called instead of `GetCommitMessage()`. Please refer to the documentation for `GetCommitMessage` for the parameters that are also used there. The parameter `commonURL` is the parent URL of all items selected to bring up the commit dialog. This is basically the URL of the `commonRoot` path. The parameter `bugID` contains the content of the bug-ID field (if it is shown, configured with the property `bugtraq:message`). The return parameter `bugIDOut` is used to fill the bug-ID field when the method returns. The `revPropNames` and `revPropValues` return parameters can contain name/value pairs for revision properties that the commit should set. A plugin must make sure that both arrays have the same size on return! Each property name in `revPropNames` must also have a corresponding value in `revPropValues`. If no revision properties are to be set, the plugin must return empty arrays.

```

HRESULT CheckCommit (
    [in] HWND hParentWnd,
    [in] BSTR parameters,
    [in] BSTR commonURL,
    [in] BSTR commonRoot,
    [in] SAFEARRAY(BSTR) pathList,
    [in] BSTR commitMessage,
    [out, retval] BSTR * errorMessage
);

```

This method is called right before the commit dialog is closed and the commit begins. A plugin can use this method to validate the selected files/folders for the commit and/or the commit message entered by the user. The parameters are the same as for `GetCommitMessage2()`, with the difference that `commonURL` is now the common URL of all *checked* items, and `commonRoot` the root path of all checked items. The return parameter `errorMessage` must either contain an error message which TortoiseSVN shows to the user or be empty for the commit to start. If an error message is returned, TortoiseSVN shows the error string in a dialog and keeps the commit dialog open so the user can correct whatever is wrong. A plugin should therefore return an error string which informs the user *what* is wrong and how to correct it.

```

HRESULT OnCommitFinished (
    // Parent window for any (error) UI that needs to be displayed.
    [in] HWND hParentWnd,

    // The common root of all paths that got committed.
    [in] BSTR commonRoot,

```

```
// All the paths that got committed.
[in] SAFEARRAY(BSTR) pathList,

// The text already present in the commit message.
[in] BSTR logMessage,

// The revision of the commit.
[in] ULONG revision,

// An error to show to the user if this function
// returns something else than 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(
    // Whether the provider provides options
    [out, retval] VARIANT_BOOL *ret
);
```

This method is called from the settings dialog where the user can configure the plugins. If a plugin provides its own configuration dialog with `ShowOptionsDialog`, it must return `TRUE` here, otherwise it must return `FALSE`.

```
HRESULT ShowOptionsDialog(
    // Parent window for the options dialog
    [in] HWND hParentWnd,

    // Parameters for your provider.
    [in] BSTR parameters,

    // The parameters string
    [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.

Dodatek A. Pogosto zastavljena vprašanja (FAQ)

Ker se TortoiseSVN nenehno razvija, je včasih težko vzdrževati dokumentacijo, tako da je ta vedno v skladu z aplikacijo. Vzdržujemo *spletno stran z najpogosteje zastavljenimi vprašanji* [<http://tortoisesvn.tigris.org/faq.html>], ki vsebujejo izbor vprašanj, ki se najpogosteje pojavljajo na dopisnih seznamih projekta TortoiseSVN <dev@tortoisesvn.tigris.org> in <users@tortoisesvn.tigris.org>.

Vzdržujemo tudi *sledilnik zadev* [<http://issues.tortoisesvn.net>], kjer lahko izveste, kaj nameravamo dodati aplikaciji v prihodnje in kateri hrošči so že odpravljeni. Če ste v programu našli napako ali želite predlagati izboljšavo, najprej preverite, če tega ni storil že kdo pred vami.

Če odgovora na svoje vprašanje ne najdete nikjer drugje, uporabite naše dopisne sezname. Naslov <users@tortoisesvn.tigris.org> uporabite v primeru vprašanje glede uporabe programa TortoiseSVN. Če želite sodelovati pri razvoju TortoiseSVN, lahko sodelujete v debatah na naslovu <dev@tortoisesvn.tigris.org>.

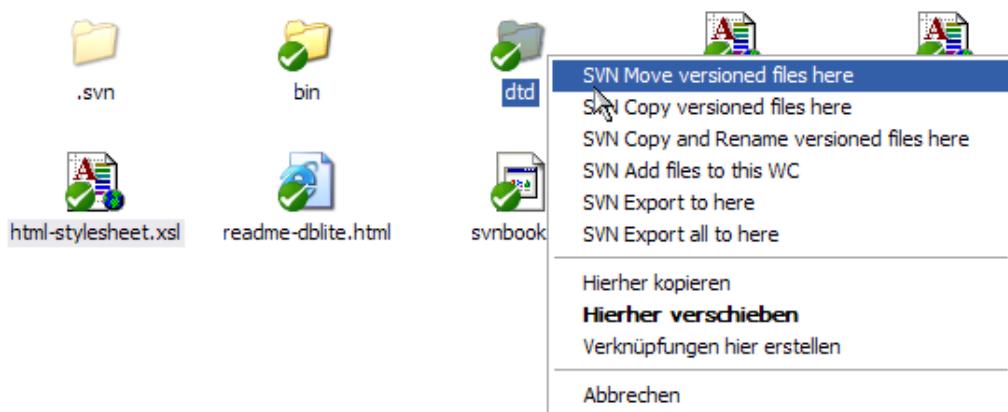
Dodatek B. Kako naredim...

Ta dodatek vsebuje rešitve težav/vprašanj, ki se lahko pojavijo pri uporabi TortoiseSVN.

B.1. Kako premaknem/kopiram večje število datotek naenkrat

Premikanje/kopiranje posameznih datotek se izvaja z ukazom TortoiseSVN → Preimenuj.... Če premikate/kopirate večje število datotek, je ta postopek počasen in zahteva preveč dela.

Priporočamo, da datoteke povlečete z desnim gumbom na novo lokacijo. Enostavno kliknite z desnim gumbom na datoteke, ki jih želite premakniti/prekopirati, ne da spustite miškin gumb. Potem datoteke povlecite na novo lokacijo in spustite miškin gumb. Prikazal se bo kontekstni meni, kjer lahko izberete Kontekstni meni → SVN - Kopiraj datoteke pod nadzorom sem ali Kontekstni meni → SVN - Premakni datoteke pod nadzorom sem.



B.2. Kako prisilim uporabnika, da vnese sporočilo dnevniškega zapisa

Obstajata dva načina, kako uporabnikom preprečiti objavljanje s praznim sporočilom dnevniškega zapisa. Eden je poseben za TortoiseSVN, drugi pa je uporaben za vse odjemalce Subversion, vendar morate za nastavev imeti neposreden dostop do strežnika.

B.2.1. Ukazna datoteka akcije na strežniku

Če imate neposreden dostop do strežnika skladišča, lahko nastavite ukazno datoteko za akcijo pred objavo, ki zavrne vse objave s praznim sporočilom dnevniškega zapisa ali s prekratnim sporočilom.

V mapi skladišča na strežniku se nahaja podmapa z imenom `hooks`, ki vsebuje nekaj primerov ukaznih datotek akcij, ki jih lahko uporabite. Datoteka `pre-commit.tmpl` vsebuje primer, ki zavrne objavo brez sporočila dnevniškega zapisa ali s prekratnim sporočilom. Zraven so zapisana tudi navodila za namestitvev/uporabo ukazne datoteke. Sledite jim.

Ta metoda je priporočljiva, če uporabniki poleg TortoiseSVN uporabljajo še druge odjemalce za Subversion. Slaba stran metode je, da objavo zavrne strežnik, zato bodo uporabniki dobili sporočilo o napaki. Odjemalec namreč ne more vnaprej vedeti, da bo objava zavrnjena. Če želite, da TortoiseSVN onemogoči gumb **V redu**, dokler ni vneseno sporočilo dnevniškega zapisa predpisane dolžine, uporabite metodo, ki je opisana spodaj.

B.2.2. Lastnosti projekta

TortoiseSVN uporablja lastnosti, da kontrolira nekatere zmožnosti programa. Ena takšnih je lastnost `tsvn:logminsize`.

Če nastavite to lastnost na mapi, bo TortoiseSVN onemogočil gumb **V redu** v vseh oknih objave, dokler uporabnik ne bo vnesel sporočila dnevniškega zapisa predpisane dolžine.

Za podrobne informacije o projektnih lastnostih poglejte [Razdelek 4.17, "Nastavitve projekta"](#)

B.3. Kako posodobim izbrane datoteke iz skladišča

Običajno posodobite delovno kopijo z ukazom TortoiseSVN → Posodobi. Če pa želite posodobiti le nekaj novih datotek, ki jih je dodal vaš sodelavec, ne želite pa hkrati spajati ostalih datotek, uporabite drugačen pristop.

Uporabite TortoiseSVN → Preveri spremembe in kliknite na gumb **Preveri skladišče**. S tem vidite, kaj se je spremenilo v skladišču. Izberite datoteke, ki jih želite posodobiti v delovni kopiji, potem pa izvedite posodobitev preko ukazov kontekstnega menija.

B.4. Kako prevrtim nazaj revizije v skladišču

B.4.1. Uporabite okno za prikaz dnevniških zapisov

Najlažji način, kako povrniti spremembe iz ene revizije ali obsega revizij je z uporabo okna dnevnika. S to metodo lahko pozabite na zadnje spremembe in določite starejšo revizijo za novo revizijo HEAD.

1. Izberite datoteko ali mapo, kateri želite povrniti spremembe. Če želite povrniti vse spremembe, izberite vrhno datoteko.
2. Za seznam revizij izberite TortoiseSVN → Pokaži dnevnik. Mogoče boste morali uporabiti gumb **Pokaži vse** ali gumb **Naslednjih 100**, da bi videli revizijo(e), ki vas zanima.
3. Izberite revizije, ki jih želite povrniti. Če želite povrniti območje revizij, izberite prvo izmed njih, držite tipko **Shift** in izberite zadnjo izmed njih. Upoštevajte, da mora biti območje neprekinjeno, brez lukenj. Kliknite z desnim gumbom na izbrane revizije in izberite **Kontekstni meni** → **Povrni spremembe te revizije**.
4. Če želite staro revizijo postaviti za novo revizijo HEAD, kliknite z desnim gumbom na izbrano revizijo in izberite **Kontekstni meni** → **Povrni na to revizijo**. S tem boste izbrisali vse spremembe, narejene po izbrani reviziji.

Povrnili ste spremembe znotraj delovne kopije. Preverite rezultate in potem objavite spremembe.

B.4.2. Uporabite okno za spajanje

Da bi razveljavili spremembe večjega števila revizij, lahko uporabite okno za spajanje. Prejšnja metoda v ozadju uporablja spajanje; ta metoda ga uporablja direktno.

1. V delovni kopiji izberite TortoiseSVN → **Spoji**.
2. V polje **Iz**: vnesite naslov URL mape veje ali oznake, ki vsebuje spremembe, ki jih želite povrniti v delovni kopiji. Pojaviti se mora prednastavljeni naslov URL.
3. V polje **Iz revizije** vnesite trenutno številko revizije. Če ste prepričani, da sprememb ne objavlja nihče drug, lahko izberete revizijo HEAD.

4. Prepričajte se, da je polje **Uporabite URL "Od:"** potrjeno.
5. V polje **Do revizije** vnesite številko revizije, na katero želite povrniti delovno kopijo. To pomeni na revizijo *pred* prvo revizijo, ki vsebuje spremembe, ki jih želite povrniti.
6. Za zaključek spajanja kliknite na gumb **V redu**.

Povrnili ste spremembe znotraj delovne kopije. Preverite rezultate in potem objavite spremembe.

B.4.3. Uporabite `svndumpfilter`

Ker TortoiseSVN nikoli ne izgubi podatkov, revizije, ki ste jih povrnili, še vedno obstajajo v skladišču. Vse, kar ste naredili, je, da ste spremenili zadnjo (HEAD) revizijo na vsebino predhodnje revizije. Če želite, da določene revizije izginejo iz skladišča in da se za njimi izgubi vsaka sled, morate uporabiti bolj drastične ukrepe. Razen v primeru zelo dobrega razloga to *ni priporočljivo*. Eden od možnih razlogov za uporabo te metode je objava zaupnega dokumenta v javno skladišče.

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/1.5/svn.reposadmin.maint.html) [http://svnbook.red-bean.com/en/1.5/svn.reposadmin.maint.html].

B.5. Compare two revisions of a file or folder

If you want to compare two revisions in an item's history, for example revisions 100 and 200 of the same file, just use TortoiseSVN → **Show Log** to list the revision history for that file. Pick the two revisions you want to compare then use **Context Menu** → **Compare Revisions**.

If you want to compare the same item in two different trees, for example the trunk and a branch, you can use the repository browser to open up both trees, select the file in both places, then use **Context Menu** → **Compare Revisions**.

If you want to compare two trees to see what has changed, for example the trunk and a tagged release, you can use TortoiseSVN → **Revision Graph** Select the two nodes to compare, then use **Context Menu** → **Compare HEAD Revisions**. This will show a list of changed files, and you can then select individual files to view the changes in detail. You can also export a tree structure containing all the changed files, or simply a list of all changed files. Read [Razdelek 4.10.3, "Primerjanje map"](#) for more information. Alternatively use **Context Menu** → **Unified Diff of HEAD Revisions** to see a summary of all differences, with minimal context.

B.6. Kako vključim skupni podprojekt

Včasih želite vključiti v delovno kopijo še en projekt, mogoče izvorno kodo za knjižnico, ki jo uporabljate. Ne želite narediti dvojnika izvorne kode v svojem skladišču, saj s tem izgubite povezavo z izvorno (vzdrževano) kodo. Ali pa imate več projektov, ki si delijo skupno kodo. Obstajajo vsaj trije načini, kako se spopasti s tem.

B.6.1. Uporabite lastnost `svn:externals`

Set the `svn:externals` property for a folder in your project. This property consists of one or more lines; each line has the name of a sub-folder which you want to use as the checkout folder for common code, and the repository URL that you want to be checked out there. For full details refer to [Razdelek 4.18, "External Items"](#).

Objavite novo mapo. Pri posodobitvi bo Subversion potegnil kopijo projekta iz skladišča v vašo delovno kopijo. Če je to potrebno, se podmape ustvarijo samodejno. Ob vsaki posodobitvi delovne kopije boste posodobili tudi vse zunanje projekte.

Če se zunanji projekt nahaja v istem skladišču, bodo vse spremembe v njem tudi na seznamu za objavo, ko boste objavljali spremembe v glavnem projektu.

Če se zunanji projekt nahaja v drugem skladišču, bodo vse spremembe, ki jih naredite na zunanjem projektu, prikazane, ko boste objavili glavni projekt, vendar morate zunanje spremembe objaviti posebej.

Od vseh treh opisanih možnosti je ta edina, ki ne potrebuje namestitve na strani odjemalca. Ko so zunanje datoteke določene v lastnostih map, bodo ob posodobitvi vsi odjemalci popolnili mape.

B.6.2. Uporabite vgnezdno delovno kopijo

Ustvarite novo mapo znotraj svojega projekta, ki naj vsebuje skupno kodo, vendar je ne dodajte v sistem Subversion.

Na novi mapi izvedite TortoiseSVN → Prezem skupne kode. Sedaj imate ločeno delovno kopijo vgnezdno v svoji glavni delovni kopiji.

Ti dve delovni kopiji sta neodvisni. Ko objavite spremembe na korenski delovni kopiji, se spremembe na vgnezdni delovni kopij prezrejo. Podobno je v primeru posodobitve: ko posodobite korensko delovno kopijo, je vgnezdna prezrta.

B.6.3. Uporabite relativno lokacijo

Če na več projektih uporabljate skupno kodo in ne želite imeti več delovnih kopij te kode, jo lahko prevzamete v posebno mapo, nato pa naredite povezavo nanjo v vseh projektih. Primer:

```
C:\Projekti\Proj1
C:\Projekti\Proj2
C:\Projekti\Proj3
C:\Projekti\Skupno
```

in naredite povezavo na skupno kodo z uporabo relativne poti, n. pr. `..\..\Skupno\SkupnaKoda`.

Če so vaši projekti razmetani na nepovezanih lokacijah, lahko uporabite posebno variacijo zgornje rešitve. Skupno kodo shranite v svojo mapo, nato pa uporabite preslikavo pogonov in preslikate to lokacijo v nekaj, kar lahko uporabite v svojih projektih. Primer: prevzemite skupno kodo v `D:\Documents\framework` ali `C:\Documents and Settings\{login}\My Documents\framework` in potem uporabite

```
SUBST X: "D:\Documents\framework"
```

da ustvarite preslikavo pogona za uporabo v izvorni kodi. V kodi sedaj uporabljate absolutne lokacije.

```
#include "X:\superio\superio.h"
```

Ta metoda deluje le v okoljih, kjer vsi uporabniki uporabljajo računalnike PC. Preslikave pogonov boste morali dokumentirati, da bodo ostali člani teama vedeli, od kje prihajajo te skrivnostne datoteke. Metoda naj se uporablja v zaprtem krogu razvijalcev, za splošno uporabo pa ni priporočljiva.

B.7. Kako ustvarim bližnjico do skladišča

Če pogosto odpirate brskalnik po skladišču na določeni lokaciji, si lahko na namizju ustvarite bližnjico z uporabo avtomatizacijskega vmesnika do TortoiseProc. Ustvarite novo bližnjico in nastavite cilj na:

```
TortoiseProc.exe /command:repobrowser /path:"url/do/skladišča"
```

Seveda morate podati pravi naslov URL skladišča.

B.8. Kako dodam na seznam prezrtih datoteke, ki so že pod nadzorom

Če ste pomotoma dodali nekatere datoteke, ki bi jih morali preskočiti, kako jih dobite iz sistema nadzora, ne da bi jih izgubili? Mogoče imate svojo konfiguracijsko datoteko za razvojno okolje, ki ni del projekta, ste pa porabili kar nekaj časa, da ste si okolje nastavili po svojih željah.

Če dodane datoteke še niste objavili, lahko uporabite ukaz TortoiseSVN → Povrni..., s čimer prekličete dodano datoteko. Nato datoteko(e) dodate na seznam prezrtih elementov, da je ne bi naslednjič pomotoma spet dodali.

Če so datoteke že v skladišču, imate malce več dela.

1. Pritisnite tipko **Shift**, da dobite razširjeni kontekstni meni, in uporabite ukaz TortoiseSVN → Izbriši (ohrani krajevno). S tem označite datoteko za brisanje iz skladišča, vendar se pri tem ohrani krajevna kopija datoteke.
2. V delovni kopiji izberite TortoiseSVN → Objavi.
3. Dodajte datoteko/mapo na seznam prezrtih, da drugič ne naletite na enake težave.

B.9. Odstranjevanje delovne kopije iz nadzora različic

Če imate delovno kopijo, ki jo želite pretvoriti v navadno drevesno strukturo brez podmap `.svn`, jo lahko izvozite samo vase. Več o tem pove [Razdelek 4.26.1, "Kako odstranim delovno kopijo iz nadzora različic"](#).

B.10. Kako odstranim delovno kopijo

Če imate delovno kopijo, ki je ne potrebujete več, kako se je znebite? Enostavno - izbrišite jo iz Raziskovalca! Delovne kopije so zasebne krajevne enote.

Dodatek C. Uporabni namigi za skrbnike sistema

Ta dodatek podaja rešitve problemov/vprašanj, ki se lahko pojavijo skrbnikom, ki so zadolženi za namestitvev TortoiseSVN na večje število računalnikov.

C.1. Namestitev TortoiseSVN preko pravic skupin

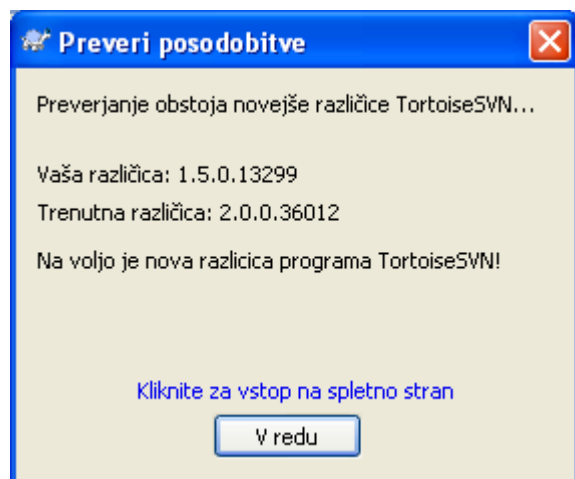
Namestitveni program TortoiseSVN je oblike datoteke MSI, kar pomeni, da ne bi smeli imeti nobenih težav pri dodajanju namestitve v pravice skupin (group policies) domenskega kontrolerja.

Dober članek na to temo se nahaja v bazi znanja podjetja Microsoft pod številko 314934: <http://support.microsoft.com/?kbid=314934>.

Različice TortoiseSVN 1.3.0 in novejšje je potrebno namestiti pod *Računalniško konfiguracijo (Computer Configuration)* in ne pod *Uporabniško konfiguracijo (User Configuration)*. Razlog je v tem, da potrebujejo nove različice nove knjižnice CRT in MFC DLL, ki se lahko namestijo le za računalnik in ne za uporabnika. Če morate TortoiseSVN res namestiti za uporabnika, potem morate najprej namestiti paket MFC in CRT verzije 8 na vsak računalnik, kjer želite namestitvev TortoiseSVN za uporabnika.

C.2. Preusmerjanje iskanja najnovejše različice

TortoiseSVN vsakih nekaj dni preveri, če obstaja novejša različica aplikacije. Če novejša različica obstaja, se pojavi okno, ki o tem obvesti uporabnika.



Slika C.1. Okno za nadgradnjo

Če ste odgovorni za veliko število uporabnikov v vaši domeni, verjetno želite, da vaši uporabniki uporabljajo le verzije, ki ste jih odobrili, in da si ne namestijo vedno zadnje verzije. Verjetno ne želite, da se pojavi pogovorno okno za nadgradnjo, da ne bi uporabniki sami nadredili nadgradnje.

TortoiseSVN različice 1.4.0 in novejšje omogoča, da preusmerite preverjanje na svoj intranetni strežnik. Nastavite ključ v registru `HKCU\Software\TortoiseSVN\UpdateCheckURL` (niz) na naslov URL, ki kaže na besedilno datoteko v intranetu. Besedilna datoteka mora imeti naslednjo obliko:

```
1.4.1.6000
```

```
Na voljo je nova različica programa TortoiseSVN!
```

`http://192.168.2.1/downloads/TortoiseSVN-1.4.1.6000-svn-1.4.0.msi`

V prvi vrstici se nahaja različica TortoiseSVN. Zagotoviti morate, da ustreza namestitveni datoteki. V drugi vrstici se nahaja besedilo, ki se prikaže v oknu z obvestilom o novi različici. Vpišete lahko karkoli želite, upoštevajte le, da je prostor omejen. V primeru predolgega besedila bo le-to skrajšano. V tretji vrstici se nahaja naslov URL namestitvenega paketa. Ta naslov se odpre, ko uporabnik klikne na sporočilo v pogovornem oknu. Uporabnika lahko pošljete tudi na spletno stran namesto neposredno na namestitveni paket. Naslov se odpre v privzetem brskalniku in če nastavite spletno stran, se bo odprla le-ta in prikazala uporabniku. Če nastavite namestitveni paket MSI, bo brskalnik uporabnika pozval, naj ga shrani na krajevnem disku.

C.3. Nastavljanje okoljske spremenljivke SVN_ASP_DOT_NET_HACK

Od verzije 1.4.0. naprej namestitveni program TortoiseSVN uporabniku ne ponudi več možnosti uporabe okoljske spremenljivke `SVN_ASP_DOT_NET_HACK`. Ta možnost je povzročila mnogo težav in zmed pri uporabnikih, ki vedno namestijo *prav vse*, ne glede na to, ali vedo, zakaj se določena zmožnost uporablja ali ne.

Ta možnost še obstaja, vendar je uporabniku skrita. Še vedno pa lahko prisilite namestitveni program TortoiseSVN, da nastavi okoljsko spremenljivko z nastavitvijo lastnosti `ASPDOTNETHACK` na `TRUE`. Primer: namestitveni program lahko poženetate takole:

```
msiexec /i TortoiseSVN-1.4.0.msi ASPDOTNETHACK=TRUE
```

C.4. Onemogočanje kontekstnega menija

Od različice 1.5.0 naprej TortoiseSVN ponuja onemogočanje (pravzaprav skrivanje) elementov v kontekstnem meniju. Ker je to zmožnost, ki se uporablja le z dobrim razlogom, uporabniškega vmesnika ni na voljo, zato je treba nastavitve opraviti neposredno v registru. Uporablja se za onemogočanje določenih ukazov, ki naj jih uporabnik ne bi uporabljal. Vendar upoštevajte, da s tem skrijete le elemente v kontekstnem meniju *Raziskovalca* in da so ukazi še vedno na razpolago na druge načine, n. pr. preko odjemalca za ukazno vrstico ali celo iz drugih pogovornih oken programa TortoiseSVN samega!

Gljuča registra, ki vsebujejo informacije o elementih kontekstnega menija, ki jih je potrebno prikazati, sta `HKEY_CURRENT_USER\Software\TortoiseSVN\ContextMenuEntriesMaskLow` in `HKEY_CURRENT_USER\Software\TortoiseSVN\ContextMenuEntriesMaskHigh`.

Vsak izmed teh vnosov v registru je vrednost `DWORD`, pri kateri vsak bit ustreza določenemu elementu v kontekstnem meniju. Nastavljen bit pomeni, da je ta element v meniju onemogočen.

Vrednost	Vnos v meniju
0x0000000000000001	Prezemi
0x0000000000000002	Posodobi
0x0000000000000004	Objavi
0x0000000000000008	Dodaj
0x0000000000000010	Povrni
0x0000000000000020	Čiščenje
0x0000000000000040	Reši
0x0000000000000080	Preklop
0x0000000000000100	Uvoz

Vrednost	Vnos v meniju
0x0000000000000200	Izvozi
0x0000000000000400	Ustvari skladišče tu
0x0000000000000800	Veja/Oznaka
0x0000000000001000	Spoji
0x0000000000002000	Izbriši
0x0000000000004000	Preimenuj
0x0000000000008000	Posodobi na revizijo
0x0000000000010000	Razlikuj
0x0000000000020000	Pokaži dnevnik
0x0000000000040000	Uredi spore
0x0000000000080000	Premakni
0x0000000000100000	Preveri spremembe
0x0000000000200000	Prezri
0x0000000000400000	Brskalnik po skladišču
0x0000000000800000	Okrivi
0x0000000001000000	Ustvari popravek
0x0000000002000000	Namesti popravek
0x0000000004000000	Revision graph
0x0000000008000000	Zaklep
0x0000000010000000	Odstrani zaklep
0x0000000020000000	Lastnosti
0x0000000040000000	Razlikuj z URL
0x0000000080000000	Izbriši datoteke brez različic
0x2000000000000000	Nastavitve
0x4000000000000000	Pomoč
0x8000000000000000	O programu

Tabela C.1. Elementi menija in njihove vrednosti

Primer: če želite onemogočiti elemente menija “Premesti”, “Izbriši elemente brez različic” in “Nastavitve”, seštejte vrednosti, dodeljene elementom:

$$\begin{aligned}
 & 0x0000000000008000 \\
 + & 0x0000000080000000 \\
 + & 0x2000000000000000 \\
 = & 0x2000000080080000
 \end{aligned}$$

Spodnjo vrednost DWORD (0x80080000) je potrebno shraniti v HKEY_CURRENT_USER\Software\TortoiseSVN\ContextMenuEntriesMaskLow, zgornjo vrednost DWORD (0x20000000) pa v HKEY_CURRENT_USER\Software\TortoiseSVN\ContextMenuEntriesMaskHigh.

Za ponoven prikaz teh elementov izbrišite zgornja vnosa v registru.

Dodatek D. Avtomatizacija TortoiseSVN

Ker je vse ukaze TortoiseSVN možno krmiliti preko parametrov ukazne vrstice, lahko za avtomatizacijo uporabite paketno datoteko ali pa ukaze in pogovorna okna poganjate z določeni parametri znotraj drugih programov (n. pr. iz vašega priljubljenega urejevalnika besedil).



Pomembno

Upoštevajte, da je TortoiseSVN odjemalec z grafičnim uporabniškim vmesnikom, to poglavje pa vam pove, kako lahko prikažete okna programa TortoiseSVN, tako da od uporabnika pridobite informacije. Če želite napisati ukazno datoteko, ki ne potrebuje vnosa s strani uporabnika, namesto TortoiseSVN uporabite uradni odjemalec Subversion za ukazno vrstico.

D.1. Ukazi TortoiseSVN

Program TortoiseSVN se imenuje `TortoiseProc.exe`. Vse ukaze kličemo s parametrom `/command:abcd`, kjer je `abcd` ime zahtevanega ukaza. Večina ukazov potrebuje vsaj eno pot kot parameter, ki ga podamo z `/path:"neka\pot"`. V spodnji tabeli ukaz ustreza `/command:abcd`, pot pa parametru `/path:"neka\pot"`.

Ker nekateri ukazi sprejmejo seznam ciljnih poti (n. pr. ko objavljate več datotek), lahko podamo parametru `/path` več poti, ki jih ločimo z znakom `*`.

TortoiseSVN za pošiljanje več parametrov med lupino in glavnim programom uporablja začasne datoteke. Od verzije 1.5.0 naprej je parameter `/notempfile` zastarel in ga ni potrebno več podajati.

Pogovorno okno, ki se uporablja pri objavah, posodobitvah in ostalih ukazih običajno ostane odprto dokler ne pritisnemo na gumb **V redu**. To obnašanje lahko spremenimo z nastavitvami aplikacije. Vendar se bo pogovorno okno zaprlo v vsakem primeru - če ga pokličete iz paketne datoteke ali iz kontekstnega menija.

Za nastavitve druge lokacije konfiguracijske datoteke uporabite parameter `/configdir:"pot\do\konfiguracijske\mape"`. S tem preprišete privzete vrednosti, vključno z zapisi v registru.

Pogovorno okno po zaključku izvedbe ukaza zaprete z uporabo parametra `/closeonend`. Na ta način ne izgubite stalne nastavitve.

- `/closeonend:0` okna ne zapri samodejno
- `/closeonend:1` samodejno zapri, če ni napak
- `/closeonend:2` samodejno zapri, če ni napak in sporov
- `/closeonend:3` samodejno zapri, če ni napak, sporov ali spajanj
- `/closeonend:4` samodejno zapri, če ni pri krajevnih operacijah ni napak, sporov ali spajanj

Spodnja tabela prikazuje vse ukaze, do katerih lahko dostopate preko programa `TortoiseProc.exe`. Ko smo že omenili zgoraj, uporabljamo obliko `/command:abcd`. V tabeli smo zaradi varčevanja s prostorom opustili predpono `/command`.

Ukaz	Opis
<code>:about</code>	Pokaže pogovorno okno "O programu". Prikaže se tudi, če ne podamo pokaza.

Ukaz	Opis
:log	Odpre dnevnik. /path določa datoteko ali mapo, za katero želite prikazati dnevnik. Nastavite lahko tri dodatne možnosti: /startrev:xxx, /endrev:xxx in /strict
:checkout	Odpre okno za prevzem. /path določa ciljno mapo, /url pa naslov, s katerega želite prevzeti vsebino.
:import	Odpre okno za uvoz. /path določa mapo s podatki, ki jih želite uvoziti.
:update	Posodobi delovno kopijo v /path na revizijo HEAD. Če je podana možnost /rev, se prikaže pogovorno okno, ki uporabniku ponudi vnos določene revizije, na katero naj se delovna kopija posodobi. Če se oknu želite izogniti, podajte določeno revizijo kot /rev:1234. Ostali možnosti sta /nonrecursive in /ignoreexternals.
:commit	Odpre pogovorno okno za objave. /path določa ciljno mapo ali seznam datotek za objavo. Določite lahko tudi stikalo /logmsg, s katerim določite sporočilo dnevniškega zapisa. Če pa ne želite podati sporočila v ukazni vrstici, uporabite /logmsgfile:pot, kjer pot kaže na datoteko, ki vsebuje sporočilo zapisa. Za izpolnitev polja ID hrošča (če ste pravilno nastavili integracijo s sledilnikom zadev) lahko uporabite /bugid:"številka hrošča".
:add	Doda datoteke v /path v sistem verzioniranja.
:revert	Povrne krajevne spremembe v delovni kopiji. /path pove, katere elemente je potrebno povrniti.
:cleanup	Počisti preklicane ali prekinjene operacije in odklene delovno kopijo v /path.
:resolve	Označi sporno datoteko, podano v /path, kot rešeno. Če je podana možnost /noquestion, se označevanje izvede brez uporanikove potrditve.
:reprocreate	V /path ustvari skladišče
:switch	Odpre okno za preklp. /path določa ciljno mapo.
:export	Izvozi delovno kopijo v /path v drugo mapo. Če /path kaže na mapo brez različic, se pojavi pogovorno okno, ki uporabnika vpraša po naslovu, ki naj se izvozi v /path.
:merge	Odpre okno za spajanje. /path določa ciljno mapo. Za spajanje območja revizij so na voljo naslednje možnosti: /fromurl:URL, /revrange:string. Za spajanje dveh dreves skladišča so na voljo naslednje možnosti: /fromurl:URL, /tourl:URL, /fromrev:xxx and /torev:xxx. S tem se nastavijo ustrezns polja v konu za spajanje.
:mergeall	Odpre okno za spajanje. /path določa ciljno mapo.
:copy	Prikaže okno za ustvarjanje vej/oznaka. /path je delovna kopija, iz katere želite ustvariti vejo/oznako. /url je ciljni naslov URL. Nastavite pa lahko tudi stikalo /logmsg, s katerim določite sporočilo dnevniškega zapisa. Če pa sporočila dnevniškega zapisa ne želite podati v kazni vrstici, uporabite /logmsgfile:pot, kjer pot kaže na datoteko s sporočilom dnevniškega zapisa.
:settings	Odpre okno za nastavitve
:remove	Odstrani datoteke v /path iz sistema Subversion.
:rename	Preimenuje datoteko v /path. Novo ime vpišete v pogovornem oknu. Če se želite izogniti vprašanju o preimenovanju podobnih datotek v enem koraku, podajte /noquestion.
:diff	Požene zunanje orodje za razlikovanje, ki je navedeno v nastavitvah TortoiseSVN. /path določa prvo datoteko. Če je možnost /path2

Ukaz	Opis
	nastavljena, se program zažene z navedenima datotekama. Če je možnost <code>/path2</code> izpuščena, potem se razlikovanje opravi med datoteko <code>/path</code> in njeno osnovo (BASE). Za določitev številke revizij uporabite <code>/startrev:xxx</code> in <code>/endrev:xxx</code> . Če je nastavljena možnost <code>/blame</code> in možnost <code>/path2</code> ni nastavljena, potem se razlikovanje naredi tako, da se naprej okrivi obe datoteki glede na podani reviziji.
<code>:showcompare</code>	Odvisno od naslovov URL in revizij, ki jih želite primerjati, se prikaže poenotena razlika (če je nastavljena možnost <code>unified</code>), seznam spremenjenih datotek ali (v primeru, da naslov URL kaže na datoteko) pregledovalnik sprememb za ti dve datoteki. Možnosti <code>url1</code> , <code>url2</code> , <code>revision1</code> in <code>revision2</code> so obvezne, možnosti <code>pegrevision</code> , <code>ignoreancestry</code> , <code>blame</code> in <code>unified</code> pa opcijske.
<code>:conflicteditor</code>	Zažene urejevalnik sporov, ki je naveden v nastavitvah TortoiseSVN, s pravilnimi datotekami za sporno datoteko <code>/path</code> .
<code>:relocate</code>	Odpre pogovorno okno za premestitev. <code>/path</code> določa pot delovne kopije za premestitev.
<code>:help</code>	Odpre datoteko pomoči.
<code>:repostatus</code>	Odpre pogovorno okno za preverjanje sprememb. <code>/path</code> določa mapo delovne kopije.
<code>:repobrowser</code>	Zažene skladišni brskalnik, ki kaže na naslov URL delovne kopije, podane v <code>/path</code> ali <code>/path</code> kaže neposredno na naslov URL. Dodatna možnost <code>/rev:xxx</code> omogoča nastavitve revizije, ki naj jo skladišni brskalnik prikaže. Če možnost <code>/rev:xxx</code> ni uporabljena, je privzeta nastavev HEAD. Če <code>/path</code> kaže na naslov URL, <code>/projectpropertiespath:pot/do/delovne/kopije</code> določa pot, iz katere preberemo in uporabimo projektne lastnosti.
<code>:ignore</code>	Doda vse cilje v <code>/path</code> na seznam prezrtih elementov, to pomeni, da datotekam doda lastnost <code>svn:ignore</code> .
<code>:blame</code>	Odpre pogovorno okno Okrivi za datoteko, podano z opcijo <code>/path</code> . Če sta nastavljeni možnosti <code>/startrev</code> in <code>/endrev</code> , se ne pojavi pogovorno okno za vnos območja revizij, ampak se uporabita navedeni vrednosti. Če je nastavljena opcija <code>/line:nnn</code> , se ob zagonu TortoiseBlame prikaže navedena vrstica. Podprte so tudi možnosti <code>/ignoreeol</code> , <code>/ignorespaces</code> in <code>/ignoreallspaces</code> .
<code>:cat</code>	Shrani datoteko z naslova URL ali delovne kopije, navedene v <code>/path</code> , na lokacijo <code>/savepath:pot</code> . <code>/revision:xxx</code> predstavlja številko revizije. S tem lahko prevzamemo datoteko v določeni reviziji.
<code>:createpatch</code>	Ustvari popravek za pot <code>/path</code> .
<code>:revisiongraph</code>	Prikaže graf revizij za pot, podano v <code>/path</code> .
<code>:lock</code>	Zaklene datoteko ali vse datoteke v mapi, podani v <code>/path</code> . Pogovorno okno za zaklepanje se pojavi, da lahko uporabnik vnese sporočilo zaklepanja.
<code>:unlock</code>	Odklene datoteko ali vse datoteke v mapi, podani v <code>/path</code> .
<code>:rebuildiconcache</code>	Ponovno zgradi predpomnilnik ikon sistema. Ta ukaz uporabite le v primeru, če se ikone na nek način poškodujejo. Neizogiben stranski učinek tega

Ukaz	Opis
	postopka je, da se ikone na namizju prerazporedijo po svoje. Če sporočila ne želite videti, podajte /noquestion.
:properties	Za pot, podano v /path, prikaže okno z lastnostmi.

Tabela D.1. Seznam ukazov in možnosti

Primeri (vpisati jih je potrebno v eno vrstico):

```
TortoiseProc.exe /command:commit
                 /path:"c:\svn_wc\file1.txt*c:\svn_wc\file2.txt"
                 /logmsg:"sporočilo" /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. Ukazi TortoiseIDiff

Orodje za razlikovanje slik omogoča nekaj možnosti ukazne vrstice, s katerimi lahko določite, kako se orodje zažene. Program se imenuje `TortoiseIDiff.exe`.

Spodnja tabela vsebuje seznam vseh možnosti, ki jih lahko v ukazni vrstici podate orodju za razlikovanje slik.

Možnost	Opis
:left	Pot do datoteke, prikazane na levi.
:lefttitle	Naslov. Uporablja se pri prikazu slike namesto polne poti datoteke.
:right	Pot do datoteke, prikazane na desni.
:righttitle	Naslov. Uporablja se pri prikazu slike namesto polne poti datoteke.
:overlay	Če je parameter podan, orodje za razlikovanje slik uporablja prekrivni način (alpha blend).
:fit	Če je parameter podan, orodje za razlikovanje slik stlači sliki v okno.
:showinfo	Pokaže okno z informacijami o sliki.

Tabela D.2. Seznam možnosti

Primer (vpisati ga je potrebno v eni vrstici):

```
TortoiseIDiff.exe /left:"c:\images\img1.jpg" /lefttitle:"image 1"
                 /right:"c:\images\img2.jpg" /righttitle:"image 2"
                 /fit /overlay
```

Dodatek E. Ustrezni ukazi v odjemalcu za ukazno vrstico

Ta priročnik vas včasih pozove, da pogledate dokumentacijo sistema Subversion, ki opisuje sistem Subversion z vidika uporabe preko ukazne vrstice (Command Line Interface - CLI). Da bi lažje razumeli, kaj se dogaja v ozadju, je tu seznam ukazov TortoiseSVN in ekvivalentnih ukazov odjemalca za ukazno vrstico.

Opomba

Čeprav za ukaze TortoiseSVN obstajajo ekvivalentni ukazi v ukazni vrstici, ne pozabite, da TortoiseSVN ukazne vrstice *ne* uporablja, ampak neposredno uporablja knjižnice sistema Subversion.

Če mislite, da ste našli napako v programu TortoiseSVN, vas bomo mogoče prosili, da jo poskusite ponoviti z uporabo odjemalca ukazne vrstice za Subversion. S tem ločimo napake programa TortoiseSVN od napak programa Subversion. Ta razpredelnica vam pove, katere ukaze poskusite.

E.1. Konvencije in osnovna pravila

V naslednjih opisih je naslov URL za lokacijo skladišča prikazan kot URL, n. pr. `http://tortoisesvn.googlecode.com/svn/trunk`. Delovno kopijo prikazujemo kot `PATH`, n. pr. `C:\TortoiseSVN\trunk`.



Pomembno

Ker je TortoiseSVN razširitev lupine Windows, ne more uporabljati trenutne mape. Vse poti delovne kopije morajo biti podane z absolutno in ne z relativno potjo.

Določeni elementi so opcijski in se ponavadi v TortoiseSVN nastavijo preko potrditvenih polj ali radijskih gumbov. Te opcije so v definicijah ukazne vrstice prikazane v [oglatih oklepajih].

E.2. Ukazi TortoiseSVN

E.2.1. Prezvemi

```
svn checkout [-N] [--ignore-externals] [-r rev] URL POT
```

Če je potrjena opcija **Prezvemi vrhjo mapo**, uporabi stikalo `-N`.

Če je potrjeno polje **Izpuusti zunanje**, uporabi stikalo `--ignore-externals`.

Če prevzimate določeno revizijo, pove to po navedbi naslova URL s stikalom `-r`.

E.2.2. Posodobi

```
svn info URL_delovne_kopije
svn update [-r rev] POT
```

Posodabljanje večjega števila elementov v sistemu Subversion ni atomična operacija. TortoiseSVN najprej najde revizijo HEAD v skladišču in potem posodobi vse elemente na to določeno številko revizije, da se izogne ustarjanju mešane delovne kopije.

Če je za posodobitev izbran le en element ali če niso vsi izbrani elementi iz istega skladišča, TortoiseSVN posodobi na revizijo HEAD.

Tu se ne uporabljajo možnosti ukazne vrstice. Posodobi na revizijo prav tako izvede posodabljanje, a ponudi več možnosti.

E.2.3. Posodobi na revizijo

```
svn info URL_delovne_kopije
svn update [-r rev] [-N] [--ignore-externals] POT
```

Če je potrjeno polje **Posodobi le vrhno mapo**, uporabi stikalo `-N`.

Če je potrjeno polje **Izpusti zunanje**, uporabi stikalo `--ignore-externals`.

E.2.4. Objavi

V programu TortoiseSVN pogovorno okno za objavo uporablja več ukazov sistema Subversion. Prvi ukaz je preverjanje stanja, ki v delovni kopiji najde elemente, ki so kandidati za objavo. Seznam lahko pregledate, pogledate spremembe datotek glede na revizijo BASE in izberete elemente, ki jih želite objaviti.

```
svn status -v POT
```

Če je potrjena možnost **Prikaži datoteke brez različic**, bo TortoiseSVN prikazal vse datoteke in mape brez različic v hierarhiji delovne kopije, pri tem pa upošteval pravila za prezrte elemente. Ta zmožnost nima neposrednega ekvivalentnega ukaza v sistemu Subversion, ker se ukaz `svn status` ne ukvarja z datotekami brez različic.

Če potrdite datoteke in mape brez različic, se le-te najprej dodajo v delovno kopijo.

```
svn add POT...
```

Ko kliknete na gumb **V redu**, se spremembe objavijo. Če ste pustili vsa potrditvena polja za izbiro datotek v privzetem stanju, TortoiseSVN uporabi enojno rekurzivno objavo delovne kopije. Če nekatere datoteke izločite, mora uporabiti nerekurzivno objavo (`-N`). Vsaka pot mora biti posamično navedena v ukazni vrstici ob objavi.

```
svn commit -m "Sporočilo" [-N] [--no-unlock] POT...
```

Sporočilo predstavlja vsebino vnosnega polja sporočila dnevniškega zapisa. Lahko je prazno.

Če je potrjena možnost **Ohrani zaklepe**, uporabi stikalo `--no-unlock`.

E.2.5. Razlikuj

```
svn diff POT
```

Če uporabljate ukaz **Razlikuj** iz glavnega kontekstnega menija, gledate spremembe v primerjavi z revizijo BASE. To se zgodi tudi pri uporabi ukazne vrstice, izhod pa je podan v obliki poenotene razlike. Vendar za prikaz razlik med dvema datotekama TortoiseSVN ne uporablja te možnosti, ampak TortoiseMerge (ali kak drug program za razlikovanje, ki ste ga izbrali), torej ekvivalentnega ukaza za ukazno vrstico ni.

S TortoiseSVN lahko razlikujete katerikoli dve datoteki, ne glede na to, ali sta pod nadzorom različic ali ne. TortoiseSVN enostavno pošlje datoteki v izbrani program za razlikovanje in mu prepusti, da vam pokaže razlike v datotekah.

E.2.6. Pokaži dnevnik

```
svn log -v -r 0:N --limit 100 [--stop-on-copy] POT
  or
svn log -v -r M:N [--stop-on-copy] POT
```

Po privezetih nastavitvah TortoiseSVN poskuša pridobiti 100 dnevniških zapisov z uporabo parametra `--limit`. Če nastavitve zahtevajo uporabo starega aplikacijskega vmesnika, potem se za pridobivanje 100 revizij dnevniških zapisov iz skladišča uporablja druga oblika.

Če je potrjeno polje **Ustavi ob kopiranju/preimenovanju**, uporabi stikalo `--stop-on-copy`.

E.2.7. Preveri posodobitve

```
svn status -v POT
  ali
svn status -u -v POT
```

Najprej se stanje preveri le v delovni kopiji. Če pa kliknete na **Preveri skladišče**, potem se preveri tudi stanje v skladišču, da se izve, katere datoteke bi se spremenile v primeru posodobitve. To zahteva stikalo `-u`.

Če je potrjena možnost **Prikaži datoteke brez različic**, bo TortoiseSVN prikazal vse datoteke in mape brez različic v hierarhiji delovne kopije, pri tem pa upošteval pravila za prezrte elemente. Ta zmožnost nima neposrednega ekvivalentnega ukaza v sistemu Subversion, ker se ukaz `svn status` ne ukvarja z datotekami brez različic.

E.2.8. Graf revizij

Graf revizij je zmožnost aplikacije TortoiseSVN. Ekvivalenten ukaz v ukazni vrstici ne obstaja.

TortoiseSVN naredi naslednje:

```
svn info URL_delovne_kopije
svn log -v URL
```

kjer je URL korenska pot skladišča, in potem analizira vrnjene podatke.

E.2.9. Brskalnik po skladišču

```
svn info URL_delovne_kopije
svn list [-r rev] -v URL
```

Da najdete korensko mapo skladišča, lahko uporabite ukaz `svn info`. Ta mapa se prikaže kot vrhnja mapa v brskalniku po skladišču. Višje od te mape ne morete. Ta ukaz vrne tudi vse informacije o zaklepih, ki se prav tako prikažejo v brskalniku po skladišču.

Ukaz `svn list` s parametrom URL in številko revizije vrne vsebino mape.

E.2.10. Uredi spore

Ekvivalenten ukaz za ukazno vrstico ne obstaja. Uporabi se orodje TortoiseMerge ali kakšen drug tristranski program za razlikovanje/spajanje, s pomočjo katerega se odločimo, kako bomo spore rešili.

E.2.11. Rešeno

```
svn resolved POT
```

E.2.12. Preimenuj

```
svn rename TRENUTNA_POT NOVA_POT
```

E.2.13. Izbriši

```
svn delete POT
```

E.2.14. Povrni

```
svn status -v POT
```

Prva faza predstavlja preverjanje stanja, s čemer ugotovimo, katere elemente v delovni kopiji je mogoče povrniti. Seznam lahko pregledate, pogledate razlike v primerjavi z revizijo BASE in izberete elemente, ki jih želite povrniti.

Ko kliknete na gumb V redu, se zgodi povrnitev. Če izbire niste spreminjali, uporabi TortoiseSVN eno samo rekurzivno (-R) povrnitev delovne kopije. Če ste izklopili nekatere datoteke, potem mora pri uporabi ukaza v ukazni vrstici podati vsako pot posebej.

```
svn revert [-R] POT...
```

E.2.15. Čiščenje

```
svn cleanup POT
```

E.2.16. Dobi zaklep

```
svn status -v POT
```

Prva faza je preverjanje stanja, ki ugotovi, katere datoteke v delovni kopiji lahko zaklenete. Nato izberete datoteke, ki jih želite zakleniti.

```
svn lock -m "Sporočilo" [--force] POT...
```

Sporočilo predstavlja vsebina sporočila zaklepa. Lahko je prazno.

Če je potrjeno polje Ukradi zaklep, uporabi stikalo --force.

E.2.17. Odstrani zaklep

```
svn unlock POT
```

E.2.18. Veja/Oznaka

```
svn copy -m "Sporočilo" URL URL  
ali  
svn copy -m "Sporočilo" URL@rev URL@rev  
ali
```



```
svn copy -m "Sporočilo" POT URL
```

Okno Veja/Oznaka izdelava kopijo skladišča. Na voljo so trije radijski gumbi:

- Revizija HEAD v skladišču
- Določena revizija v skladišču
- Delovna kopija

Ti ustrezajo trem opisanim variacijam ukazne vrstice.

Sporočilo predstavlja vsebino vnosnega polja sporočila dnevniškega zapisa. Lahko je prazno.

E.2.19. Preklop

```
svn info URL_delovne_kopije  
svn switch [-r rev] URL POT
```

E.2.20. Spoji

```
svn merge [--dry-run] --force Iz_URL@revN V_URL@revM POT
```

The Test Merge performs the same merge with the `--dry-run` switch.

```
svn diff Iz_URL@revN V_URL@revM
```

Gumb Poenotena različica pokaže operacijo razlikovanja, ki bo uporabljena za spajanje.

E.2.21. Izvozi

```
svn export [-r rev] [--ignore-externals] URL POT_za_izboz
```

Ta oblika se uporablja v mapi brez različic, kadar je ta mapa tudi ciljna mapa.

Izvoz delovne kopije na drugo lokacijo se ne izvede s pomočjo knjižnice Subversion, zato ekvivalentnega ukaza ni.

TortoiseSVN prekopira vse datoteke na novo lokacijo, hkrati pa prikazuje napredek operacije. Nastavite lahko, da se izvozijo tudi datoteke/mape brez različic.

Če je potrjeno potrditveno polje `Izpuusti zunanje`, v obeh primerih uporabi stikalo `--ignore-externals`.

E.2.22. Premakni

```
svn switch --relocate Iz_URL V_URL
```

E.2.23. Tu ustvari skladišče

```
svnadmin create --fs-type fsfs PATH
```

E.2.24. Dodaj

```
svn add POT...
```

Če izberete mapo, jo TortoiseSVN najprej rekurzivno pregleda, da najde elemente, ki jih je možno dodati.

E.2.25. Uvoz

```
svn import -m Sporočilo POT URL
```

Sporočilo predstavlja vsebino vnosnega polja sporočila dnevniškega zapisa. Lahko je prazno.

E.2.26. Okrivi

```
svn blame -r N:M -v POT  
svn log -r N:M POT
```

Če za pregled krivdnih informacij uporabljate TortoiseBlame, potem potrebujete tudi informacije o dnevniških zapisih datoteke, ki se prikažejo kot namig. Če krivdne informacije pregledujete v obliki besedilne datoteke, to ni potrebno.

E.2.27. Dodaj na seznam prezrtih

```
svn propget svn:ignore POT > začasnadatoteka  
{vstavi nov prezrti element v začasno datoteko}  
svn propset svn:ignore -F začasnadatoteka POT
```

Ker lastnost `svn:ignore` pogosto obsega več vrstic, tukaj prikazujemo, kako jo spreminjamo preko besedilne datoteke namesto neposredno v ukazni vrstici.

E.2.28. Ustvari popravek

```
svn diff POT > datotekapopravkov
```

TortoiseSVN ustvari datoteko popravkov v obliki poenotene razlike s primerjanjem delovne kopije in verzije BASE.

E.2.29. Namesti popravek

Nameščanje datotek popravkov je zahtevno opravilo, razen če imata datoteka popravkov in delovna kopija isto številko revizije. Na srečo lahko uporabite orodje TortoiseMerge, ki nima ekvivalentnega orodja v sistemu Subversion.

Dodatek F. Podrobnosti o izvedbi

Ta dodatek vsebuje podrobno razlago izvedbe nekaterih zmožnosti programa TortoiseSVN.

F.1. Prekrivne ikone

Vsaka datoteka in mapa pod nadzorom ima v sistemu Subversion stanje, ki ga sporoči knjižnica Subversion. V odjemalcu za ukazno vrstico je status prikazan z črko, v programu TortoiseSVN pa je prikazan grafično s prekrivnimi ikonami. Ker je število prekrivnih ikon zelo omejeno, lahko vsaka ikona predstavlja več vrednosti stanja.



Prekrivna ikona za *spor* se uporablja za prikaz spornega stanja. Pri posodobitvi ali preklopu se pri spajanju krajevnih sprememb in sprememb iz skladišča pojavijo spori. Uporablja se tudi za prikaz oviranega stanja, ki se pojavi, ko operacije ni mogoče zaključiti.



Prekrivna ikona *spremenjeno* predstavlja spremenjeno stanje pri krajevnih spremembah, spojeno stanje, ko so bile spremembe iz skladišča spojene s krajevnimi spremembami in zamenjano stanje, ko je bila datoteka izbrisana in nadomeščena z drugo datoteko z istim imenom.



Prekrivna ikona *izbrisano* predstavlja izbrisano stanje, kjer je element označen za brisanje ali stanje manjkajoče, ko elementa ni v delovni kopiji. Seveda element, ki ne obstaja, ne more imeti prekrivne ikone, zato je ikona za manjkajoč element pripeta na nadrejeno mapo.



Prekrivna ikona *dodano* predstavlja dodano stanje, ko je element dodan sistemu za nadzor različic.



Prekrivna ikona *V sistemu Subversion* predstavlja običajno stanje ali element pod nadzorom, katerega stanje še ni znano. Ker TortoiseSVN uporablja poseben proces za predpomnenje stanja, lahko traja nekaj sekund, preden se prekrivna ikona posodobi.



Prekrivna ikona *Potrebuje zaklep* se uporablja za datoteke, ki imajo nastavljeno lastnost `svn:needs-lock`. Za delovne kopije, ustvarjene z različico Subversion 1.4.0 ali novejšo, je stanje `svn:needs-lock` predpomnjeno krajevno in se uporablja tudi za prikaz prekrivne ikone. Za delovne kopije, ustvarjene z različico 1.4.x ali starejšo TortoiseSVN prikaže to ikono v primeru datoteke, ki je namenjena samo za branje. Upoštevajte, da Subversion samodejno nadgradi delovne kopije, ko jih posodobite. Predpomnenje lastnosti `svn:needs-lock` se morda ne bo zgodilo, dokler ne boste posodobili datoteke.



Prekrivna ikona *zaklenjeno* se uporablja, kadar je delovna kopija lastnica zaklepa za datoteko.



Ikona za *prezrte* elemente se uporablja za elemente, ki so *prezrti* zaradi splošnega vzorca prezrtih elementov ali zaradi nastavljene lastnosti `svn:ignore` na nadrejeni mapi. Prikaz je možno izključiti.



Ta ikona se uporablja za elemente *brez različic*. To so elementi, ki se nahajajo znotraj mape pod nadzorom, sami pa niso pod nadzorom. Prikaz te ikone je možno izključiti.

Če je stanje elementa *brez stanja* (element se ne nahaja v delovni kopiji), prekrivna ikona ni prikazana. Če ste izključili prikaz prekrivnih ikon za *prezrte* elemente in elemente *brez različic*, potem prekrivne ikone ne bodo prikazane tudi za te elemente.

Element ima lahko le eno stanje sistema Subversion. Primer: datoteka je krajevno spremenjena in hkrati označena za brisanje. Subversion vrne eno samo stanje - v tem primeru *izbrisan*. Prioritete stanj definira Subversion.

Kadar TortoiseSVN prikazuje stanje rekurzivno (privzeta nastavitvev), vsaka mapa prikazuje svoje stanje in stanje vseh svojih elementov. Da prikažemo le eno vrednost stanja - prekrivno ikono za *povzetek* - uporabljamo prioritetni red, ki je prikazan zgoraj. Najvišjo prioriteto ima stanje *sporno*.

Lahko pa ugotovite, da omenjene ikone niso vse na voljo na vašem sistemu. Razlog je v omejitvi sistema Windows, ki omogoča le petnajst prekrivnih ikon. Štiri uporablja operacijski sistem sam, enajst pa jih je na razpolago ostalim aplikacijam. Če na sistemu ni na voljo dovolj prostih mest, poskuša biti TortoiseSVN "pošten državljan (TM)" in ne uporablja veliko prekrivnih ikon, da ostane prostor tudi za ostale aplikacije.

- Ikone *Navadno*, *Spremenjeno* in *Sporno* se vedno naložijo in so vidne.
- Ikona *Izbrisan* se naloži, če je to mogoče, če pa prostih mest ni dovolj, se prikaže ikona *Spremenjeno*.
- Ikona *Samo za branje* se naloži, če je to mogoče, če pa prostih mest ni dovolj, se prikaže ikona *Običajno*.
- Ikona *Zaklenjeno* se naloži, če je naloženih manj kot trinajst ikon, v nasprotnem primeru pa se prikaže ikona *Običajno*.
- Ikona *Dodano* se naloži, če je naloženih manj kot štirinajst ikon, v nasprotnem primeru pa se prikaže ikona *Spremenjeno*.

Dodatek G. Zavarovanje strežnika Svnserve z SSL

V tem razdelku so opisani koraki, ki jih je potrebno izvesti, če želite pri sistemu Subversion in TortoiseSVN uporabljati protokol `svn+ssh`. Če za dostop do strežnika Subversion in avtentikacijo že uporabljate SSH, potem tega ne potrebujete; kaj več o tem si lahko preberete v knjigi *The Subversion Book*. Če še ne uporabljate SSH in želite svoj sistem Subversion zaščititi, vam ta razdelek ponuja enostaven način, kako to storite, ne da bi morali za vsakega uporabnika sistema Subversion ustvariti tudi račun SSH na strežniku.

Pri tem načinu ustvarimo en račun SSH za vse uporabnike sistema Subversion in uporabimo različne ključe za avtentikacijo, da ločimo med posameznimi uporabniki sistema Subversion.

V tem dodatku predpostavljamo, da imate nameščena vsa potrebna orodja in ustvarjeno skladišče, kot je opisano v predhodnjih poglavjih. Pri uporabi SSH strežnika `svnserve` *ne* smete poganjati kot storitev ali prikriti process.

Veliko informacij smo povzeli po učbeniku Marca Logemanna, ki ga najdete na povezavi www.logemann.org [<http://www.logemann.org/2007/03/13/subversion-tortoisesvn-ssh-howto/>] Dodatne informacije v zvezi s postavitvijo strežnika v sistemu Windows je podal Thorsten Müller. Hvala obema!

G.1. Nameščanje strežnika na operacijskem sistemu Linux

Na strežniku morate omogočiti SSH. Tu predpostavljamo, da uporabljate sistem OpenSSH. Na večini sistemov je OpenSSH že nameščen. Vtipkajte:

```
ps xa | grep sshd
```

in prikaže se vam seznam opravil SSH.

Če zgradite Subversion sami iz izvorne kode in ne podate parametra v `./configure`, Subversion ustvari podmapo `bin` v mapi `/usr/local` in tja shrani binarne datoteke. Če želite uporabiti tunelski način s SSH, se morate zavedati, da prijava uporabnika preko SSH zahteva izvržbo programa `svnserve` in še nekaterih knjižnic. Zato dodajte mapo `/usr/local/bin` v okoljsko spremenljivko `PATH` ali ustvarite simbolično povezavo binarnih datotek v mapo `/usr/sbin` ali katerokoli mapo, ki se nahaja v okoljski spremenljivki `PATH`.

Preverite, če vse deluje, tako da se s SSH prijavite kot ciljni uporabnik, in vtipkajte:

```
which svnserve
```

Ta ukaz vam pove, če je strežnik `svnserve` dosegljiv.

Ustvarite novega uporabnika, ki bo imel dostop do skladišča Subversion:

```
useradd -m svnuporabnik
```

Uporabniku dajte popoln dostop do skladišča.

G.2. Nameščanje strežnika na operacijskem sistemu Windows

Namestite Cygwin SSH daemon, kot je opisano na naslednjem naslovu: <http://pigtail.net/LRP/printsrv/cygwin-sshd.html>

Ustvarite nov račun v sistemu Windows in ga poimenujte `svnuporabnik`. S tem računom bomo dostopali do skladišča. Temu računu dajte popoln dostop do skladišča.

Če datoteka gesel še ne obstaja, jo ustvarite iz ukazne vrstice Cygwina:

```
mkpasswd -l > /etc/passwd
```

G.3. Odjemalci SSH za uporabo s TortoiseSVN

Orodja, ki jih potrebujemo za delo s SSH na sistemu Windows, dobite na naslednjem naslovu: <http://www.chiark.greenend.org.uk/~sgtatham/putty/> Pojdi na odsek Download in prenesite aplikacije Putty, Plink, Pageant in Puttygen.

G.4. Ustvarjanje certifikatov OpenSSH

V naslednjem koraku izdelamo par ključev za avtentikacijo. Za izdelavo ključev obstajata dva načina. Prvi način: ključa ustvarimo s programom PuTTYgen na strani odjemalca, javni ključ naložimo na strežnik, zasebni ključ pa uporabljamo s programom PuTTY. Drugi način: par ključev ustvarimo s programom `ssh-keygen`, ki je del OpenSSH, prenesemo zasebni ključ v odjemalca in ga pretvorimo v obliko PuTTY.

G.4.1. Ustvarjanje ključev s `ssh-keygen`

Na strežnik se prijavite z uporabniškim imenom `root` ali `svnupornik` in vnesite:

```
ssh-keygen -b 1024 -t dsa -N šifrirnafraza -f datotekaključev
```

pri čemer šifrirno frazo (ki jo poznate le vi) in datoteko ključev zamenjate z dejanskimi vrednostmi. Pravkar smo ustvarili ključ SSH2 DSA s 1024-bitno frazo. Če vnesete

```
ls -l datotekaključev*
```

vidite dve datoteki, `datotekaključev` in `datotekaključev.pub`. Datoteka `.pub` je javni ključ, datoteka brez končnice pa zasebni.

Javni ključ pripnite ključem v mapi `.ssh` znotraj domače mape uporabnika `svnuporabnik`:

```
cat keyfile.pub >> /home/svnuporabnik/.ssh/authorized_keys
```

Da zasebni ključ, ki smo ga ustvarili, lahko uporabimo, ga moramo pretvoriti v obliko PuTTY. Oblika privatnega ključa namreč ni standardizirana. Ko ste zasebni ključ prenesli na odjemalca, zaženite orodje PuTTYgen in izberite **Conversions** → **Import key**. Poiščite svojo datoteko `ključev`, ki ste jo dobili s strežnika, in uporabite šifrirno frazo, ki ste jo uporabili pri ustvarjanju ključa. Nazadnje kliknite na gumb **Save private key** in shranite datoteko z kot `datotekaključev.PPK`.

G.4.2. Ustvarjanje ključev s programom PuTTYgen

Za ustvarjanje para javnega in zasebnega ključa uporabite PuTTYgen. Ključa shranite. Javni ključ prekopirajte na strežnik in ga dodajte ključem v mapi `.ssh` znotraj domače mape uporabnika `svnuporabnik`:

```
cat keyfile.pub >> /home/svnuporabnik/.ssh/authorized_keys
```

G.5. Preizkušanje s programom PuTTY

Za preizkus povezave uporabimo PuTTY. Zaženite program in v zavihku **Session** nastavite ime gostitelja ali naslov IP vašega strežnika, nastavite protokol SSH in shranite sejo s poljubnim imenom, npr

SvnPovezava. Na zavihku **SSH** nastavite verzijo protokola na 2 in v zavihku **Auth** nastavite polno pot do datoteke zasebnega ključa .PKK, ki ste ga pred tem pretvorili. Vrnite se na zavihke **Sessions** in kliknite na gumb **Save**. Seja SvnPovezava bo sedaj navedena med shranjenimi sejami.

Kliknite **Open** in prikazalo sem vam bo okno za prijavo, podobno kot v programu telnet. Za uporabniško ime uporabite svnuporabnik in če ste vse naredili pravilno, se boste povezali na strežnik brez vnosa gesla.

Na strežniku boste mogoče morali urediti datoteko /etc/sshd_config. Uredite datoteko, kot je navedeno spodaj, nato pa ponovno zaženite storitev SSH.

```
PubkeyAuthentication yes
PasswordAuthentication no
PermitEmptyPasswords no
ChallengeResponseAuthentication no
```

G.6. Preizkušanje SSH s TortoiseSVN

Preverili smo le, če se lahko prijavite preko SSH. Sedaj moramo zagotoviti, da ta povezava lahko poganja svnserve. Na strežniku spremenite datoteko /home/svnuser/.ssh/authorized_keys, kot je navedeno spodaj. S tem omogočite, da več uporabnikov sistema Subversion istočasno uporablja isti sistemski račun, svnuporabnik. Upoštevajte, da vsak uporabnik sistema Subversion uporablja iste podatke za prijavo, vendar ima drugačen avtentikacijski ključ, zato morate dodati za vsakega uporabnika eno vrstico.

Opomba: besedilo je potrebno podati v eni vrstici.

```
command="svnserve -t -r <ReposRootPath> --tunnel-user=<author>",
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,
no-pty ssh-rsa <PublicKey> <Comment>
```

Vrednosti morate nastaviti tako, da ustrezajo vašemu sistemu.

<ReposRootPath> je potrebno zamenjati s potjo do mape, ki vsebuje vaša skladišča. Na ta način nam ni potrebno navajati polnih poti pri navajanju naslovov URL. Upoštevajte, da je potrebno tudi v primeru uporabe operacijskega sistema Windows uporabiti poševnice, n. pr. c:/svn/reposroot. V naslednjih primerih predpostavljamo, da se skladiščna mapa nahaja v mapi sklad v korenski mapi skladišča.

Niz <author> je potrebno zamenjati z uporabniškim imenom avtorja, ki naj se uporabi pri objavi. Prav tako s tem omogočimo procesu svnserve uporabo lastne avtorizacije z datoteko svnserve.conf.

<PublicKey> je potrebno nadomestiti z javnim ključem, ki ste ga pred tem ustvarili.

<Comment> je lahko poljuben komentar, vendar je na tem mestu priporočljivo zapisati uporabniško ime v sistemu Subversion in pravo ime uporabnika.

Desno kliknite na katerokoli mapo v Raziskovalcu in izberite TortoiseSVN → Brskalnik po skladišču. Vnesti boste morali naslov URL. Vnesite ga v naslednji obliki:

```
svn+ssh://svnuporabnik@SvnPovezava/sklad
```

Kaj ta naslov pomeni? Ime sheme je svn+ssh. To programu TortoiseSVN pove, kako obravnavati zahteve do strežnika. Za dvojno poševnico sledi ime uporabnika, ki se priklaplja na strežnik, v našem primeru je to svnuporabnik. Po znaku @ sledi ime seje v programu PuTTY. Seja vsebuje podatke o zasebnem ključu, IP naslovu strežnika, DNS strežniku. Nazadnje podamo še pot do skladišča in sicer relativno na korensko mapo skladišč na strežniku, kot je podano v datoteki authorized_keys.

Kliknite na gumb **V redu** in sedaj naj bi imeli možnost brskanja po vsebini skladišča. Če je res tako, imate delujoč tunel SSH v povezavi s TortoiseSVN.

Upoštevajte, da po privzetih nastavitvah TortoiseSVN za povezavo uporablja lastno različico aplikacije Plink. S tem se izognemo pojavljanju konzolnih oken ob vsakem poskusu avtentikacije, hkrati pa to pomeni, da sporočil o napakah ni možno prikazati. Če se pojavi napaka "Unable to write to standard output", lahko poskusite nastaviti Plink za odjemalca v mrežnih nastavitvah TortoiseSVN. Tako boste lahko videli dejansko obvestilo o napaki, ki ga vrne Plink.

G.7. Variacije konfiguracije SSH

Eden izmed načinov, kako poenostavimo naslov URL v TortoiseSVN, je, da nastavimo uporabnika znotraj seje PuTTY. Naložite že narejeno sejo `SvnPovezava` v PuTTY in v zavihku **Connection** nastavite **Auto login user** na uporabniško ime, n. pr. `svnuporabnik`. Sejo shranite in poskusite v TortoiseSVN uporabiti naslednji naslov URL:

```
svn+ssh://SvnPovezava/sklad
```

V tem primeru odjemalcu SSH, ki ga uporablja TortoiseSVN (TortoisePlink.exe), podamo le ime seje. Odjemalec bo podatke pridobil iz navedene seje.

V trenutni različici orodja PuTTY le-ta ne preverja vseh shranjenih konfiguracij. Če imate več konfiguracij, ki se nanašajo na isti strežnik, PuTTY uporabi prvega ustreznega. Poleg tega se lahko zgodi, da se pri urejanju privzete konfiguracije in njenem shranjevanju *ne* shrani uporabniško ime, ki se uporablja za samodejno prijavo.

Veliko ljudi uporablja Pageant za shranjevanje ključev. Ker je seja PuTTY sposobna shranjevati ključe, programa Pageant ne potrebujete vedno. Predstavljajte pa si, da želite shraniti ključe za različne strežnike; v tem primeru bi morali sejo PuTTY urejevati znova in znova, odvisno od strežnika, s katerim bi se želeli povezati. V tem primeru je uporaba programa Pageant zelo smiselna. Ko PuTTY, Plink, TortoisePlink ali katerikoli program, ki temelji na programu PuTTY, poskuša narediti povezavo do strežnika SSH, le-ta preveri vse ključe, ki jih ima shranjene Pageant, da ustvari povezavo.

Poženite Pageant in dodajte zasebni ključ. Uporabiti morate istega kot v seji programa PuTTY zgoraj. Če uporabljate program Pageant za shranjevanje zasebnih ključev, lahko zbrisete referenco na zasebni ključ, ki ste ga shranili v seji PuTTY. Seveda lahko dodate tudi ključe za druge strežnike ali uporabnike.

Če tega postopka ne želite ponavljati ob vsakem zagonu računalnika, dodajte Pageant v mapo za samodejni zagon (Startup). Ključe vključno s potmi lahko dodate kot parametre ukazne vrstice programa Pageant.exe

Zadni način priklopa na strežnik SSH je z uporabo naslednjega naslova URL znotraj TortoiseSVN:

```
svn+ssh://svnuporabnik@100.101.102.103/sklad
svn+ssh://svnuporabnik@mojadomena.si/sklad
```

Kot vidite, ne uporabljamo shranjene seje programa PuTTY, ampak naslov IP (ali domeno) kot cilj priklopa. Dodamo tudi uporabnika. Verjetno se vprašate, kako najdemo zasebni ključ. Ker je TortoisePlink.exe le predelana verzija standardnega programa Plink iz svežnja programov PuTTY, TortoiseSVN poskusi uporabiti vse ključe, ki so shranjeni v Pageantu.

If you use this last method, be sure you do not have a default username set in PuTTY. We have had reports of a bug in PuTTY causing connections to close in this case. To remove the default user, simply clear `HKEY_CURRENT_USER\Software\SimonTatham\Putty\Sessions\Default%20Settings\HostName`

Slovar

BDB	Berkeley DB. Dobro preizkušena baza podatkov za skladišča, ki se ne more uporabljati v mapah v skupni rabi v omrežju. To je privzeta baza za skladišča pred različico 1.2.
Čiščenje	Citat iz knjige The Subversion Book: “Rekurzivno počisti delovno kopijo, tako da odstrani zaklepe in zaključi nedokončane operacije. Če se vam kdaj prikaže sporočilo <i>delovna kopija zaklenjena</i> , poženite ta ukaz, da ostanite zaklepe in naredite delovno kopijo spet uporabno.” Upoštevajte, da v tem kontekstu <i>zaklep</i> pomeni zaklepanje datotečnega sistema in ne zaklepanje skladišča.
Delovna kopija	To je vaš krajeni “peskovnik”, področje, kjer lahko delate z datotekami pod nadzorom in se običajno nahaja na vašem trdem disku. Z uporabno ukaza “Prezemi” naredite delovno kopijo iz skladišča, spremembe, ki jih naredite, pa shranite v skladišče z ukazom “Objavi”.
Dnevnik	Prikaže zgodovino revizij datoteke ali mape. Uporablja se tudi termin “Zgodovina”.
Dodaj	Ukaz Subversion, ki se uporablja za dodajanje datoteke ali mape v delovno kopijo. Novi elementi so dodani v skladišče, ko naredite objavo.
FSFS	Ekskluziven datotečni sistem za skladišča sistema Subversion. Uporablja se lahko tudi v omrežnih datotekah v skupni rabi. Privzet sistem pri različicah 1.2 in novejših.
GPO	Group policy object
Izbriši	Ko izbrišete datoteko pod nadzorom (in objavite spremembo), tega elementa ni več v skladišču od objavljene revizije naprej. Seveda pa še vedno obstaja v prejšnjih revizijah skladišča in tako lahko še vedno dostopate do njega. Če je potrebno, lahko skopirate izbrisan element in ga pripeljete nazaj med žive s celotno zgodovino.
Izvozi	Ta ukaz ustvari kopijo mape pod nadzorom, prav takšno, kot je delovna kopija, vendar brez krajevnih map <code>.svn</code> .
Kopiraj	V skladišču Subversion lahko naredite kopijo datoteke ali celotnega drevesa. Kopije so izvedene kot “poceni kopije”, kar je nekaj podobnega kot povezava na originalen element in ne zavzema skoraj nič prostora. Kopija ohrani zgodovino originala, tako da lahko vedno pogledate spremembe, ki so bile narejene preden se je element prekopiral.
Lastnost	Poleg vodenja različic map in datotek Subversion omogoča tudi vodenje različic meta podatkov - rečemo jim “lastnosti” - za vsak element pod nadzorom. Vsaka lastnost ima ime in vrednost, podobno kot register. Subversion interno uporablja nekaj posebnih lastnosti, n. pr. <code>svn:eol-style</code> . Tudi TortoiseSVN uporablja nekaj internih lastnosti, n. pr. <code>tsvn:logminsize</code> . Lahko pa dodate svoje lastnosti s poljubnim imenom in vrednostjo.
Lastnost revizije (revprop)	Prav tako, kot imajo lastnosti datoteke, jih imajo tudi revizije v skladišču. Nekatero posebne se dodajo samodejno, ko se revizija ustvari: <code>svn:date</code> <code>svn:author</code> <code>svn:log</code> , ki predstavljajo

	<p>datum/čas objave, avtorja in sporočilo dnevniškega zapisa. Vrednosti teh lastnosti lahko urejate, vendar upoštevajte, da niso pod nadzorom različic, tako da so spremembe trajne in se jih ne da povrniti.</p>
Objavi	<p>Ta ukaz sistema Subversion se uporablja za pošiljanje sprememb v delovni kopiji nazaj na skladišče. S tem se tam ustvari nova revizija.</p>
Okrivi	<p>Ta ukaz se uporablja samo pri besedilnih datotekah. Vsako vrstico datoteke označi z informacijo, v kateri reviziji je bila nazadnje spremenjena in kdo je spremembo naredil. Grafični ekvivalent temu ukazu je TortoiseBlame, ki vam pokaže tudi datum in dnevniški zapis, če greste z miško na številko revizije.</p>
Popravek	<p>Če so se v delovni kopiji spremenile le besedilne datoteke, je možno z ukazom Razlikuj ustvariti datoteko s povzetkom sprememb v obliki poenotene razlike (unified diff). Datoteka takšnega tipa se pogosto imenuje "popravek" in se lahko uporablja za pošiljanje sprememb po elektronski pošti vašemu sodelavcu ali dopisnemu seznamu. Ta datoteka se potem uporabi na neki drugi delovni kopiji. Nekdo, ki nima pravic pisanja po skladišču, lahko naredi spremembe in ustvari popravek, ki ga objavi nek drug uporabnik, ki ustrezne pravice ima. Ali pa če niste prepričani, da so vaše spremembe dobre, lahko na ta način drugim uporabnikom pošljete spremembe v pregled.</p>
Posodobi	<p>Ta ukaz prenese vse zadnje spremembe iz skladišča v delovno kopijo, pri tem pa spoji spremembe, ki so jih naredili ostali uporabniki, z vašimi spremembami v delovni kopiji.</p>
Povrni	<p>Subversion hrani krajevno "izvirno" kopijo vsake datoteke, kakršna je bila ob zadnji posodobitvi. Če ste na datoteki naredili spremembe in jih želite povrniti, uporabite ukaz "povrni", da dobite nazaj izvirno datoteko.</p>
Preklop	<p>Prav tako kot ukaz "posodobi-na-revizijo" spremeni časovno okno delovne kopije, da izgleda tako, kot v neki določeni točki v zgodovini, ukaz "Preklopi" spremeni prostorsko okno delovne kopije, da kaže na drugo mesto v skladišču. To je uporabno predvsem takrat, kot delate na glavni veji in na stranskih vejah, na katerih je spremenjenih le nekaj datotek. Delovno kopijo lahko preklapljate, prenesejo pa se le spremenjene datoteke.</p>
Premakni	<p>Če se skladišče premakne (ker ste ga premaknili v drugo mapo na strežniku ali pa se je spremenilo domensko ime), morate narediti "premik" delovne kopije, da bodo poti URL do skladišča kazale na novo lokacijo.</p> <p>Opomba: ta ukaz uporabljajte le, če delovna kopija kaže na isto lokacijo znotraj istega skladišča, skladišče pa se je premaknilo na drugo lokacijo. V vseh ostalih primerih uporabite ukaz "Preklopi".</p>
Prezemi	<p>Ukaz, ki ustvari krajevno delovno kopijo v prazni mapi in prenese datoteke pod nadzorom iz skladišča.</p>
Razlikuj	<p>Bližnica za "Prikaži razlike". Zelo uporabno, ko želite vedeti, kaj točno se je spremenilo.</p>
Reši	<p>Ko so datoteke v delovni kopiji sporne (zaradi spajanja), mora uporabnik te spore rešiti z uporabo urejevalnika (ali pa programa TortoiseMerge). Procesu rečemo "reševanje sporov". Ko ste spore rešili, označite datoteko kot rešeno, kar vam omogoča njeno objavo.</p>

Revizija	<p>Vsakič ko objavite spremembe, naredite novo “revizijo” v skladišču. Vsaka revizija predstavlja stanje skladišča ob določeni točki v njegovi zgodovini. Če želite, si lahko ogledate skladišče, kakršno je bilo ob reviziji številka N.</p> <p>Drug pogled: revizija je množica sprememb, ki so bile narejene, ko je bila revizija ustvarjena.</p>
Revizija BASE	Trenutna osnovna revizija datoteke v vaši <i>delovni kopiji</i> . To je različica datoteke, kakršna je bila pri zadnjem prevzemu, posodobitvi ali objavi. Revizija BASE običajno ni ista kot revizija HEAD.
Revizija HEAD	Najnovejša revizija datoteke ali mape v <i>skladišču</i> .
Skladišče	Skladišče je središčna lokacija, kjer se podatki shranjujejo in vzdržujejo. Skladišče je lahko lokacija, kjer so shranjene številne baze podatkov ali datoteke za distribucijo preko omrežja, lahko pa je tudi lokacija, ki je neposredno dostopna uporabniku, ne da bi uporabljal omrežje.
Spoj	<p>Proces dodajanja sprememb iz skladišča v vašo delovno kopijo, ne da ob tem povozite krajevne spremembe. Včasih sprememb ni možno samodejno spojiti in takrat je delovna kopija sporna.</p> <p>Spajanje se zgodi samodejno, ko posodabljate delovno kopijo. Lahko pa opravljate spajanje specifičnih sprememb iz druge veje z uporabo ukaza Spoji.</p>
Spor	Ko spremembe iz skladišča spojimo s krajevnimi spremembami, se včasih spremembe zgodijo na isti vrstici. V takšnem primeru se Subversion ne more sam odločiti, katero različico naj uporabi in datoteka je v spornem stanju. Pred objavo morate datoteko ročno urediti in rešiti spor.
SVN	<p>Pogosto uporabljena okrajšava za Subversion.</p> <p>Ime lastnega protokola sistema Subversion, ki ga uporablja strežnik skladišč “svnserve”.</p>
Uvoz	Ukaz sistema Subversion, ki omogoča uvoz celotne hierarhije map v skladišče v eni reviziji.
Veja	Ta izraz, ki se pri nadzoru različic pogosto uporablja, opisuje, kaj se zgodi, ko se razvoj prelomi v določeni točki in se nadaljuje v dveh smereh. Iz glavne veje lahko ustvarite vejo, da razvijate novo zmožnost programa, ne da bi naredili glavno vejo nestabilno. Lahko pa naredite vejo iz stabilne verzije programa, da naredite popravke, razvoj pa se nadaljuje na nestabilni veji. V sistemu Subversion so veje izvedene kot “poceni kopije”.
Zaklep	Ko element pod nadzorom zaklenete, ga s tem označite v skladišču in preprečite objave, razen iz delovne kopije, ki je zaklep zahtevala.
Zgodovina	Prikaže zgodovino revizij datoteke ali mape. Uporablja se tudi termin “Dnevnik”.

Stvarno kazalo

Simboli

'delovna kopija' brez različic, 124
.svn folder, 161
_svn folder, 161

A

akcije, 19
akcije na strani strežnika, 117
Apache, 25
auto-props, 93
avtentikacija, 39
avtomatizacija, 180, 183
Avtorizacija, 29

B

bližnjica, 175
brez različic, 126, 176
brskalnik po skladišču, 117

C

CLI, 184
COM, 162, 167
compare folders, 174
črkovalnik, 3

D

datoteke/mape brez različic, 83
delovna kopija, 9
desni klik, 36
dnevnik, 62
dobi spremembe, 50
dodaj, 81
dodaj datoteke v skladišče, 40
Domena Windows, 30
domenski kontroler, 30, 177
Dostop, 16
držalo za poteg, 38

E

elementi kontekstnega menija, 178

F

filter, 71

G

GPO, 177
graf, 119
graf revizij, 119

H

hvali, 115

I

IBugtraqProvider, 167
ikone, 56
Indeks projektov, 29
iskanje najnovejše verzije, 177
iskanje vzorcev, 84
izbriši, 85
izluščenje verzije, 162
izvoz, 124
izvozi spremembe, 77

J

jezikovni paketi, 3

K

ključne besede, 91
kontekstni meni, 36
kopija, 98, 117
kopiraj datoteke, 81

L

lasnosti projekta, 93
lastnosti, 89
lastnosti Subversion, 90
lastnosti TortoiseSVN, 93
Lupina Windows, 1

M

mod_authz_svn, 26, 29
msi, 177

N

nadzor različic, 1
namesti, 3
namestitev, 177
nastavitve, 132
NTLM, 30

O

objava, 45
objavi, 45
odjemalec za ukazno vrstico, 184
odstrani, 85
odstrani iz skladišča, 176
odstrani verzioniranje, 176
okrivi, 115
Omrežni deljeni pogon, 16
onemogoči funkcije, 178
orodja za razlikovanje, 80
orodja za spajanje, 80
označi, 115
oznaka, 81, 98

P

plugin, 167
počisti, 89

poenotena razlika, 113
 poglej nove datoteke, 81
 poglej spremembe, 56
 Pogoni SUBST, 144
 Pogosto zastavljena vprašanja, 171
 popravek, 113
 popravljanje zapisa/avtorja, 70
 posebne datoteke, 42
 pošlji spremembe, 45
 posodobitev, 50, 173
 Posredniški strežnik, 145
 poteg z desnim gumbom, 38
 Poti UNC, 16
 povečaj, 40
 povezava, 19
 povezava TortoiseSVN, 19
 povezava za prevzem, 19
 povleci-in-spusti, 38
 povrni, 88, 173
 povrnitev, 173
 pravice skupin, 177, 178
 pravzno sporočilo, 172
 predpomnilnik dnevnika, 152
 pregledovalnik skladišča, 131
 pregledovalnik strežnika, 117
 preimenuj, 86, 117, 172
 preimenuj datoteke, 81
 preklop, 100, 126
 prekrivki, 56, 190
 premakni, 86
 premakni datoteke, 81
 premikanje, 172
 preseljen strežnik, 126
 preveri obstoj nove verzije, 177
 prevodi, 3
 prevzem, 42
 prezri, 83
 primerja datoteke, 174
 primerjaj, 75
 primerjaj revizije, 77
 prioriteta prekrivnih ikon, 190
 projekti ASP, 178

R

raziskovalec, 1
 razlika, 75, 113
 razlike slike, 79
 razlikovanje, 59
 razširi ključne besede, 91
 razveljavi, 88
 razveljavi objavo, 173
 razveljavi spremembo, 173
 register, 159
 reši, 52
 revision properties, 70
 revizija, 11, 119
 revizija v datoteki, 162
 revprops, 70

S

samo za branje, 110
 SASL, 23
 selednje hroščev, 127
 seznam sprememb, 60
 skladišče, 4, 40
 skupni projekti, 174
 sledenje spajanja, 108
 sledilnik zadev, 127, 127, 127, 167
 slovar, 3
 spletna stran, 19
 spletni pogled, 131
 splošni vzorec prezrtih, 133
 spoji, 101

- dve drevesi, 105
- območje revizij, 102
- vklučitev, 104

 spor, 8, 52
 spori spajanja, 108
 sporočila objav, 62
 sporočilo objave, 62
 sprememba URL, 126
 spremembe, 58, 174
 spremembe velikih in malih črk, 86
 spremembnjen URL naslov skladišča, 126
 spročilo objave, 172
 SSL, 32
 SSPI, 30
 stanje, 56, 58
 stanje delovna kopija, 56
 statistika, 72
 Stolpci v Raziskovalcu, 57
 strežnik preseljen, 126
 Subversion book, 4
 SubWCRev, 162
 SVNParentPath, 28, 29
 SVNPath, 28
 svnservice, 20, 21
 SVN_ASP_DOT_NET_HACK, 178

T

TortoiseIDiff, 79
 tree conflict, 52

U

ukazna vrstica, 180, 183
 ukazne datoteke akcij, 19, 155
 ukazne datoteke akcij na strežniku, 19
 ukazne datoteke akcij odjemalca, 155
 uredi, 172
 Ustvari

- Odjemalec za ukazno vrstico, 15
- TortoiseSVN, 15

 ustvari delovno kopijo, 42
 ustvari skladišče, 15
 usvari izdajo, 98
 uvažanje na mestu, 42

uvoz, 40

V

varnostna kopija, 18

Večkratna avtentikacija, 31

veja, 81, 98

veje zunanjih izvajalcev, 174

verzija, 177

ViewVC, 131

vkjučitev veje, 109

vmesnik COM SubWCRev, 164

VS2003, 178

vzorci, 84

vzorec za izključevanje, 133

W

WebDAV, 25

WebSVN, 131

Z

začasne datoteke, 40

zaklepanje, 110

zapis dnevnika, 172

zapis sledenja spajanja, 69

zgodovina, 62

zunajnja skladišča, 95

zunajni, 95, 174

zvoki, 132