

TortoiseSVN

Cliente de Subversion para Windows

Versão 1.6.16

**Stefan Küng
Lübbe Onken
Simon Large**

TortoiseSVN: Cliente de Subversion para Windows: Versão 1.6.16

por Stefan Küng, Lübbe Onken e Simon Large

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

Índice

Prefácio	xi
1. Audiência	xi
2. Guia de Leitura	xi
3. O TortoiseSVN é grátis!	xii
4. Comunidade	xii
5. Agradecimentos	xii
6. Terminologia utilizada neste documento	xii
1. Introdução	1
1.1. O que é o TortoiseSVN?	1
1.2. A história do TortoiseSVN	1
1.3. Funcionalidades do TortoiseSVN	1
1.4. Instalando o TortoiseSVN	2
1.4.1. Requisitos de sistema	2
1.4.2. Instalação	3
1.4.3. Pacotes de Linguas	3
1.4.4. Corretor Ortográfico	3
2. Conceitos Básicos de Controlo de Versões	5
2.1. O Repositório	5
2.2. Modelos de Controlo de Versões	5
2.2.1. O Problema da Partilha de Ficheiros	6
2.2.2. A Solução Bloquear-Modificar-Desbloquear	6
2.2.3. A solução Copiar-Modificar-Integrar	8
2.2.4. O que faz o Subversion?	10
2.3. O Subversion em Acção	10
2.3.1. Cópias de Trabalho.	10
2.3.2. URLs do Repositório	12
2.3.3. Revisões	12
2.3.4. Como as Cópias de Trabalho Seguem o Repositório	14
2.4. Resumo	15
3. O Repositório	16
3.1. Criação do Repositório	16
3.1.1. Criar um Repositório com o Cliente de Linha de Comandos	16
3.1.2. Criar O Repositório Com o TortoiseSVN	16
3.1.3. Acesso Local ao Repositório	17
3.1.4. Aceder ao Repositório numa Partilha de Rede	17
3.1.5. Estrutura do Repositório	18
3.2. Cópia de Segurança do Repositório	19
3.3. Scripts de gancho de servidor	20
3.4. Ligações de Checkout	20
3.5. Aceder ao Repositório	21
3.6. Servidor Baseado em Svnserve	21
3.6.1. Introdução	21
3.6.2. Instalar o svnserve	21
3.6.3. Executar o svnserver	22
3.6.4. Autenticação Básica com o svnserve	24
3.6.5. Segurança Melhorada com SASL	25
3.6.6. Autenticação com svn+ssh	26
3.6.7. Autorização com o svnserve baseada em Caminhos	26
3.7. Servidor Baseado em Apache	26
3.7.1. Introdução	26
3.7.2. Instalar o Apache	27
3.7.3. Instalar o Subversion	28
3.7.4. Configuração	28
3.7.5. Repositórios Múltiplos	30
3.7.6. Autorização Baseada-em-Caminhos	31

3.7.7. Autenticação Com um Domínio Windows	31
3.7.8. Múltiplas Fontes de Autenticação	33
3.7.9. Proteger o servidor com SSL	34
3.7.10. Usando certificador cliente com hospedeiros SSL virtuais	36
4. Guia de Uso Diário	37
4.1. Começando	37
4.1.1. Sobreposição de Ícones	37
4.1.2. Menus de Contexto	37
4.1.3. Arrastar e Largar	39
4.1.4. Atalhos comuns	40
4.1.5. Autenticação	40
4.1.6. Maxiizando Janelas	41
4.2. Importando Dados Para Um Repositório	41
4.2.1. Importar	41
4.2.2. Importar no local	43
4.2.3. Ficheiros Especiais	43
4.3. SVN Exportar Para Uma Cópia de Trabalho	44
4.3.1. Profundidade do Checkout	44
4.4. Submetendo as tuas alterações para o Repositório	46
4.4.1. A Caixa de Diálogo Submeter	46
4.4.2. Listas de Alterações	49
4.4.3. Excluir itens da lista a Submeter	49
4.4.4. Mensagens de Registo de Submeter	49
4.4.5. Progresso do Submeter	51
4.5. Actualizar a Tua Cópia de Trabalho Com Alterações de Outros	52
4.6. Resolvendo Conflitos	54
4.6.1. Conflitos de Ficheiro	54
4.6.2. Conflitos de Árvore	55
4.7. Obter informação de Estado	58
4.7.1. Sobreposição de Ícones	58
4.7.2. Colunas TortoiseSVN no Explorador do Windows	59
4.7.3. Estado Remoto e Local	60
4.7.4. Vendo diferenças	62
4.8. Listas de Alterações	62
4.9. Caixa de Diálogo Registo de Revisões	64
4.9.1. Invocando a Caixa de Diálogo Registo de Revisão	65
4.9.2. Acções de Registo de Revisões	65
4.9.3. Obtendo Informação Adicional	66
4.9.4. Obtendo mais mensagens de registo	70
4.9.5. Revisão Actual da Cópia de Trabalho	71
4.9.6. Funcionalidades de Rastreamento de Integração	71
4.9.7. Alterando a Mensagem de Registo e Autor	72
4.9.8. Filtrando Mensagens de Registo	73
4.9.9. Informação Estatística	73
4.9.10. Modo Fora de Linha	77
4.9.11. Refrescar a Vista	77
4.10. Ver Diferenças	77
4.10.1. Diferenças em ficheiros	78
4.10.2. Opções de Fim-de-Linha e Espaços-Brancos	79
4.10.3. Comparando Pastas	79
4.10.4. Comparando Imagens usando o TortoiseDiff	81
4.10.5. Ferramentas de Comparação/Integração	82
4.11. Adicionar Novos Ficheiros e Pastas	82
4.12. Copiando/Movendo/Renomeando Ficheiros e Pastas	83
4.13. Ignorando Ficheiros E Pastas	84
4.13.1. Correspondência de Padrões em Listas de Ignorados	85
4.14. Removendo, Movendo e Renomeando	86
4.14.1. Removendo ficheiros e pastas	87

4.14.2. Movendo ficheiros e pastas	88
4.14.3. Mudando maiúsculas e minúsculas no nome do ficheiro	89
4.14.4. Lidando com conflitos de maiúsculas e minúsculas no nome do ficheiro	89
4.14.5. Reparando Renomeações de Ficheiros	89
4.14.6. Removendo Ficheiros Não Versionados	90
4.15. Desfazer Alterações	90
4.16. Limpar	91
4.17. Configurações de Projecto	91
4.17.1. Propriedades Subversion	92
4.17.2. Propriedades de Projecto TortoiseSVN	96
4.18. Itens Externos	98
4.18.1. Pastas Externas	98
4.18.2. Ficheiros Externos	100
4.19. Ramificando/Etiquetando	100
4.19.1. Criando um Ramo ou Etiqueta	101
4.19.2. SVN Exportar ou Trocar...	102
4.20. Integrar	103
4.20.1. A Integrar Um Intervalo de Revisões	105
4.20.2. Reintegrar um ramo	106
4.20.3. A Integrar Duas Árvores Diferentes	107
4.20.4. Opções de Integração	108
4.20.5. Rever os Resultados de Integração	109
4.20.6. Rastreamento de Integração	110
4.20.7. Lidando com Conflitos durante a Integração	110
4.20.8. Integrar um Ramo Completo	111
4.20.9. Manutenção do Ramo de Funcionalidade	112
4.21. A bloquear	112
4.21.1. Como Funciona o Sistema de Bloqueio no Subversion	113
4.21.2. Obter um Bloqueio	113
4.21.3. Libertar um Bloqueio	114
4.21.4. Verificar o Estado dos Bloqueio	115
4.21.5. Tornar os Ficheiros Não-Bloqueados Só de Leitura	115
4.21.6. Os Scripts do Gancho de Bloqueio	116
4.22. Criar e Aplicar Correções	116
4.22.1. Criar um Ficheiro de Correção	116
4.22.2. Aplicar um Ficheiro de Correção	117
4.23. Quem Alterou Que Linha	117
4.23.1. Responsabilidade para Ficheiros	118
4.23.2. Diferenças de responsabilidade	120
4.24. O Navegador de Repositório	120
4.25. Gráficos de Revisões	123
4.25.1. Nós do Gráfico de Revisões	124
4.25.2. Alterando a Vista	124
4.25.3. Utilizar o Gráfico	126
4.25.4. Refrescar a Vista	127
4.25.5. Podar as Árvores	127
4.26. Exportar uma Cópia de Trabalho do Subversion	127
4.26.1. Remover uma cópia de trabalho do controlo de versões	129
4.27. Reposicionar uma cópia de trabalho	129
4.28. Integração com Sistemas de controlo de problemas/Gestores de Problemas	130
4.28.1. Adicionar Números de Problemas nas Mensagens de Registo	130
4.28.2. Obter Informações do Gestor de Problemas	133
4.29. Integração com visualizadores de repositório Web-based	134
4.30. Preferências do TortoiseSVN	135
4.30.1. Preferências Gerais	135
4.30.2. Preferências do Gráfico de Revisões	143
4.30.3. Preferências de Sobreposição de Ícones	146
4.30.4. Preferências de Rede	149

4.30.5. Preferências de Programas Externos	151
4.30.6. Preferências de Dados Guardados	154
4.30.7. Cache de Registo	155
4.30.8. Scripts de Gancho do Lado do Cliente	159
4.30.9. Preferências do TortoiseBlame	163
4.30.10. Configurações do Registo	163
4.30.11. Pastas de Trabalho do Subversion	165
4.31. Passo Final	165
5. O Programa SubWCRev	166
5.1. A Linha de Comando SubWCRev	166
5.2. Substituição de Palavra-Chave	166
5.3. Exemplo de Palavra-Chave	167
5.4. Interface COM	168
6. Interface IBugtraqProvider	175
6.1. O interface do IBugtraqProvider	175
6.2. A interface IBugtraqProvider2	176
A. Questões Mais Frequentes (FAQ)	179
B. Como Farei Para...	180
B.1. Mover/copiar muitos ficheiros de uma vez só	180
B.2. Forçar utilizadores a introduzir uma mensagem de registo	180
B.2.1. Script-gancho no servidor	180
B.2.2. Propriedades do projecto	181
B.3. Actualizar ficheiros seleccionados a partir do repositório	181
B.4. Reverter (Anular) revisões no repositório	181
B.4.1. Usar a caixa de diálogo registo de revisão	181
B.4.2. Usa a caixa de diálogo integrar	181
B.4.3. Usa o <code>svndumpfilter</code>	182
B.5. Comparar duas revisões de um ficheiro ou pasta	182
B.6. Incluir um subprojecto comum	182
B.6.1. Usa o <code>svn:externals</code>	182
B.6.2. Usar uma cópia de trabalho aninhada.	183
B.6.3. Usa uma localização relativa	183
B.7. Criar um atalho para um repositório	184
B.8. Ignorar ficheiros que já estão versionados	184
B.9. Remover uma cópia de trabalho do controlo de versões	184
B.10. Remover uma cópia de trabalho	184
C. Dicas Úteis para Administradores	185
C.1. Instalar o TortoiseSVN via políticas de grupo	185
C.2. Redireccionar a verificação de actualização	185
C.3. Configurar a variável de ambiente <code>SVN_ASP_DOT_NET_HACK</code>	186
C.4. Desactivar entradas do menu de contexto	186
D. Automatizar o TortoiseSVN	188
D.1. Comandos TortoiseSVN	188
D.2. Comandos TortoiseIDiff	191
E. Referência Cruzada da Interface de Linha de Comandos	193
E.1. Convenções e Regras Básicas	193
E.2. Comandos TortoiseSVN	193
E.2.1. Checkout	193
E.2.2. Actualizar	193
E.2.3. Actualizar para Revisão	194
E.2.4. Submeter	194
E.2.5. Comparar	194
E.2.6. Mostrar Registo	195
E.2.7. Verificar Modificações	195
E.2.8. Gráfico de Revisões	195
E.2.9. Navegador de Repositório	195
E.2.10. Editar Conflitos	196
E.2.11. Resolvido	196

E.2.12. Alterar nome	196
E.2.13. Remover	196
E.2.14. Reverter	196
E.2.15. Limpar	196
E.2.16. Obter "Lock"	196
E.2.17. Libertar "Lock"	197
E.2.18. Ramo/Etiqueta	197
E.2.19. Trocar	197
E.2.20. Integrar	197
E.2.21. Exportar	197
E.2.22. Reposicionar	198
E.2.23. Criar Repositório Aqui	198
E.2.24. Adicionar	198
E.2.25. Importar	198
E.2.26. Responsabilizar	198
E.2.27. Adicionar á list de ítems a ignorar	198
E.2.28. Criar Correção	198
E.2.29. Aplicar Correção	198
F. Detalhes de Implementação	199
F.1. Sobreposição de Ícones	199
G. Protegendo o Svnserve ao usar o SSH	201
G.1. Configuração de um Servidor Linux	201
G.2. Configurar um Servidor Windows	201
G.3. Ferramentas de cliente SSH para uso com o TortoiseSVN	202
G.4. Criação de certificados OpenSSH	202
G.4.1. Criar chaves com o ssh-keygen	202
G.4.2. Criar chaves usando o PuTTYgen	202
G.5. Testar com o PuTTY	202
G.6. Testar SSH com o TortoiseSVN	203
G.7. Variantes da Configuração do SSH	204
Glossário	206
Índice Remissivo	210

Lista de Figuras

2.1. Um sistema típico Cliente/Servidor	5
2.2. O Problema a Evitar	6
2.3. A Solução Bloquear-Modificar-Desbloquear	7
2.4. A solução Copiar-Modificar-Integrar	8
2.5. ...Continuação do Copiar-Modificar-Integrar	9
2.6. O Sistema de Ficheiros do Repositório	11
2.7. O Repositório	13
3.1. O menu TortoiseSVN para pastas não versionadas	16
4.1. O explorador mostrando os ícones sobrepostos	37
4.2. Menu de Contexto de uma pasta sob controlo de versões.	38
4.3. Menu de ficheiro do Explorador para um atalho numa pasta versionada.	39
4.4. Menu arrastar com o botão direito para uma pasta sob controlo de versões.	40
4.5. Caixa de diálogo de autenticação	41
4.6. Caixa de diálogo Importar	42
4.7. A caixa de diálogo Checkout	44
4.8. A Caixa de Diálogo Submeter	47
4.9. A Verificação de Sintaxe na Caixa de Diálogo Submeter	50
4.10. A caixa de diálogo de Progresso, mostrando a submissão em progresso	51
4.11. Caixa de diálogo de progresso mostrando uma actualização finalizada.	52
4.12. O explorador mostrando os ícones sobrepostos	58
4.13. Verificar Modificações	60
4.14. Caixa de diálogo Submeter com Listas de Alterações	63
4.15. A Caixa de Diálogo Registo de Revisão	65
4.16. O Painel de Topo da Caixa de Diálogo Registo de Revisões com Menu de Contexto	66
4.17. Menu de Contexto do Painel Superior para 2 revisões seleccionadas	68
4.18. O Painel de fundo da Caixa de Diálogo Registo com o Menu de Contexto	69
4.19. A Caixa de Diálogo Registo Mostra Rasto das Revisões de Integração	72
4.20. Histograma de Submissões-por-Autor	74
4.21. Gráfico de queijo Submissões-por-Autor	75
4.22. Gráfico de Submissões-por-data	76
4.23. Caixa de diálogo Colocar-se em Fora de Linha	77
4.24. A Caixa de Diálogo Comparar Revisões	80
4.25. O leitor de diferenças de imagem	81
4.26. Menu de contexto do Explorador para ficheiros não versionados	83
4.27. Menu arrastar com o botão direito para uma pasta sob controlo de versões.	84
4.28. Menu de contexto do Explorador para ficheiros não versionados	85
4.29. Menu de contexto do Explorador para ficheiros versionados	87
4.30. Caixa de Diálogo Reverter	90
4.31. Página de propriedades do explorador, aba do Subversion	92
4.32. Página de propriedades do Subversion	93
4.33. Adicionando propriedades	94
4.34. A Caixa de Diálogo de Ramificar/Etiquetar	101
4.35. A Caixa de Diálogo Trocar	103
4.36. O Assistente de Integração - Selecciona o Intervalo de Revisões	105
4.37. O Assistente de Integração - Integração para Reintegrar	107
4.38. O Assistente de Integração - Integração de Árvores	108
4.39. A Caixa de Diálogo Conflitos de Integração	111
4.40. Caixa de Diálogo de reintegração de Integração	112
4.41. A Caixa de Diálogo Bloquear	114
4.42. A Caixa de Diálogo Verificar Alterações	115
4.43. A caixa de diálogo de Criar Correção	116
4.44. A Caixa de Diálogo de Anotar/Responsabilizar	118
4.45. TortoiseBlame	119
4.46. O Navegador de Repositório	121
4.47. O Gráfico de Revisões	123

4.48. A Caixa de Diálogo Exportar-do-URL	128
4.49. A Caixa de Diálogo Reposicionar	129
4.50. Caixa de diálogo de exemplo da consulta ao gestor de problemas	134
4.51. A Caixa de Diálogo Preferências, Página Geral	136
4.52. A Caixa de Diálogo Preferências, Página Menu de Contexto	138
4.53. A Caixa de Diálogo Preferências, Página Diálogos 1	139
4.54. A Caixa de Diálogo Preferências, Página Diálogos 2	141
4.55. A Caixa de Diálogo Preferências, Página de Cores	142
4.56. A Caixa de Diálogo Preferências, Página Gráfico de Revisões	143
4.57. A Caixa de Diálogo preferências, Página Cores do Gráfico de Revisões	144
4.58. A Caixa de Diálogo Preferências, Página Sobreposição de Ícones	146
4.59. A Caixa de Diálogo preferências, Página Conjunto de Ícones	149
4.60. A Caixa de Diálogo Preferências, Página de Rede	150
4.61. A Caixa de Diálogo Preferências, Página Visualizador de Comparação	151
4.62. A Caixa de Diálogo Preferências, Caixa de Diálogo Comparar/Integrar Avançados	153
4.63. A Caixa de Diálogo, Página de Dados Guardados	154
4.64. A Caixa de Diálogo Preferências, Página Cache de Registo	156
4.65. A Caixa de Diálogo Preferências, Estatísticas da Cache de Registo	158
4.66. A Caixa de Diálogo Preferências, Página Scripts de Gancho	159
4.67. A Caixa de Diálogo Preferências, Configurar Scripts de Gancho	160
4.68. A Caixa de Diálogo Preferências, Página Integração com Controlador de Problemas	162
4.69. A Caixa de Diálogo Preferências, Página do TortoiseBlame	163
C.1. A Caixa de Diálogo Actualização	185

Lista de Tabelas

2.1. URLs de Acesso ao Repositório	12
3.1. Configurações do <code>httpd.conf</code> do Apache	29
5.1. Lista de opções de linha de comando disponíveis	166
5.2. Métodos COM/automação suportados	168
C.1. Entradas de menu e seus valores	186
D.1. Lista de comandos e opções disponíveis	189
D.2. Lista de opções disponíveis	191

Prefácio



TortoiseSVN

- Trabalhas numa equipa?
- Já alguma vez te aconteceu, estares a trabalhar num ficheiro e alguém estar a trabalhar ao mesmo tempo no mesmo ficheiro? Perdeste as tuas alterações por causa disso?
- Alguma vez guardaste um ficheiro, e quiseste reverter alterações que efectuaste? Alguma vez desejava ver como estava, algum tempo atrás, determinado ficheiro?
- Alguma vez encontraste um problema no teu projecto e quiseste saber quando foi introduzido esse problema nos teus ficheiros?

Se respondeste “sim” a alguma destas questões, então o TortoiseSVN é pra ti! Continua então a ler para descobrires como pode o TortoiseSVN ajudar-te no teu trabalho. Não é assim tão difícil.

1. Audiência

Este livro é escrito para pessoal letrado em computadores que pretende usar o Subversion para gerir os seus dados, mas sentem-se desconfortáveis a usar o cliente de linha de comandos para o fazer. Visto que o TortoiseSVN é uma extensão da shell do Windows, é assumido que o utilizador está familiarizado com o explorador do Windows e sabe utilizá-lo.

2. Guia de Leitura

Este **Prefácio** explica um pouco sobre o projecto TortoiseSVN, a comunidade de pessoas que trabalham nele, e as condições de licenciamento para o usar e distribuí-lo.

A **Capítulo 1, Introdução** explica o que é o TortoiseSVN, o que faz, de onde vem e o essencial para o instalar no teu PC.

No **Capítulo 2, Conceitos Básicos de Controlo de Versões** nós damos uma curta introdução ao sistema de controlo de versões *Subversion* que está na base do TortoiseSVN. Esta é emprestada da documentação do projecto do Subversion, e explica as diferentes abordagens ao controlo de versões e como funciona o Subversion.

O capítulo no **Capítulo 3, O Repositório** explica como instalar um repositório local, que é útil para testar o Subversion e o TortoiseSVN usando um só PC. Também explica um pouco sobre a administração do repositório, que também é relevante para os repositórios localizados num servidor. Existe também uma secção em como instalar um servidor, se precisares de um.

O **Capítulo 4, Guia de Uso Diário** é o capítulo mais importante, já que explica as principais funcionalidades do TortoiseSVN e como usá-las. Toma a forma de um tutorial, partindo assim do checkout de uma cópia de trabalho, modificá-la, submeter as tuas alterações, etc. Então avança-se para tópicos mais avançados.

O **Capítulo 5, O Programa SubWCRev** é um programa separado e incluído com o TortoiseSVN, que pode extrair informação a partir da tua cópia de trabalho e escrevê-la num ficheiro. Isto é útil para incluir informação de construção (build) nos teus projectos.

A secção **Apêndice B, Como Farei Para...** responde a algumas questões mais comuns, sobre a execução de tarefas que não estão explicitamente cobertas noutros lados.

A secção no [Apêndice D, Automatizar o TortoiseSVN](#) explica como podem ser chamadas a partir da linha de comandos as caixas de diálogo do Interface gráfico (GUI) do TortoiseSVN. Isto é útil para scripting, onde ainda podes necessitar de interacção com o utilizador.

O [Apêndice E, Referência Cruzada da Interface de Linha de Comandos](#) dá uma correlação entre os comandos do TortoiseSVN e os seus equivalentes no cliente de linha de comandos do Subversion `svn.exe`.

3. O TortoiseSVN é grátis!

O TortoiseSVN é grátis. Não precisas de pagar para usá-lo, e podes usá-lo da maneira que bem entenderes. É desenvolvido sob a licença 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. Comunidade

Ambos, TortoiseSVN e Subversion, são desenvolvidos por uma comunidade de pessoas que estão a trabalhar nesses projectos. Eles vêm de vários países de todo o mundo, e trabalham juntos para criar programas fantásticos.

5. Agradecimentos

Tim Kemp

por ter fundado o projecto TortoiseSVN

Stefan Küng

pelo trabalho árduo para tornar o TortoiseSVN no que ele é agora

Lübbe Onken

pelos ícones maravilhosos, logo, caça ao erro, tradução e gestão das traduções

Simon Large

por ajudar com a documentação e caça ao bug

O livro sobre Subversion

pela excelente introdução a Subversion e ao capítulo 2 que copiamos aqui

O projecto de Estilo da Tigris

por alguns dos estilos reutilizados nesta documentação

Os nossos contribuintes

para as correcções, relatórios de bugs e novas ideias, e por ajudar outros ao responder a questões na nossa lista de correio.

Os nossos doadores

por muitas horas de prazer com a musica que nos enviaram

6. Terminologia utilizada neste documento

Para tornar mais fácil a leitura dos documentos, os nomes de todos os écrans e Menus do TortoiseSVN estão escritos com um tipo de letra diferente. Por exemplo a **Caixa de Diálogo Registo**.

O menu escolhido é indicado com uma seta. TortoiseSVN → **Mostrar Registo** significa: selecciona *Mostrar Registo* a partir do menu de contexto *TortoiseSVN*.

Onde aparece um menu de contexto local, no interior de uma das caixas de diálogo do TortoiseSVN, ele é mostrado do seguinte modo: **Menu Contexto** → **Guardar como ...**

Os botões do interface de utilizador são indicados da seguinte maneira: Premir **OK** para continuar.

Acções do Utilizador são indicadas com fonte a negrito. **Alt+A**: premir a tecla **Alt** no teu teclado e enquanto a primes, prime também a tecla **A**. Arrasto-direito: premir o botão direito do rato e enquanto o primes *arrasta* os itens para a nova localização.

A saída do sistema e a entrada do teclado é também indicado com um tipo de letra diferente.



Importante

As notas importantes estão marcadas com um ícone.



Dica

Dicas que te facilitam a vida.



Cuidado

Lugares onde é preciso cuidado com o que fazes.



Atenção

Onde é preciso tomar um cuidado extremo, corrupção de dados ou outras coisas nefastas podem ocorrer se esses avisos forem ignorados.



Capítulo 1. Introdução

O controlo de versões é a arte de gerir modificações na informação. Desde há muito, tem sido uma ferramenta crítica para os programadores que usualmente efectuam pequenas alterações no software e, desfazem ou confirmam algumas dessas alterações no dia seguinte. Imaginem uma equipa desses desenvolvedores a trabalhar concorrentemente - e até talvez em simultâneo nos mesmos ficheiros! - e podes ver o porquê da necessidade de um bom sistema para *gerir o caos potencial*.

1.1. O que é o TortoiseSVN?

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

Alguns sistemas de controlo de versões são também sistemas de software configuration management (SCM). Estes sistemas são especificamente desenhados para gerir arvores de código fonte e têm muitas funcionalidades que são específicas ao desenvolvimento de software; como compreender nativamente as linguagens de programação, ou fornecerem ferramentas para construir (build) software. O Subversion no entanto não é um desses sistemas; é um sistema genérico que pode ser utilizado para gerir *qualquer* colecção de ficheiros, incluindo os de código fonte.

1.2. A história do TortoiseSVN

Em 2002, Tim Kemp achou que o Subversion era um ótimo sistema de controlo de versões mas que carecia de um bom GUI de cliente. A ideia para um cliente do Subversion com integração com a shell do Windows foi inspirada por um cliente similar para o CVS, chamado TortoiseCVS.

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

à medida que o Subversion se tornou mais estável, atraiu mais e mais utilizadores, que também começaram a usar o TrotoiseSVN como o seu cliente SVN. A base de utilizadores cresceu rapidamente (e continua a crescer todos os dias). Foi quando o Lübbe Onken se ofereceu para ajudar com uns ícones porreiros e um logo para o TrotoiseSVN. E ele toma conta do sítio web e gere as traduções.

1.3. Funcionalidades do TortoiseSVN

O que torna o TortoiseSVN um bom cliente Subversion? Aqui vai uma pequena lista de funcionalidades:

Integração na "shell"

O TortoiseSVN integra-se naturalmente na shell do Windows (i.e. o explorador). Isto significa que podes continuar a trabalhar com as ferramentas com que estás familiarizado. E não precisas de mudar para uma aplicação diferente, de cada vez que precisas de funções do controlo de versões!

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

Soberposição de ícones

O estado de cada ficheiro e pasta versionada é indicado através de pequenos ícones sobrepostos. Desta maneira é possível visualizar de imediato o estado da tua cópia de trabalho.

Acceso fácil aos comandos do Subversion

Todos os comandos do Subversion estão disponíveis através do menu de contexto do explorador. O TortoiseSVN adiciona aqui o seu sub-menu.

Sendo o TortoiseSVN um cliente do Subversion, gostaríamos também de mostrar algumas das funcionalidades específicas do Subversion:

Versionamento de Pastas

O CVS só segue o histórico de ficheiros individuais, mas o Subversion implementa um sistema “virtual” de ficheiros versionados que segue as alterações em todo o sistema de ficheiros ao longo do tempo. Ficheiros e pastas são versionadas. Como resultado, existem comandos reais do lado do cliente de, **mover** e **copiar** que actuam em ficheiros e pastas.

Submissões atómicas

Uma submissão para o repositório ou é executada por completo ou não o é de todo. Este comportamento permite aos desenvolvedores construir e submeter alterações para o repositório como blocos lógicos.

Metadata versionada

Cada ficheiro ou pasta possuem um conjunto invisível de “propriedades” agarrados a si. Poderás então inventar e armazenar qualquer conjunto arbitrário de pares chave/valor que desejes. Propriedades são versionadas ao longo do tempo tal como os conteúdos dos ficheiros.

Escolha de camadas de rede

O Subversion tem uma noção abstracta do acesso ao repositório, tornando fácil para as pessoas a implementação de novos mecanismos de rede. O servidor de rede “avançado” do Subversion é um módulo para o servidor web Apache, que fala uma variante do protocolo HTTP, chamado WebDAV/DeltaV. Isto dá ao Subversion uma grande vantagem em estabilidade e interoperabilidade e fornece várias funcionalidades chave gratuitamente: autenticação, autorização, compressão em linha e navegação de repositório, por exemplo. Um processo de servidor Subversion mais pequeno e autónomo é também providenciado. Este servidor fala um protocolo customizado que pode ser facilmente tunelizado através de ssh.

Processamento consistente de dados

O Subversion exprime as diferenças nos ficheiros usando um algoritmo de diferencial binário, que funciona de modo igual tanto para ficheiros de texto (legíveis para humanos) e ficheiros binários (ilegíveis para humanos). Ambos os tipos de ficheiros estão armazenados de igual modo, e comprimidos, no repositório e as diferenças são transmitidas em ambas as direcções através da rede.

Ramificação e etiquetagem eficiente

O custo de ramificação e etiquetagem não precisa de ser proporcional à dimensão do projecto. O Subversion cria ramos e etiquetas através da simples cópia do projecto utilizando um mecanismo semelhante a um hard-link. Sendo assim, estas operações necessitam de apenas de uma pequena e constante fracção de tempo, necessitando de muito pouco espaço no repositório.

Hackabilidade

O Subversion não tem bagagem histórica; está implementado como uma colecção de bibliotecas partilhadas em C com APIs bem definidas. Isto torna o Subversion extremamente manutenível e usável por outras aplicações e linguagens.

1.4. Instalando o TortoiseSVN

1.4.1. Requisitos de sistema

TortoiseSVN corre em Windows 2000 SP2, Windows XP ou superior. Windows 98, Windows ME e Windows NT4, já não são suportados desde o TortoiseSVN 1.2.0, mas ainda podes descarregar versões mais antigas, se realmente necessitas delas.

Se encontrares algum problema durante ou após a instalação do TortoiseSVN consulta por favor, em primeiro lugar, [Apêndice A, Questões Mais Frequentes \(FAQ\)](#).

1.4.2. Instalação

O TortoiseSVN vem com um instalador fácil de usar. Efectua duplo clique no ficheiro de instalação e segue as instruções. O instalador irá tomar conta do resto.



Importante

Necessitas de privilégios de administrador para instalar o TortoiseSVN.

1.4.3. Pacotes de Línguas

O interface gráfico do TortoiseSVN tem sido traduzido em muitas e variadas línguas, sendo então possível descarregar um pacote de línguas que sirva as tuas necessidades. Tu podes encontrar os pacotes de línguas no nosso [translation status página de ponto de situação da tradução](http://tortoisesvn.net/translation_status) [http://tortoisesvn.net/translation_status]. Se não existir um pacote de língua ainda disponível porque não juntares-te á equipa e submeteres a tua própria tradução ;-)

Cada pacote de língua está empacotado como um instalador com extensão .exe. Apenas precisas de correr o ficheiro de instalação e seguir as instruções. Da próxima vez que reiniciares o computador, a tradução estará disponível.

1.4.4. Corretor Ortográfico

O TortoiseSVN inclui um corretor ortográfico que permite verificar as mensagens de log de submissão. Isto é de utilidade especial se a língua do projecto não é a língua nativa. O verificador ortográfico usa os mesmos ficheiros de dicionário que o [OpenOffice](http://openoffice.org) [http://openoffice.org] e [Mozilla](http://mozilla.org) [http://mozilla.org].

O instalador instala automaticamente os dicionários de Inglês UK e US. Se quiseres outras línguas, a opção mais simples é instalar um dos pacotes de línguas do TortoiseSVN. Este irá instalar os ficheiros de dicionário apropriados bem como o interface local do TortoiseSVN. Da próxima vez que reiniciares o computador, o dicionário estará também disponível.

Ou poderás instalar os dicionários manualmente. Se tens o OpenOffice ou Mozilla instalado, podes copiar esses dicionários, que estão localizados nas pastas de instalação dessas aplicações. De outro modo, necessitarás de descarregar os ficheiros de dicionário necessários a partir de <http://wiki.services.openoffice.org/wiki/Dictionary>

Após obteres os ficheiros de dicionário, provavelmente necessitarás de renomeá-los para que os ficheiros só tenham os códigos de localização no nome. Exemplo:

- en_US.aff
- en_US.dic

Então copia-os apenas para a sub-pasta bin da pasta de instalação do TortoiseSVN. Normalmente esta será C:\Program Files\TortoiseSVN\bin. No entanto se não quiseres lixo na sub-pasta bin poderás colocar os teus ficheiros de correção ortográfica na pasta C:\Program Files\TortoiseSVN\Languages. Se essa pasta não existe, terás de a criar primeiro. Da próxima vez que iniciares o TortoiseSVN o corrector ortográfica estará disponível.

Se quiseres instalar múltiplos dicionários, o TortoiseSVN utiliza estas regras para seleccionar o dicionário em uso.

1. Verificar o parametro de configuração `tsvn:projectlanguage`. Ver a secção [Secção 4.17, “Configurações de Projecto”](#) para mais informações sobre configuração das propriedades do projecto
2. Se não é seleccionada nenhuma língua para o projecto, ou essa língua não está instalada, tente por favor a língua local do Windows.
3. Se a configuração local do Windows não funciona, experimente a língua “Base”, i.e. `de_CH` (Alemão-Suíço) cairá por defeito para `de_DE` (German).
4. Se nenhuma das abordagens anteriores funcionou, então a língua por defeito será o Inglês, que está incluído na instalação standard.

Capítulo 2. Conceitos Básicos de Controlo de Versões

Este capítulo é uma versão ligeiramente modificada do mesmo capítulo no livro do Subversion. Uma versão em linha do livro do Subversion pode ser encontrada aqui: <http://svnbook.red-bean.com/>.

Este capítulo é uma introdução curta e casual ao Subversion. Se estás a iniciar-te no controlo de versões, este capítulo é mesmo para ti. Nós vamos começar com um discussão dos conceitos gerais de controlo de versões, abordar ideias específicas por detrás do Subversion, e mostrar alguns exemplos simples do uso do Subversion.

Mesmo que os exemplos deste capítulo mostrem pessoas a partilhar colecções de código fonte de programas, tem em mente que o Subversion pode gerir qualquer tipo de colecções de ficheiros - não está limitado a ajudar os programadores de computadores.

2.1. O Repositório

O Subversion é um sistema centralizado para partilha de informação. No seu cerne está o *repository*, que é um armazém central de dados. O repositório armazena informação na forma de uma *firstterm*>árvore de ficheiros

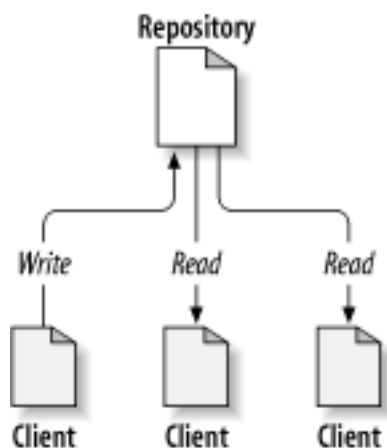


Figura 2.1. Um sistema típico Cliente/Servidor

Então porque será isto interessante? Até agora isto parece-se com a definição de um servidor de ficheiros típico. E de facto, o repositório é uma espécie de servidor de ficheiros, mas não da sua raça normal. O que torna o repositório do Subversion especial é que *ele lembra-se de qualquer alteração* que lhe tenha sido alguma vez escrita: todas as alterações a qualquer ficheiro, e mesmo alterações na própria árvore de pastas, tal como adicionar, remover, e remodelar ficheiros e pastas.

Quando um cliente lê dados do repositório, ele normalmente vê só a última versão da árvore do sistema de ficheiros. Mas o cliente também tem a possibilidade de ver estados *prévios* do sistema de ficheiros. Por exemplo, um cliente efectuar perguntas de histórico, tais como “o que é que esta pasta continha na última Quarta-feira?”, ou “Quem foi a última pessoa a modificar este ficheiro, e que alterações foram efectuadas?” Essas são o tipo de questões que estão no coração de qualquer *sistema de controlo de versões*: sistemas que são desenhados para recordar e guardar alterações em dados ao longo do tempo.

2.2. Modelos de Controlo de Versões

Todos os sistemas de controlo de versões têm de resolver um problema fundamental: como irá o sistema permitir aos utilizadores partilhar informação, evitando ao mesmo tempo que pisem, por acidente, os

pés uns dos outros? É muito fácil os utilizadores escreverem acidentalmente por cima das alterações de outros utilizadores no repositório.

2.2.1. O Problema da Partilha de Ficheiros

Considera o seguinte cenário: supõem que temos dois colegas, Harry e Sally. Cada um decide editar o mesmo ficheiro do repositório, ao mesmo tempo. Se o Harry guardar primeiro as suas alterações no repositório, é possível que (momentos depois) a Sally possa as substituir com a sua versão do ficheiro. Enquanto que a versão do Harry não se perderá definitivamente (porque o sistema recorda-se de cada alteração), qualquer alteração feita pelo Harry *não* estará presente na nova versão do ficheiro da Sally porque, para começar, ela nunca viu as alterações do Harry. O trabalho do Harry estará efectivamente perdido - ou pelo menos ausente da última versão do ficheiro - e provavelmente por acidente. Esta é definitivamente uma situação que queremos evitar!

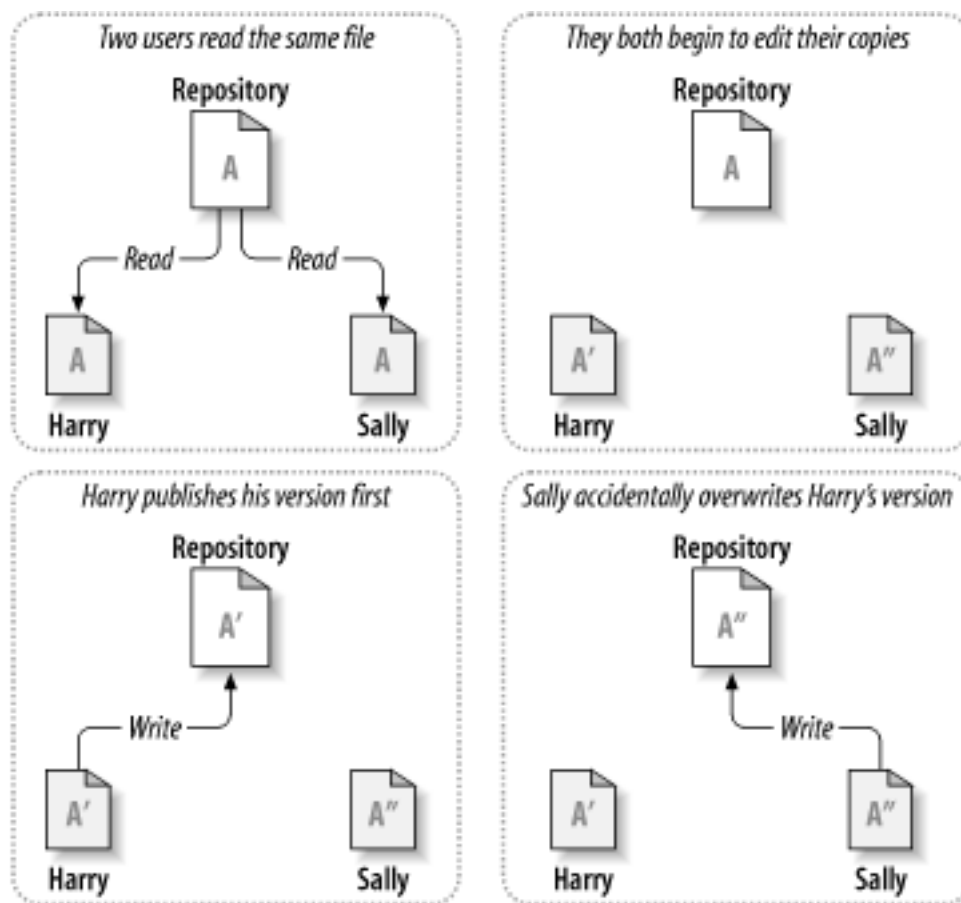


Figura 2.2. O Problema a Evitar

2.2.2. A Solução Bloquear-Modificar-Desbloquear

Muitos sistemas de controlo de versões usam o modelo *bloquear-modificar-desbloquear* para resolver este problema, o que é uma solução muito simples. Em tal sistema, o repositório apenas permite que uma pessoa de cada vez possa modificar um ficheiro. Primeiro o Harry tem de *bloquear* o ficheiro, antes de poder editar o ficheiro. Bloquear um ficheiro é um pouco como requisitar um livro da biblioteca; se o Harry tem o ficheiro bloqueado, então a Sally não pode efectuar alterações nele. Se ela tentar bloquear o ficheiro, o repositório ir-lhe-á negar o pedido. Tudo o que ela pode fazer é apenas ler o ficheiro, e esperar que o Harry termine as suas alterações e liberte o bloqueio. Após o Harry desbloquear o ficheiro, termina a sua vez, e agora a Sally pode ter a sua vez ao bloquear e editar o ficheiro.

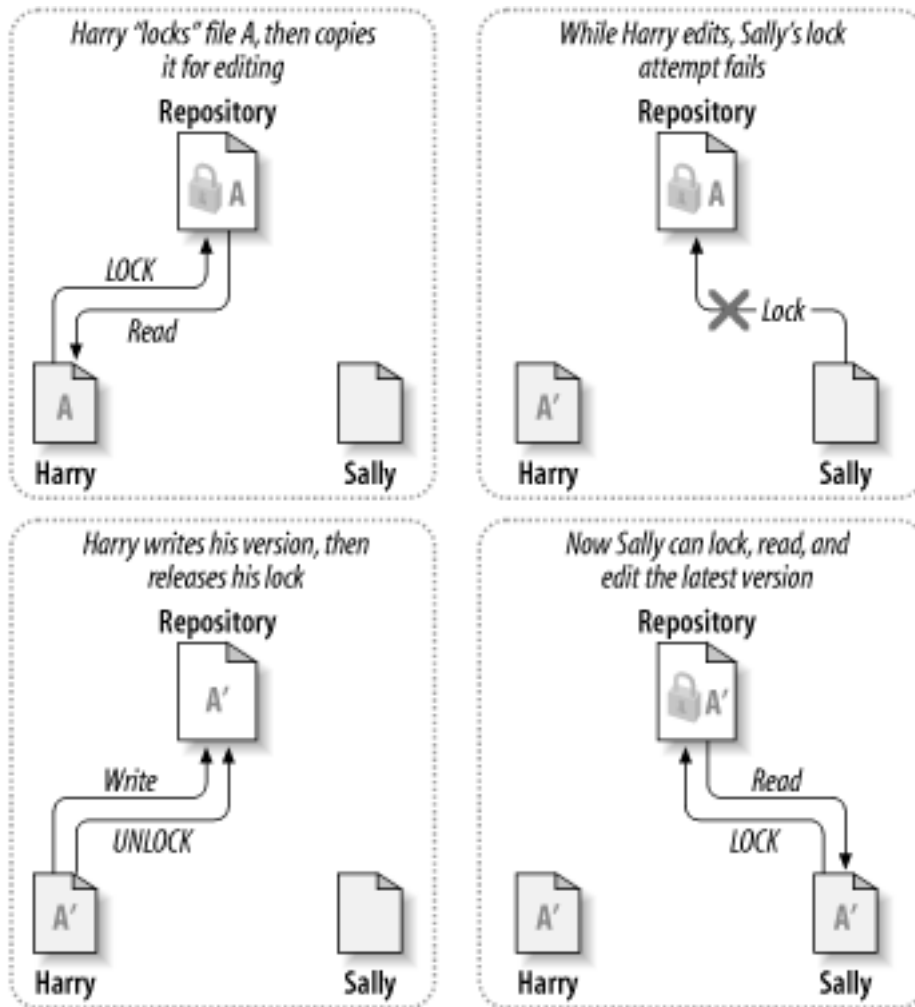


Figura 2.3. A Solução Bloquear-Modificar-Desbloquear

O problema com o modelo bloquear-modificar-desbloquear é que é um pouco restritivo, e torna-se muitas vezes um bloqueio para os utilizadores:

- *Bloquear pode causar problemas administrativos.* Por vezes o Harry irá bloquear um ficheiro e esquecer-se disso. Enquanto, porque a Sally está ainda à espera para editar o ficheiro, tem as mãos atadas. Então o Harry vai de férias. Agora a Sally tem de se dirigir ao administrador para este libertar o bloqueio do Harry. A situação acaba por causar um grande e desnecessário atraso e tempo perdido.
- *Bloquear pode causar serialização desnecessária.* E se o Harry estiver a editar o início de um ficheiro de texto, e a Sally simplesmente quiser editar o final do mesmo ficheiro? Essas alterações não se sobrepõem de todo. Eles poderiam facilmente editar o ficheiro simultaneamente, e nenhum grande mal adviria daí, assumindo que as alterações seriam propriamente integradas em conjunto. Não há nenhuma necessidade deles recorrerem a turnos nesta situação.
- *Bloquear pode criar um falso sentido de segurança.* Imagina que o Harry bloqueia e edita o ficheiro A, enquanto a Sally simultaneamente bloqueia e edita o ficheiro B. Mas supõem que A e B dependem um do outro, e as alterações feitas a cada um deles são semanticamente incompatíveis. De repente A e B já não funcionam juntos. O sistema de bloqueio foi impotente na prevenção deste problema - mas no entanto forneceu um falso sentimento de segurança. É fácil para o Harry e para a Sally imaginar que ao bloquear os ficheiros, cada um estará a iniciar uma tarefa segura e isolada, o que por sua vez os inibe de discutir de antemão as suas alterações incompatíveis.

2.2.3. A solução Copiar-Modificar-Integrar

O Subversion, CVS e outros sistemas de controlo de versões usam o modelo *copiar-modificar-integrar* como alternativa ao bloqueio. Neste modelo cada utilizador cliente lê o repositório e cria uma *cópia de trabalho* pessoal do ficheiro ou projecto. Os utilizadores então trabalham em paralelo, modificando as suas cópias privadas. Finalmente, as cópias privadas são integradas conjuntamente numa versão final. O sistema de controlo de versões assiste frequentemente a integração, mas cabe ao humano ser o responsável último por garantir que este aconteça correctamente.

Eis um exemplo. Digamos que o Harry e a Sally criaram cada um, cópias de trabalho do mesmo projecto, copiadas do repositório. Eles trabalharão concorrentemente e efectuaram alterações no mesmo ficheiro A, em suas cópias. A Sally guarda as suas alterações no repositório, primeiro. Então o Harry tenta guardar as suas alterações mais tarde o repositório informa-o que o seu ficheiro A está *desactualizado*. Noutras palavras, o ficheiro A do repositório foi de algum modo alterado desde a última vez que o copiaste. Então o Harry pede ao seu cliente para *integrar* quaisquer novas alterações do repositório na sua cópia de trabalho do ficheiro A. Há probabilidades que as alterações da Sally não se sobreponham com as suas; pelo que uma vez que ambos os conjuntos de alterações estejam integrados, ele guardará a sua cópia de trabalho de volta para o repositório.

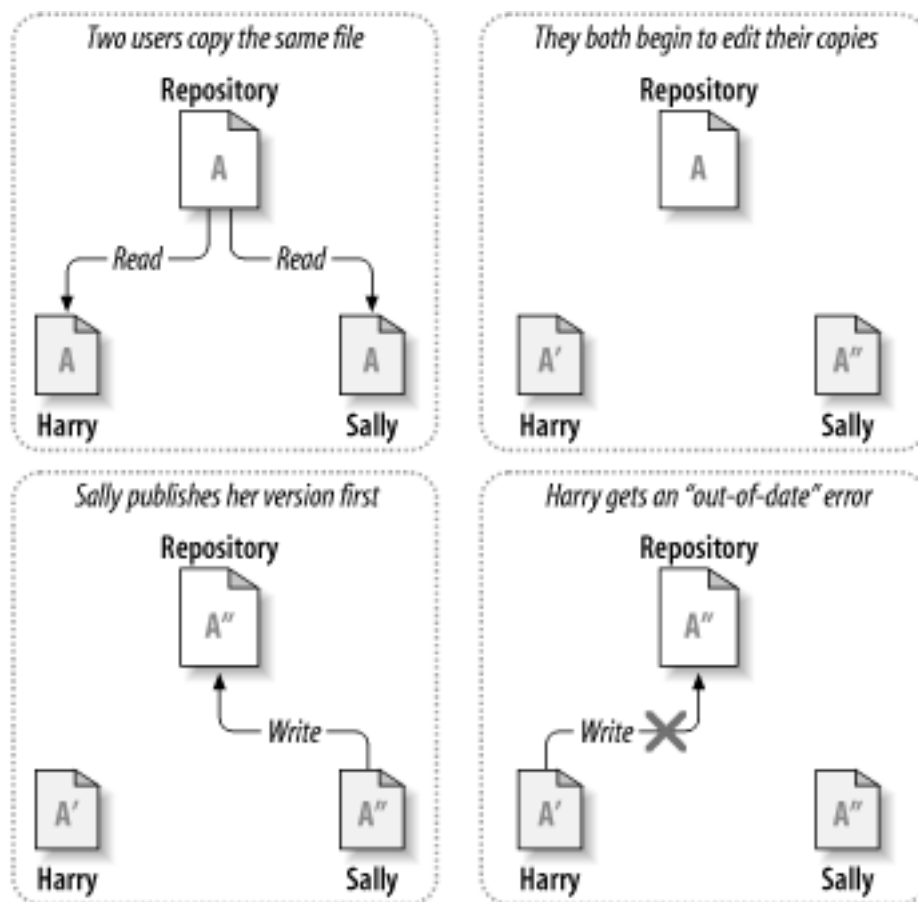


Figura 2.4. A solução Copiar-Modificar-Integrar

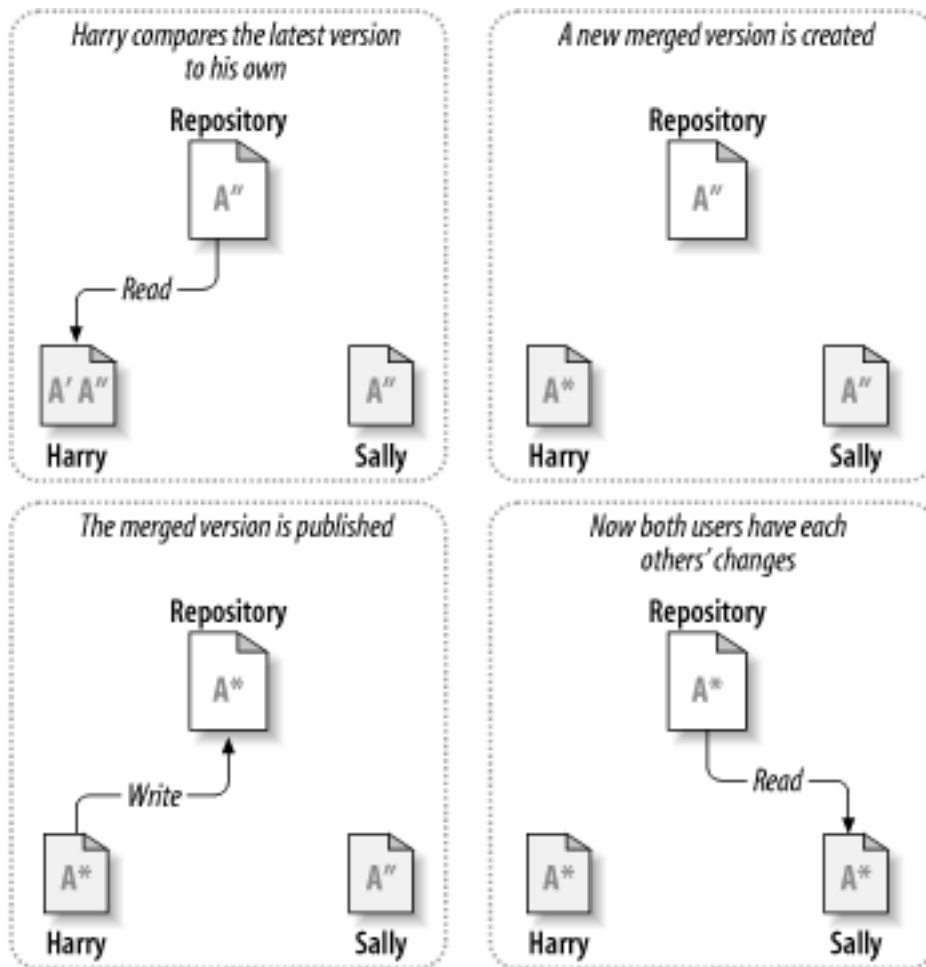


Figura 2.5. ...Continuação do Copiar-Modificar-Integrar

Mas se as alterações da Sally *se* sobrepuserem as do Harry? E depois? Esta situação é chamada de *conflito*, e não é normalmente um grande problema. Quando o Harry pede ao seu cliente para integrar as últimas alterações do repositório para a sua cópia de trabalho, a sua cópia do ficheiro A e de alguma maneira sinalizada como estando em conflito: poderemos observar ambos os conjuntos de alterações, e escolher manualmente entre elas. Ter em atenção que o software não pode resolver automaticamente os conflitos: só os humanos são capazes de compreender e efectuar as necessárias escolhas inteligentes. Uma vez que o Harry tenha resolvido manualmente a sobreposição das alterações (talvez ao discutir o conflito com a Sally), ele pode salvar com segurança o ficheiro integrado de volta para o repositório.

O modelo copiar-modificar-integrar pode soar um pouco caótico, mas na prática funciona de um modo muito fluido. Os utilizadores podem trabalhar em paralelo, nunca esperando uns pelos outros. Quando trabalham nos mesmos ficheiros, acontece que a, a maior parte das suas alterações concorrentes não se sobrepõem; conflitos são raros. E a quantidade de tempo levada a resolver conflitos é muito inferior ao tempo perdido com o sistema de bloqueio

No final tudo se resume a um factor crítico: comunicação com o utilizador. Quando os utilizadores não comunicam bem, aumentam os conflitos sintácticos e semânticos. Nenhum sistema pode forçar os utilizadores a comunicar na perfeição, e nenhum sistema pode detectar conflitos semânticos. Portanto não existe vantagens em ser iludido por uma falsa promessa de que o sistema de bloqueio irá evitar conflitos; na prática os sistemas bloqueantes parecem inibir mais a produtividade que qualquer outra coisa.

Existe uma situação particular onde o modelo bloquear-modificar-desbloquear funciona melhor, que é quando tens ficheiros não integráveis. Por exemplo, se o teu repositório contém algumas imagens, e duas

peessoas alteram a imagem ao mesmo tempo, não existe forma de integrar essas alterações. Ou o Harry ou a Sally irão perder as suas alterações.

2.2.4. O que faz o Subversion?

O Subversion usa por defeito a solução copiar-modificar-integrar, e em muitos casos, isto será tudo o que alguma vez irás precisar. No entanto, tal como na versão 1.2, o Subversion também suporta o bloqueio de ficheiros, pelo que se tiveres ficheiros não integráveis, ou simplesmente, fores forçado pela administração a usar uma política de bloqueio, o Subversion poderá ainda te fornecer as funcionalidades de que necessitas.

2.3. O Subversion em Acção

2.3.1. Cópias de Trabalho.

Tu já ouviste falar em cópias de trabalho; agora vamos demonstrar como o cliente Subversion as cria e usa.

Uma cópia de trabalho do Subversion é uma normal árvore de pastas no teu sistema local contendo uma colecção de ficheiros. Tu podes editar esses ficheiros como desejares, e no caso de serem ficheiros de código fonte, poderás compilar o teu programa a partir deles da forma habitual. A tua cópia de trabalho é a tua área de trabalho privada: o Subversion nunca irá incorporar alterações de outras pessoas, nem tornar as tuas alterações disponíveis a outros, até tu explicitamente lhe indicares para o fazer.

Após efectuares algumas alterações nos ficheiros da tua cópia de trabalho, e verificares que elas estão correctas, o Subversion ir-te-á fornecer comandos para *publicares* as tuas alterações para as outras pessoas que trabalham contigo no projecto (escrevendo para o repositório). Se outras pessoas publicarem as suas próprias alterações, o Subversion fornece-te comandos para integrar essas alterações na tua cópia de trabalho (ao ler do repositório).

A cópia de trabalho também contém alguns ficheiros extra, criados e mantidos pelo Subversion para ajudá-lo a executar esses comandos. Em particular, cada pasta na tua cópia de trabalho contém uma subpasta chamada `.svn`, também conhecida como a *pasta administrativa* da cópia de trabalho. Os ficheiros em cada pasta administrativa ajudam o Subversion a reconhecer quais os ficheiros que contêm alterações não publicadas, e quais os ficheiros que estão desactualizados em relação ao trabalho dos outros.

Um repositório típico do Subversion armazena com frequência os ficheiros (ou código fonte) de vários projectos; normalmente, cada projecto é uma subpasta no sistema de ficheiros do repositório. Nesta configuração, a cópia de trabalho do utilizador corresponde normalmente a uma sub-árvore em particular do repositório.

Por exemplo, supõem que tens um repositório que contém dois projectos de software.

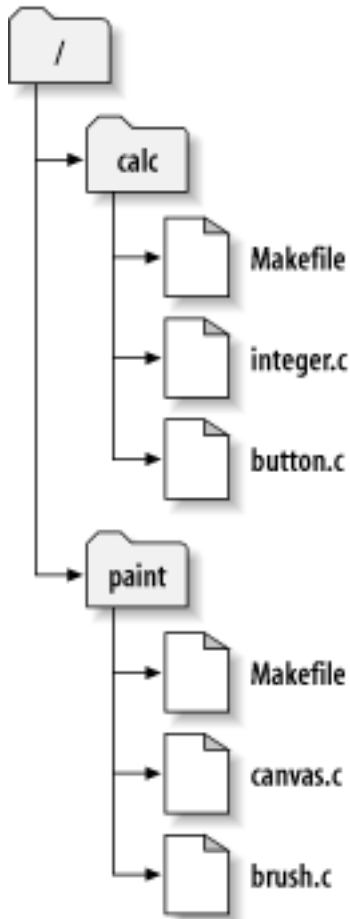


Figura 2.6. O Sistema de Ficheiros do Repositório

Noutras palavras, a pasta raiz do repositório terá duas subpastas: `paint` e `calc`.

Para obteres uma cópia de trabalho, terás de efectuar *checkout* nalgumas pastas do repositório. (O termo *emphasis>*checkout

Supõem que tu efectuas alterações no `button.c`. Visto que a pasta `.svn` lembra-se da data das modificações do ficheiro e do seu conteúdo original, pode assim o Subversion saber se alteraste o ficheiro. No entanto o Subversion não tornará as tuas alterações públicas até explicitamente tu lhe disseres para o fazer. O acto de publicar as tuas alterações é mais usualmente conhecido por *submiter* (ou *checking in*) as alterações para o repositório.

Para publicar as tuas alterações para outros, podes usar o comando **submiter** do Subversion.

Agora as tuas alterações no `button.c` foram submetidas para o repositório; se outro utilizador efectuar um *checkout* de uma cópia de trabalho do `/calc`, irá ver as tuas alterações na última versão do ficheiro.

Supõem que tens um colaborador, Sally, que efectuou *checkout* a uma cópia de trabalho do `/calc` ao mesmo tempo que tu. Quando tu submeteres a tua alteração ao `button.c`, a cópia de trabalho da Sally irá ficar inalterada; o Subversion só modifica as cópias de trabalho a pedido do utilizador.

Para actualizar o seu projecto, a Sally pode pedir ao Subversion para *actualizar* a sua cópia de trabalho, usando o comando **actualizar** do Subversion. Isto irá incorporar as tuas alterações na sua cópia de trabalho, tal como quaisquer outras que tenham sido submetidas desde que a criou (*checkout*).

Note-se que a Sally não precisou de especificar quais os ficheiros a actualizar; o Subversion usa a informação na pasta `.svn`, e informação posterior no repositório, para decidir que ficheiros precisam de ser actualizados.

2.3.2. URLs do Repositório

Os repositórios do Subversion podem ser acedidos através de vários métodos diferentes - no disco local, ou através de vários protocolos de rede. No entanto, a localização do repositório será sempre um URL. O esquema do URL indica o método de acesso:

Esquema	Método de Acesso
file://	Acesso directo ao repositório, numa unidade de rede ou local.
http://	Acesso através do protocolo WebDAV a um servidor Apache consciente do Subversion.
https://	O mesmo que http://, mas com encriptação SSL.
svn://	Acesso TCP/IP não autorizado, via um protocolo personalizado, a um servidor svnservice.
svn+ssh://	Acesso TCP/IP autenticado e encriptado, através de protocolo personalizado a um servidor svnservice.

Tabela 2.1. URLs de Acesso ao Repositório

Na maioria dos casos, os URLs do Subversion usam a sintaxe padrão, permitindo a especificação de nomes de servidor e números de porto, como parte do URL. O método de acesso `file://` é normalmente usado para acessos locais, embora possa ser usado em caminhos UNC para um hospedeiro de rede. Portanto o URL toma a forma de `file://hostname/path/to/repos`. Para a máquina local, a parcela do URL `hostname` pode ser `localhost` ou ausente. Por esta razão os caminhos locais aparecem normalmente com três barras, `file:///path/to/repos`.

Igualmente, os utilizadores do esquema `file://` nas plataformas Windows, irão precisar de usar uma sintaxe “padrão” não oficial para aceder aos repositórios que estão na mesma máquina, mas numa unidade diferente da unidade de trabalho do utilizador. Qualquer uma das duas seguintes sintaxes de caminhos URL irão funcionar aqui, X é a unidade onde reside o repositório:

```
file:///X:/path/to/repos
...
file:///X|/path/to/repos
...
```

Observa que o URL usa barras comuns, mesmo que a forma nativa de um caminho (sem-URL) no Windows utilize barras invertidas.

Podes aceder com segurança a um repositório FSFS através de uma partilha de rede, mas *não poderás* aceder desta forma a um repositório BDB.



Atenção

Não cries ou acedas a um repositório Berkeley DB numa partilha de rede. Ele *não poderá* existir num sistema de ficheiros remoto. Nem mesmo se tiveres a unidade de rede mapeada com uma letra de unidade. Se tentares usar o Berkeley DB numa partilha de rede, os resultados serão imprevisíveis - poderás observar de imediato erros misteriosos, ou poderá levar-te meses até descobrires que a tua base de dados do repositório está subtilmente corrompida.

2.3.3. Revisões

Uma operação **svn submeter** pode publicar alterações de qualquer número de ficheiros e pastas, como uma transacção atómica. Na tua cópia de trabalho, tu podes alterar o conteúdo dos ficheiros, criar,

remover, alterar o nome e copiar ficheiros e pastas, e então submeter o conjunto completo de alterações como uma unidade.

No repositório, cada submissão é tratada como uma transacção atómica: ou todas as alterações da submissão ocorrem, ou nenhuma ocorrerá. O Subversion mantém a sua atomicidade em caso de estouro do programa, problemas de rede, ou outras acções dos utilizadores.

Cada vez que o repositório aceita uma submissão, este cria um novo estado da árvore do sistema de ficheiros chamado *revisão*. A cada revisão é atribuído um número natural, uma unidade acima à da revisão prévia. A revisão inicial de um repositório recém-criado é zero, e consiste em nada mais que uma pasta raiz vazia.

Uma maneira agradável de visualizar o repositório é como uma série de árvores. Imagina um array de números de revisão, a começar no 0, alinhados da esquerda para a direita. Cada número de revisão tem uma árvore de revisão por baixo, e cada árvore é uma "fotografia" do aspecto do repositório após cada submissão.

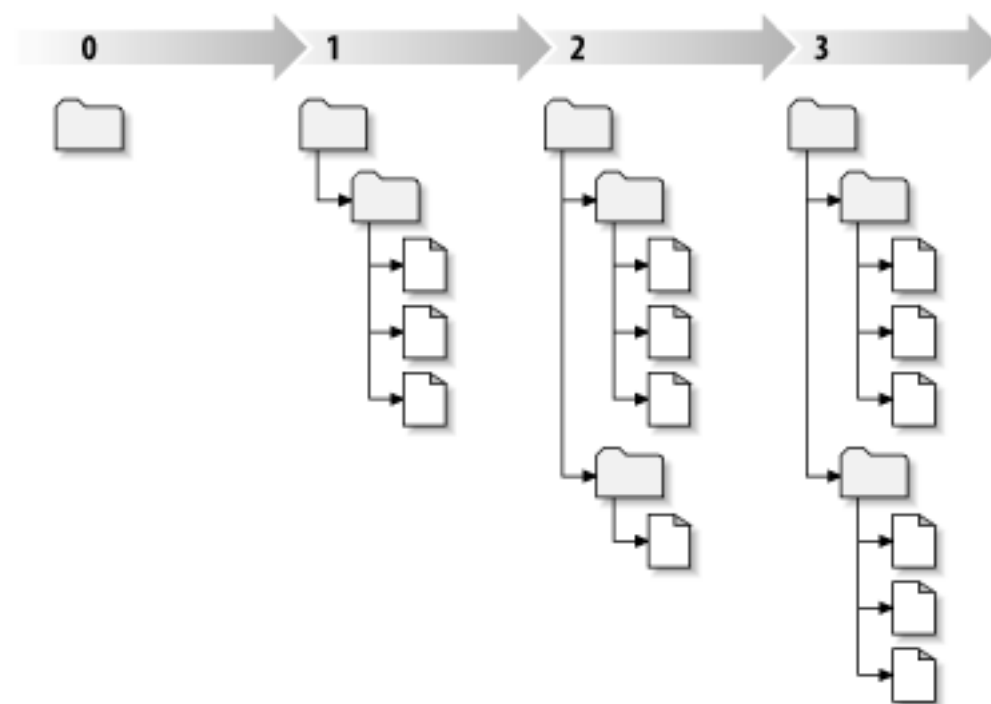


Figura 2.7. O Repositório

Números de Revisão Globais

Ao contrário de muitos outros sistemas de controlo de versões, os números de revisão do Subversion aplicam-se a *árvores completas* e não a ficheiros individuais. Cada número de revisão selecciona uma árvore inteira, um estado particular do repositório após alguma alteração submetida. Outra maneira de pensar sobre isso é que, a revisão N representa o estado do sistema de ficheiros do repositório após o N^o submissão. Quando um utilizador do Subversion fala sobre a "revisão 5 do foo.c", o que ele realmente quer dizer é, "foo.c como ele estava na revisão 5 do repositório". Observa que em geral as revisões N e M de um ficheiro *não* precisam necessariamente de diferir!

É importante observar que as cópias de trabalho nem sempre correspondem a um única revisão do repositório; Elas podem conter ficheiros de várias revisões diferentes. Por exemplo, supõe que efectuas o checkout de um repositório para uma cópia de trabalho, cuja revisão mais recente é 4:

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

Neste momento, esta cópia de trabalho corresponde exactamente à revisão 4 do repositório. No entanto, supõe que fizeste uma alteração ao `button.c`, e submetes essa alteração. Assumindo que não foram efectuadas outras submissões, a tua submissão irá criar a revisão 5, e a tua cópia de trabalho irá ficar assim:

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

Supõe que, neste momento, a Sally submete uma alteração ao `integer.c`, criando a revisão 6. Se usares o **svn actualizar** para actualizares a tua cópia de trabalho, ela irá ficar do seguinte modo:

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

As alterações da Sally ao `integer.c` irão aparecer na tua cópia de trabalho e a tua alteração no `button.c` ainda estará presente. Neste exemplo, o texto do `Makefile` será idêntico nas revisões 4,5,6 mas o Subversion irá marcar a tua cópia de trabalho do `Makefile` com a revisão 6, para indicar que esta é ainda a corrente. Então, após uma actualização de limpeza no topo da tua cópia de trabalho, esta irá corresponder exactamente a uma revisão do repositório.

2.3.4. Como as Cópias de Trabalho Seguem o Repositório

Para cada ficheiro na pasta de trabalho, o Subversion guarda duas partes essenciais de informação na área administrativa `.svn/`:

- em que revisão é baseada o teu ficheiro de trabalho (esta é chamada a *revisão de trabalho* do ficheiro), e
- uma etiqueta temporal que grava quando foi actualizada, pelo repositório, a cópia local pela última vez.

Dada esta informação, comunicando com o repositório, o Subversion pode dizer em qual dos seguintes quatro estados está o ficheiro de trabalho:

Inalterado, e actual

O ficheiro está inalterado na pasta de trabalho, e nenhuma alteração a esse ficheiro foi submetida no repositório, desde a revisão de trabalho. O **submeter** do ficheiro não fará nada, e uma **actualização** ao mesmo também nada fará.

Modificado localmente, e actual

O ficheiro foi alterado na pasta de trabalho, e nenhuma alteração a esse ficheiro foi submetida para o repositório desde a sua revisão base. Existem alterações locais que não foram ainda submetidas para o repositório, pelo que **submeter** do ficheiro irá ter sucesso na publicação das tuas alterações, e um **actualizar** do ficheiro não irá fazer nada.

Inalterado, e desactualizado

O ficheiro não foi alterado na pasta de trabalho, mas foi alterado no repositório. O ficheiro deverá eventualmente ser actualizado, para o tornar actual com a revisão pública. O **submeter** do ficheiro não irá fazer nada, e um **actualizar** do ficheiro irá trazer as últimas alterações para a tua cópia de trabalho.

Modificado localmente, e desactualizado

O ficheiro foi alterado na pasta de trabalho e no repositório. O **submeter** do ficheiro irá falhar com o erro *desactualizado*. O ficheiro deverá ser primeiro actualizado; o comando **actualizar** irá tentar integrar as alterações públicas com as alterações locais. Se o Subversion não conseguir completar

automaticamente a integração, de um modo plausível, irá deixar para o utilizador a resolução do conflito.

2.4. Resumo

Neste capítulo cobrimos um número de conceitos fundamentais do Subversion:

- Introduzimos as noções de repositório central, cópia de trabalho do cliente, e o array de árvores de revisão do repositório.
- Vimos alguns exemplos simples em como dois colaboradores podem usar o Subversion para publicar e receber alterações um do outro, usando o modelo "copiar-modificar-integrar.
- Falámos um pouco sobre o modo como o Subversion segue e gere a informação na cópia de trabalho.

Capítulo 3. O Repositório

Independentemente do protocolo que usas para aceder aos teus repositórios, precisas sempre de criar pelo menos um. Isto pode ser feito com a linha de comandos do Subversion ou com o TortoiseSVN.

Se ainda não criaste um repositório SVN, está é a altura para o fazer.

3.1. Criação do Repositório

Podes criar um repositório com o formato FSFS ou com o mais antigo Berkeley Database (BDB). O formato FSFS é geralmente mais rápido e fácil de administrar, e funciona em partilhas de rede e com o Windows98 sem problemas. O formato BDB já foi considerado mais estável, simplesmente porque estava em uso à mais tempo, mas desde que o FSFS tem estado em uso no campo por vários anos, esse argumento é agora um pouco fraco. Consulta *Choosing a Data Store* [<http://svnbook.red-bean.com/en/1.5/svn.reposadmin.planning.html#svn.reposadmin.basics.backends>] no livro do Subversion, para mais informações.

3.1.1. Criar um Repositório com o Cliente de Linha de Comandos

1. Cria uma pasta vazia com o nome SVN (e.g. D:\SVN\), que será usada como raiz para todos os teus repositórios.
2. Cria outra pasta MyNewRepository dentro da D:\SVN\.
3. Abre a linha de comandos (ou caixa do DOS), muda para a D:\SVN\ e digita

```
svnadmin create --fs-type bdb MyNewRepository
```

or

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

Agora tens um novo repositório localizado em D:\SVN\MyNewRepository.

3.1.2. Criar O Repositório Com o TortoiseSVN

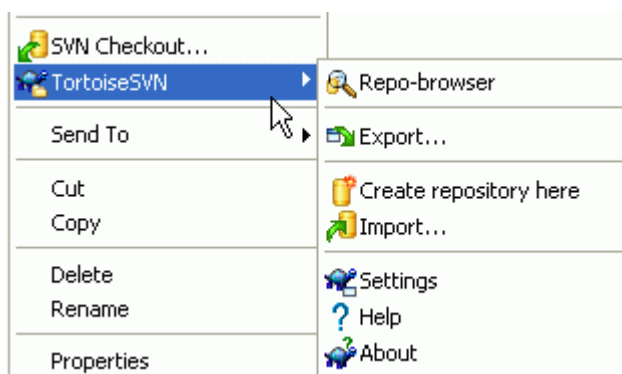


Figura 3.1. O menu TortoiseSVN para pastas não versionadas

1. Abre o explorador do Windows
2. Cria uma nova pasta e chama-a, e.g. SVNRepository

3. Clica com o botão direito na pasta recém-criada e selecciona TortoiseSVN → Criar repositório aqui....

É então criado o repositório dentro da nova pasta. *Não edites esses ficheiros manualmente!!!*. Se obtiveres algum erro, tem a certeza que essa pasta é vazia e não está protegida contra escrita.



Dica

O TortoiseSVN não oferece mais a opção para criar repositórios BDB, mesmo que tu ainda uses a linha de comandos para os criar. Os repositórios FSFS são na generalidade mais fáceis para tu os maneres, o que também nos torna mais fácil manter o TortoiseSVN, devido a problemas de compatibilidade entre diferentes versões do BDB.

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

É claro que nós também recomendamos que tu não uses, de modo algum, o acesso `file://` para além do uso para testes locais. Usar um servidor é muito mais seguro e mais confiável para todos os usos, excepto o desenvolvimento a solo.

3.1.3. Acesso Local ao Repositório

Para aceder ao teu repositório local, precisas do caminho para essa pasta. Lembra-te que o Subversion está à espera que todos os caminhos estejam na forma `file:///C:/SVNRepository/`. Observa o uso de barras por todo o lado.

Para aceder a um repositório localizado numa partilha de rede, podes usar o mapeamento de unidades, ou podes usar um caminho UNC. Para caminhos UNC a forma é `file://ServerName/path/to/repos/`. Observa que só existem aqui 2 barras dianteiras.

Antes do SVN 1.2, os caminhos UNC tinham de ser especificados numa forma mais obscura, `file:///\\ServerName/path/to/repos`. Esta forma é ainda suportada, mas não recomendada.



Atenção

Não criar ou aceder a um repositório Berkeley DB numa partilha de rede. Ele *não pode* existir num sistema de ficheiros remoto. Nem mesmo se tu tens uma unidade de rede mapeada numa letra de unidade. Se tentares usar um Berkeley DB numa partilha de rede, os resultados serão inesperados - tu poderás ver de imediato erros misteriosos, ou poderão passar meses antes de descobrires que a tua base de dados do teu repositório está subtilmente corrompida.

3.1.4. Aceder ao Repositório numa Partilha de Rede

Embora em teoria é possível por um repositório FSFS numa partilha de rede, e ter múltiplos utilizadores a acedê-lo usando o protocolo `file://`, mas isto *não* é definitivamente recomendável.

Em primeiro lugar, tu estás a dar a cada utilizador o acesso directo ao repositório, pelo que qualquer utilizador pode acidentalmente apagar todo o repositório ou inutilizá-lo de alguma forma.

Em segundo lugar, nem todos os protocolos de partilha de ficheiros em rede suportam o bloqueio que o Subversion requer, pelo que podes encontrar o teu repositório corrompido. Poderá não acontecer de imediato, mas um dia dois utilizadores tentarão aceder ao repositório ao mesmo tempo.

Em terceiro lugar, as permissões dos ficheiros tem de ser configuradas como estão. Poderás, no entanto, quase dar a volta a isso numa partilha Windows nativa, mas será particularmente difícil no SAMBA.

O acesso `file://` é dirigido para acesso local e de utilizador único, em particular para testes e depuração de problemas. Quando tu queres partilhar o repositório, precisarás *realmente* de instalar correctamente um servidor, e não é tão difícil como pensas. Consulta o [Secção 3.5, “Aceder ao Repositório”](#) para guias orientadoras na escolha e instalação de um servidor.

3.1.5. Estrutura do Repositório

Antes de tu importares os teus dados para o repositório, deverás primeiro pensar em como o queres organizar. Se tu usares uma das estruturas recomendadas, mais tarde terás a tudo muito mais facilitado.

Existem alguns métodos padrão e recomendados de organizar o repositório. A maioria das pessoas a pasta `trunk` para guardar a “linha principal” de desenvolvimento, uma pasta `ramos` para conter as cópias de ramos, e a pasta `etiquetas` para guardas as cópias etiquetas. Se o repositório tiver só um projecto, então frequentemente as pessoas criaram essas pastas de nível de topo:

```
/trunk
/ramos
/etiquetas
```

Se o repositório conter vários projectos, as pessoas normalmente indexarão a sua estrutura por ramo:

```
/trunk/paint
/trunk/calc
/ramos/paint
/ramos/calc
/etiquetas/paint
/etiquetas/calc
```

...ou por projecto:

```
/paint/trunk
/paint/ramos
/paint/etiquetas
/calc/trunk
/calc/ramos
/calc/etiquetas
```

Indexar por projecto faz sentido, se os projectos não estão intimamente relacionados, e cada um é obtido individualmente por checkout. Para projectos relacionados onde poderás querer efectuar o checkout de todos os projectos de uma só vez, ou onde os projectos estão todos ligados num único pacote de distribuição. É normalmente melhor indexar por ramo. Deste modo tu podes ter um trunk para efectuar checkout e as relações entre subprojectos são mais facilmente visível.

Se adoptares por uma abordagem de um nível de topo `/trunk /tags /branches`, não é necessário referir que terás de copiar o trunk inteiro para cada ramo e etiqueta, e de algum modo esta estrutura oferece a maior flexibilidade.

Para projectos não relacionados, poderás preferir o uso de repositórios separados. Quando submetes as alterações, é o número de revisão do repositório inteiro que muda, e não o número de revisão do projecto. Ter 2 projectos não relacionados a partilhar o repositório, pode significar grandes falhas nos números de revisões. Os projectos Subversion e TortoiseSVN aparecem no mesmo endereço de hospedeiro, mas estão completamente separados, permitindo desenvolvimentos separados, e sem confusão sobre números de construção.

Está claro que és livre para ignorar essas estruturas comuns. Tu podes criar qualquer género de variação, e o que funcionar melhor para ti e para a tua equipa. Lembra-te que o que escolheres não será um comprometimento permanente. Tu podes reorganizar, em qualquer altura, o teu repositório. Porque ramos e etiquetas são pastas comuns, o TortoiseSVN pode mover ou alterar o seu nome, quando assim o desejares.

Trocar de uma estrutura para outra é só uma questão de executar uma série de movimentos do lado do servidor. Se não gostares do modo como as coisas estão organizadas no repositório, poderás manipular as pastas a teu modo.

Então, se ainda não criaste a estrutura de pastas básica dentro do teu repositório, deverás fazê-lo agora. Existem duas maneiras para o obteres. Se queres simplesmente criar a estrutura `/trunk /tags /branches`, tu podes usar para isso o navegador de repositório, para criar as três pastas (em três submissões separadas). Se queres criar uma hierarquia mais profunda, então é mais simples criar primeiro a estrutura de pastas em disco e importá-la numa simples submissão, assim:

1. cria uma nova pasta vazia no teu disco rígido
2. cria a estrutura de pastas de nível de topo que desejas, dentro dessa pasta - não ponhas lá ainda nenhum ficheiro!
3. import this structure into the repository via a right click on the folder and selecting TortoiseSVN → Import... This will import your temp folder into the repository root to create the basic repository layout.

Tem em conta que o nome da pasta que estás a importar não aparecerá no repositório, só o seu conteúdo. Por exemplo, cria a seguinte estrutura de pastas:

```
C:\Temp\New\trunk
C:\Temp\New\ramos
C:\Temp\New\etiquetas
```

Importa o `C:\Temp\New` para a raiz do repositório, que então terá o seguinte aspecto:

```
/trunk
/ramos
/etiquetas
```

3.2. Cópia de Segurança do Repositório

Qualquer que seja o tipo de repositório que uses é de importância vital manter cópias de segurança regulares, e que também as verifiques. Se o servidor falhar poderás ter acesso a uma versão recente dos teus ficheiros, mas sem o repositório, todo o teu histórico será perdido para sempre.

O método mais simples (mas não recomendado) é copiar apenas a pasta do repositório para um meio de reserva. No entanto, tens de ter a certeza absoluta que não existe nenhum processo a aceder aos dados. Neste contexto, acesso significa *qualquer* tipo de acesso. Um repositório BDB é escrito mesmo quando a operação parece apenas requerer leitura, como obter o estado. Se o teu repositório estiver a ser acedido de qualquer modo, durante a cópia (navegador web aberto, WebSVN, etc.), a cópia de segurança ficará inutilizada.

O método recomendado é correr

```
svnadmin hotcopy path/to/repository path/to/backup --clean-logs
```

para criar uma cópia, de maneira segura, do teu repositório. Então efectua a cópia de segurança da cópia. A opção `--clean-logs` não é necessária, mas remove quaisquer ficheiros de registo redundantes, quando efectuas a cópia de segurança de um repositório BDB, o que pode poupar algum espaço.

A ferramenta `svnadmin` é automaticamente instalada quando tu instalas o cliente de linha de comandos do Subversion. Se estás a instalar as ferramentas de linha de comandos num PC Windows, o melhor método é descarregar a versão com instalador Windows. É comprimido de modo mais eficiente do que a versão `.zip`, pelo que a descarga é mais pequena, e toma conta da configuração dos caminhos por ti. Tu podes descarregar a última versão do cliente de linha de comandos do Subversion a partir de <http://subversion.apache.org/getting.html>.

3.3. Scripts de gancho de servidor

Um script de gancho é um programa disparado por algum evento no servidor, como a criação de uma nova revisão ou a modificação de uma propriedade não versionada. A cada gancho é dada informação suficiente para poder distinguir qual o evento, o alvo(s) da operação, e o nome de utilizador da pessoa que disparou o evento. Dependendo do produto dos ganchos ou estado de retorno, o programa gancho pode continua a acção, pará-la, ou suspendê-la de algum modo. Consulta ao capítulo *Hook Scripts* [<http://svnbook.red-bean.com/en/1.5/svn.reposadmin.create.html#svn.reposadmin.create.hooks>] no livro do Subversion para todos os detalhes sobre os ganchos que estão implementados.

Esses scripts de gancho são executados pelo servidor que hospeda o repositório. O TortoiseSVN também permite-te configurar scripts de gancho de cliente, que são executados localmente devido a certos eventos. Consultar [Secção 4.30.8, “Scripts de Gancho do Lado do Cliente”](#) para mais informações.

Amostras de scripts de gancho podem ser encontrados na pasta `hooks` do repositório. Essas amostras de scripts são adequados para servidores Unix/Linux, mas precisam de ser modificados se o teu servidor é Windows. O gancho pode ser um ficheiro batch ou um executável. A amostra abaixo mostra um ficheiro batch que poderá ser usado para implementar um gancho `pre-revprop-change`.

```
rem Only allow log messages to be changed.
if "%4" == "svn:log" exit 0
echo Property '%4' cannot be changed >&2
exit 1
```

Podes observar que tudo o que for enviado para o `stdout` é descartado. Se tu queres que uma mensagem apareça na caixa de diálogo Rejeição de Submissão, deverás enviá-la para o `stderr`. Num ficheiro batch isto é obtido através do uso de `>&2`

3.4. Ligações de Checkout

Se quiseres tornar o teu repositório Subversion disponível para outros, poderás querer incluir uma ligação para ele a partir do teu sítio web. Uma maneira de tornar isto mais acessível é incluir uma *ligação de checkout* para outros utilizadores TortoiseSVN.

Quando instalas o TortoiseSVN, este regista um novo protocolo `tsvn:`. Quando o utilizador do TortoiseSVN clica em tal ligação, uma caixa de diálogo de checkout ir-se-á abrir automaticamente, com o URL do repositório já preenchido.

Para incluíres tal ligação na tua própria página de html, necessitas de adiciona o seguinte código:

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

É claro que terá melhor aspecto se incluíres uma imagem adequada. Poderás usar o *logo TortoiseSVN* [<http://tortoisesvn.tigris.org/images/TortoiseCheckout.png>] ou podes providenciar a tua própria imagem.

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

```
<img src=TortoiseCheckout.png></a>
```

Também podes fazer com que a ligação aponte para uma revisão específica, por exemplo

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

3.5. Aceder ao Repositório

Para usar o TortoiseSVN (ou qualquer outro cliente Subversion) necessitas de um local para os repositórios. Podes armazená-los localmente e aceder a eles usando o protocolo `file://`, ou podes colocá-los num servidor e aceder a eles com os protocolos `literal` ou `http://`

Se estiveres a usar um serviço público de alojamento como o *Google Code* [<http://code.google.com/hosting/>] ou o teu servidor já foi configurado por alguém, então não precisas de fazer mais nada. Segue directamente para [Capítulo 4, Guia de Uso Diário](#).

Se não tens um servidor e trabalhas a solo, ou se estás a avaliar o Subversion e o TortoiseSVN isoladamente, então os repositórios locais serão, provavelmente, a tua melhor escolha. Cria apenas um repositório no teu PC, como descrito anteriormente em [Capítulo 3, O Repositório](#). Podes saltar o resto deste capítulo e ir directamente para o [Capítulo 4, Guia de Uso Diário](#) para descobrires como começar a usá-lo.

Se estavas a pensar em instalares um repositório multi-utilizador numa partilha de rede, pensa outra vez. Consulta [Secção 3.1.4, “Aceder ao Repositório numa Partilha de Rede”](#) para descobrires porque é que nós pensamos que é uma má ideia. Instalar um servidor não é tão difícil como pensas, e dar-te-á mais fiabilidade e provavelmente também velocidade.

As próximas secções são um guia passo-a-passo em como instalar tal servidor numa máquina Windows. É claro que também podes instalar o servidor numa máquina Linux, mas isso está para além do âmbito deste guia. Mais informação sobre as opções do servidor Subversion, e em como escolher a melhor arquitectura para a tua situação, podem ser encontradas no livro do Subversion, em *Server Configuration* [<http://svnbook.red-bean.com/en/1.5/svn.serverconfig.html>].

3.6. Servidor Baseado em Svnserve

3.6.1. Introdução

O Subversion inclui o Svnserve - um servidor ligeiro e independente que usa um protocolo personalizado sobre uma vulgar ligação TCP/IP. É ideal para instalações pequenas, ou onde não pode ser usado um servidor Apache completo.

Na maioria dos casos o svnserve é mais fácil de instalar e corre mais rápido que um servidor baseado em Apache, apesar de que não terá algumas das funcionalidades avançadas. E agora que o suporte SASL está incluído é também mais fácil torná-lo seguro.

3.6.2. Instalar o svnserve

1. Obter a última versão do Subversion a partir de <http://subversion.apache.org/getting.html>. Ou em alternativa obter um instalador pré-embalado a partir do site CollabNet em <http://www.collab.net/downloads/subversion>. O instalador irá instalar o svnserve como um serviço Windows, e irá também incluir algumas das ferramentas que necessitarás se fores usar o SASL por razões de segurança.
2. Se já tens uma versão do Subversion instalada, e o svnserve está a correr, necessitarás de o parar antes de continuares.
3. Corre o instalador do Subversion. Se estiveres a correr o instalador no teu servidor (o que é recomendado) podes saltar o passo 4.

4. Abre o explorador do Windows, vai para a pasta de instalação do Subversion (normalmente C:\Program Files\Subversion) e na pasta bin, encontra os ficheiros `svnserve.exe`, `intl3_svn.dll`, `libapr.dll`, `libapriconv.dll`, `libapriutil.dll`, `libdb*.dll`, `libeay32.dll` e `ssleay32.dll` - copiar esses ficheiros, ou copia apenas toda a pasta bin para uma pasta no teu servidor, e.g. `c:\svnserve`

3.6.3. Executar o svnserver

Agora que o `svnserve` está instalado, precisas de o correr no teu servidor. A abordagem mais simples é correr o seguinte a partir da shell do DOS, ou criar um atalho do Windows:

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

o `svnserve` irá arrancar à espera de pedidos na porta 3690. O parâmetro `--daemon` indica ao `svnserve` para correr como um processo `daemon`, para que exista sempre até ser terminado manualmente.

Se ainda não criaste um repositório, segue as instruções dadas com a configuração do servidor Apache [Secção 3.7.4, “Configuração”](#).

Para testar se o `svnserve` está a funcionar, usa o TortoiseSVN → Navegador de repositório para visualizares o repositório.

Assumindo que o teu repositório está localizado em `c:\repos\TestRepo`, e o teu servidor chama-se `localhost`, introduz:

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

quando fores solicitado pelo navegador de repositório.

Podes também aumentar a segurança e poupar tempo a introduzir URLs com o `svnserve` usando o parâmetro `--root` para indicares a localização raiz e restringir o acesso a determinada pasta no servidor:

```
svnserve.exe --daemon --root drive:\path\to\repository\root
```

Usando o teste prévio como guia, o `svnserve` irá agora correr como:

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

E no TortoiseSVN o nosso URL do navegador de repositório está agora encurtado para:

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

Observa que o parâmetro `--root` é também preciso no caso de o teu repositório estar localizado numa partilha ou unidade diferente, da localização do `svnserve` no teu servidor.

O `svnserve` irá servir um número qualquer de repositórios. Coloca-os apenas sobre a pasta raiz que acabaste de definir, a acede a eles usando um URL relativo a essa raiz.



Atenção

Não crie ou acesse a um repositório Berkeley DB numa partilha de rede. Ele *não poderá* existir num sistema de ficheiros remoto. Nem mesmo se tiveres a unidade de rede mapeada com uma letra de unidade. Se tentares usar o Berkeley DB numa partilha de rede, os

resultados serão imprevisíveis - poderás observar de imediato erros misteriosos, ou poderá levar-te meses até descobrires que a tua base de dados do repositório está subtilmente corrompida.

3.6.3.1. Executar o svnservice como um Serviço

Correr o svnservice como utilizador não é normalmente a melhor abordagem. Isso significa ter um utilizador ligado no teu servidor, e lembra-lo de o voltar a arrancar após um reboot. A melhor abordagem é correr o svnservice como um serviço Windows. A partir do Subversion 1.4, o svnservice pode ser instalado como um serviço Windows nativo.

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

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

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

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

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

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

Observa o formato de linha, não usual, usado pelo sc. Nos pares chave= valor não devem existir espaços entre a chave e o =, mas deverá existir um espaço antes do valor.



Dica

A Microsoft agora recomenda que os serviços corram sobre uma conta Serviço Local ou Serviço de Rede. Consulta o *The Services and Service Accounts Security Planning Guide* [<http://www.microsoft.com/technet/security/topics/serversecurity/serviceaccount/default.mspx>]. Para criar o serviço sob uma conta Serviço Local, anexa o seguinte ao exemplo acima citado.

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

Tem atenção que terás de dar à conta de Serviço Local, direitos de acesso apropriados para o Subversion e para os teus repositórios, tal como para qualquer aplicação que seja usada por qualquer script de gancho. O grupo integrado para este propósito chama-se "LOCAL SERVICE".

Uma vez que tenhas instalado o serviço, necessitarás de ir ao gestor de serviços para o arrancar (só desta vez; ele irá arrancar automaticamente quando o servidor reiniciar).

Para informação mais detalhada, consulta *Windows Service Support for Svnserve* [<http://svn.collab.net/repos/svn/trunk/notes/windows-service.txt>].

Se instalaste uma versão anterior do `svnserve`, usando o empacotador `SVNService`, e agora desejas usar em alternativa o suporte nativo, necessitarás de, des-registar como um serviço, o empacotador (lembra-te de parar primeiro o serviço!). Usa simplesmente o comando:

```
svnservice -remove
```

para remover a entrada do serviço do registo.

3.6.4. Autenticação Básica com o `svnserve`

A instalação por padrão do `svnserve` fornece acesso apenas de leitura anónima. Isso significa que podes usar um URL `svn://` para efectuar checkout e actualizar, ou usar o navegador-de-repositório no TortoiseSVN para visualizar o repositório, mas não poderás submeter quaisquer alterações.

Para activares o acesso de escrita ao repositório, necessitas de editar o ficheiro `conf/svnserve.conf` na pasta do teu repositório. Este ficheiro controla a configuração do daemon `svnserve` e também contém documentação útil.

Podes activa a escrita anónima configurando simplesmente:

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

No entanto não saberás quem efectuou as alterações no repositório, já que a propriedade `svn:author` estará vazia. Também não poderás controlar quem efectua alterações no repositório. Esta é uma configuração arriscada!

Uma maneira de ultrapassar isto é criar uma base de dados de palavras-chave:

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

Onde `userfile` é um ficheiro que existe na mesma pasta do `svnserve.conf`. Este ficheiro pode estar em qualquer parte do teu sistema de ficheiros (útil para quando tens múltiplos repositórios que requerem os mesmos direitos de acesso) e pode ser referenciado usando um caminho absoluto, ou um caminho relativo à pasta `conf`. Se incluíres um caminho este deve ser escrito do seguinte modo `/the/unix/way`. Usar `\` ou letras de unidades não irá funcionar. O `userfile` deverá ter a seguinte estrutura:

```
[users]
username = password
...
```

Este exemplo irá negar todo o acesso a utilizadores não autenticados (anónimos), e irá dar acesso leitura-escrita a os utilizadores listados em `userfile`.



Dica

Se mantiveres múltiplos repositórios utilizando a mesma base de dados de palavras-chave, o uso de um domínio de autenticação irá facilitar a vida dos utilizadores, já que o TortoiseSVN pode efectuar cache das tuas credenciais, para que só tenhas de as introduzir uma vez. Mais informações poderam ser encontradas no livro do Subversion, mais especificamente nas secções [Create a 'users' file and realm](http://svnbook.red-bean.com/en/1.5/svn.serverconfig.svnserve.html#svn.serverconfig.svnserve.auth.users) and [Client](#)

3.6.5. Segurança Melhorada com SASL

3.6.5.1. O que é o SASL?

O Cyrus Simple Authentication and Security Layer é um software open source escrito na Universidade Carnegie Mellon. Adiciona funcionalidades de autenticação genérica e encriptação a qualquer protocolo de rede, e a partir do Subversion 1.5, ambos o servidor `svnserve` e o cliente TortoiseSVN sabem utilizar esta biblioteca.

Para uma discussão mais completa sobre as opções disponíveis, deverás olhar para o livro do Subversion na secção [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]. Se estás apenas a pensar numa forma simples de utilizar uma autenticação segura e encriptada num servidor Windows, para que o teu repositório possa ser acedido na grande e má Internet, continua a ler.

3.6.5.2. Autenticação SASL

Para activares mecanismos SASL específicos no servidor necessitarás de três coisas. Primeiro, criar uma secção `[sasl]` no ficheiro `svnserve.conf` dos teus repositórios, com este par chave-valor:

```
use-sasl = true
```

Segundo, criar um ficheiro chamado `svn.conf` numa localização conveniente - tipicamente na pasta onde está instalado o Subversion.

Terceiro, criar duas novas entradas no registo para dizer ao SASL onde procurar as coisas. Cria uma chave de registo chamada `[HKEY_LOCAL_MACHINE\SOFTWARE\Carnegie Mellon\Project Cyrus\SASL Library]` e coloa dois novos valores string lá dentro: o `SearchPath` configurado com o caminho da pasta contendo os plug-ins `sasl*.dll` (normalmente na pasta de instalação do Subversion), e o `ConfFile` configurado com a pasta contendo o ficheiro `svn.conf`. Se usares o instalador CollabNet, essas chaves já terão sido instaladas por ti.

Edita o ficheiro `svn.conf` para que contenha o seguinte:

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

A última linha mostra a localização da base de dados de autenticação, que é um ficheiro chamado `sasldb`. Este poderá ir para qualquer localização, mas uma escolha conveniente será o caminho pai do repositório. Tem a certeza que o serviço `svnserve` tem acesso de leitura para este ficheiro.

Se o `svnserve` já está a correr, necessitarás de o reiniciar para garantir que lê a configuração actualizada.

Agora que está tudo configurado, tudo o que precisas é de criar alguns utilizadores e palavras-chave. Para o fazeres necessitarás do programa `saslpasswd2`. Se usares o instalador CollabNet esse programa estará na pasta `install`. Usa um comando parecido com este:

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

O parâmetro `-f` dá-te a localização da base de dados, o `realm` deverá ter o mesmo valor que definiste no ficheiro `svnserve.conf` dos teus repositórios, e o `username` será exactamente aquilo que estás à espera. Observa que no `realm` não é permitido ter espaços.

Tu podes listar os nomes de utilizadores (usernames) armazenados na base de dados usando o programa `sasldblistusers2`.

3.6.5.3. Encriptação SASL

Para activar ou desactivar diferentes níveis de encriptação, tu podes configurar dois valores no ficheiro `svnserve.conf` do teu repositório:

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

As variáveis `min-encryption` e `max-encryption` controlam o nível de encriptação requerida pelo servidor. Para desactivar completamente a encriptação, coloca ambos os valores a 0. Para activar a simples soma de verificação de dados (i.e., para prevenir a adulteração de dados e garantir a sua integridade sem recurso à encriptação), coloca ambos os valores a 1. Se desejas permitir (mas não obrigar) a encriptação, coloca o valor mínimo a 0 e o valor máximo a um qualquer bit-length. Para requisitares incondicionalmente a encriptação, coloca em ambos os valores números maiores que 1. No nosso exemplo prévio, nós requeremos que os clientes tivessem uma encriptação mínima de 128-bits, mas não mais que 256-bits.

3.6.6. Autenticação com svn+ssh

Outra forma de autenticar utilizadores num servidor baseado em `svnserve`, é usar o secure shell (SSH) para criar um túnel. Não é tão simples como configurar o SASL, mas poderá ser útil em alguns casos.

Com esta abordagem o `svnserve` não irá correr como um processo `daemon`, mas em alternativa a shell segura irá arrancá-lo por ti, correndo-o como utilizador autenticado SSH. Para o activares necessitas de um `daemon` de shell no teu servidor.

Um método básico para configures teu servidor é dado em [Apêndice G, Protegendo o Svnserve ao usar o SSH](#). Podes encontrar outros tópicos SSH no interior da FAQ, procurando por “SSH”.

Mais informações sobre o `svnserve` podem ser encontradas em [Version Control with Subversion](http://svnbook.red-bean.com) [http://svnbook.red-bean.com].

3.6.7. Autorização com o svnserve baseada em Caminhos

A partir do Subversion 1.3, o `svnserve` suporta o mesmo esquema de autorização `mod_authz_svn` baseado em caminhos que está disponível com o servidor Apache. Precisas de editar o ficheiro `conf/svnserve.conf` na pasta do teu repositório, e adicionar uma linha indicando o teu ficheiro de autorização.

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

Aqui, o `authz` é um ficheiro que criaste para definir as permissões de acesso. Podes usar um ficheiro separado para cada repositório, ou poderás usar o mesmo ficheiro para vários repositórios. Consulta [Secção 3.7.6, “Autorização Baseada-em-Caminhos”](#) para uma descrição do formato do ficheiro.

3.7. Servidor Baseado em Apache

3.7.1. Introdução

A instalação para o Subversion mais flexível de todas é a baseada e Apache. Apesar de um pouco mais complicada de configurar, oferece no entanto mais benefícios que as outras:

WebDAV

O servidor Subversion baseado em Apache usa o protocolo WebDAV, que também é suportado por muitos outros programas. Tu podes, por exemplo, montar um repositório como uma “pasta Web” no explorador do Windows e aceder a ela como outra qualquer pasta do sistema de ficheiros.

Navegar no Repositório

Podes apontar o teu navegador de internet para o URL do teu repositório e navegar nos seus conteúdos, sem teres instalado um cliente Subversion. Isto dá acesso aos teus dados a um círculo mais abrangente de utilizadores.

Autenticação

Podes usar qualquer tipo de autenticação que o Apache suporte, incluindo SSPI e LDAP.

Segurança

Visto que o Apache é muito estável e seguro, tu automaticamente obterás a mesma segurança para o teu repositório. Isto inclui encriptação SSL.

3.7.2. Instalar o Apache

O primeiro pré-requisito para instalar o Apache é um computador com o Windows 2000, Windows XP +SP1, Windows 2003, Vista ou Server 2008.



Atenção

Tem em atenção que o Windows XP, sem o serviço pack 1, irá adulterar os dados de rede e com isso corromper o teu repositório!

1. Descarrega a última versão do servidor web Apache a partir de <http://httpd.apache.org/download.cgi>. Tem a certeza que descarregas a versão 2.2.x - a versão 1.3.xx não funcionará!

O instalador msi para o Apache pode ser encontrado ao clicares em `other files`, e navegares para `binaries/win32`. Poderás querer escolher o ficheiro `msi apache-2.2.x-win32-x86-openssl-0.9.x.msi` (o que inclui o OpenSSL).

2. Uma vez que tenhas o instalador Apache2, podes efectuar duplo clique sobre ele, e então serás guiado através do processo de instalação. Tem a certeza que inseres o URL-servidor correctamente (se não tiveres um nome DNS para o teu servidor, insere apenas o seu endereço-IP). Eu recomendo instalar o Apache *para Todos os Utilizadores, no Porto 80, como um Serviço*. Nota: se já tens o IIS ou outro programa a correr que esteja à escuta no porto 80, poderá causar a falha da instalação. Se tal acontecer, vai à pasta do programas, `\Apache Group\Apache2\conf` e localiza o ficheiro `httpd.conf`. Edita-o para que o `Listen 80` seja alterado para um porto livre, e.g. `Listen 81`. Então reinicia a instalação - desta vez deverá concluir sem problemas.
3. Agora testa se o teu servidor web Apache está a correr correctamente, apontando o teu navegador de internet para `http://localhost/` - um sítio web pré-configurado deverá aparecer.



Cuidado

Se decidires instalar o Apache como um serviço, tem atenção que por defeito este irá correr com uma conta de sistema local. Será uma prática mais segura para ti criar uma conta separada para o Apache correr.

Tem a certeza que a conta em que o Apache está a correr, tem uma entrada explícita na lista de controlo de acessos às pastas dos repositórios (clica com o botão direito na pasta | propriedades | segurança), com acesso total. De outro modo, os utilizadores não poderão submeter as suas alterações.

Mesmo que o Apache corra como sistema local, irás necessitar na mesma de tal entrada (que será neste caso a conta SYSTEM).

Se o Apache não tiver esta permissão configurada, os teus utilizadores irão obter mensagens de erro de “Acesso negado”, que irão aparecer no registo de erro do Apache como error 500.

3.7.3. Instalar o Subversion

1. Descarrega a última versão dos binários Win32 do Subversion para o Apache. Tem a certeza que obténs a versão correcta para integrares com a tua versão do Apache, de outro modo obterás uma mensagem de erro obscura, quando tentares reiniciar. Se tens o Apache 2.2.x vai a <http://subversion.tigris.org/servlets/ProjectDocumentList?folderID=8100>.
2. Corre o instalador do Subversion e segue as instruções. Se o instalador do Subversion reconhece que tens o Apache instalado, então está praticamente concluído. Se não encontra o servidor Apache, então tens de efectuar passos adicionais.

3.

Usando o explorador do Windows, vai à pasta de instalação do Subversion (usualmente `c:\programas\Subversion`) e procura os ficheiros `/httpd/mod_dav_svn.so` e `mod_authz_svn.so`. Copia esses ficheiros para a pasta de módulos do Apache (normalmente `c:\programas\apache group\apache2\modules`).

4. Copia o ficheiro `/bin/libdb*.dll` e o `/bin/intl3_svn.dll` da pasta de instalação do Subversion para a pasta bin do Apache.
5. Edita o ficheiro de configurações do Apache (normalmente `C:\Programas\Apache Group\Apache2\conf\httpd.conf`) com um editor de texto como o Notepad, e efectua as seguintes alterações:

Descomenta (remove a marca '#') das seguintes linhas:

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

Adiciona as duas seguintes linhas no fim da secção LoadModule.

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

3.7.4. Configuração

Agora já instalaste o Apache e o Subversion, mas o Apache ainda não sabe lidar com os clientes do Subversion, como o TortoiseSVN. Para fazeres com que o Apache conheça o URL que irá ser usado para os repositórios do Subversion, terás de editar o ficheiro de configuração do Apache (normalmente localizado em `c:\programas\apache group\apache2\conf\httpd.conf`) com um qualquer editor de texto com o (e.g. Notepad):

1. No final do ficheiro de configuração, adiciona as duas seguintes linhas::

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

```
Require valid-user
</Location>
```

Estas configuram o Apache para que todos os repositórios Subversion estejam fisicamente localizados debaixo da pasta D:\SVN. Os repositórios são servidos para o mundo exterior a partir do URL: `http://MyServer/svn/`. O acesso é restricto a utilizadores/palavras-passe listadas no ficheiro `passwd`.

2. Para criar o ficheiro `passwd`, abre outra vez a linha de comandos (caixa-do-DOS) e vai para a pasta `apache2` (normalmente em `c:\programas\apache_group\apache2`) e cria o ficheiro ao introduzires

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

Isto irá criar um ficheiro com o nome `passwd` que será usado para autenticação. Utilizadores adicionais podem ser adicionados com

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

3. Reinicia novamente o serviço Apache.
4. Aponta o teu navegador de internet para `http://MyServer/svn/MyNewRepository` (onde `MyNewRepository` é o nome do repositório Subversion que criaste anteriormente). Se tudo correu bem, deverás ser solicitado com o nome de utilizador e palavra passe, então poderás ver os conteúdos do teu repositório.

Uma explicação curta do que acabaste de introduzir:

Configuração	Explicação
<code><Location /svn></code>	significa que os repositórios Subversion estão disponíveis a partir do URL <code>http://MyServer/svn/</code>
<code>DAV svn</code>	indica ao Apache qual o módulo que será responsável por servir esse URL - Neste caso o módulo Subversion.
<code>SVNListParentPath on</code>	A partir da versão 1.3 do Subversion, esta directiva permite listar todos os repositórios disponíveis sob <code>SVNParentPath</code> .
<code>SVNParentPath D:\SVN</code>	indica ao Subversion para procurar por repositórios sob <code>D:\SVN</code>
<code>SVNIndexXSLT "/svindex.xsl"</code>	Usado para tornar mais bonita a navegação com um navegador de internet.
<code>AuthType Basic</code>	é para activar a autenticação básica, i.e. Utilizador/Palavra-passe
<code>AuthName "Subversion repositories"</code>	é usada como informação, quando aparece uma caixa de diálogo, a dizer ao utilizador qual o propósito da autenticação
<code>AuthUserFile passwd</code>	especifica qual o ficheiro de palavra-passe a utilizar para a autenticação.
<code>AuthzSVNAccessFile</code>	Localização do ficheiro de Acessos, a caminhos no interior do repositório Subversion.
<code>Require valid-user</code>	indica que só os utilizadores que introduziram o utilizador/palavra-passe correctas, estão autorizados a aceder ao URL

Tabela 3.1. Configurações do `httpd.conf` do Apache

Mas é só um exemplo. Existem muitas mais possibilidades do que podes fazer com o servidor web Apache.

- Se quiseres que o teu repositório tenha acesso de leitura para toda a gente, mas só permita acesso de escrita para utilizadores específicos, podes alterar a linha

```
Require valid-user
```

```
to
```

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

- Ao utilizar um ficheiro `passwd` limitas a concedes o acesso a todos os teus repositórios como uma unidade. Se quiseres mais controlo sobre quais os utilizadores que tem acesso a cada pasta, dentro do repositório, podes descomentar a linha

```
#AuthzSVNAccessFile svnaccessfile
```

e criar um ficheiro de acesso ao Subversion. O Apache irá garantir que só os utilizadores válidos poderão aceder à tua localização `/svn`, e irá então enviar o nome de utilizador ao módulo `AuthzSVNAccessFile` do Subversion, para que este possa garantir um acesso mais granular, baseado nas regras listadas no ficheiro de acesso do Subversion. Observa que os caminhos são especificados ou como `repos:path` ou simplesmente `path`. Se não especificares um repositório em particular, essa regra ir-se-á aplicar a todos os repositórios sob `SVNParentPath`. O formato do ficheiro de políticas-de-autorização usada por `mod_authz_svn` é descrita em [Secção 3.7.6, “Autorização Baseada-em-Caminhos”](#)

- Para tornar a navegação do repositório mais bonita, utilizando um navegador de internet, descomenta a linha

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

e coloca os ficheiros `svnindex.xsl`, `svnindex.css` e `menucheckout.ico` na tua pasta raiz dos documentos (normalmente `C:/Program Files/Apache Group/Apache2/htdocs`). A pasta está configurada com a directiva `DocumentRoot`, no teu ficheiro de configuração do Apache.

Tu podes obter esses três ficheiros directamente do nosso repositório fonte, em <http://tortoissvn.googlecode.com/svn/trunk/contrib/svnindex>. ([Secção 3, “O TortoiseSVN é grátis!”](#) explica como aceder ao repositório fonte do TortoiseSVN).

O ficheiro XSL do repositório do TortoiseSVN tem um bom chamariz: se navegares no repositório com o teu navegador web, então cada pasta do teu repositório mostrará um ícone à direita. Se clicares nesse ícone, a caixa de diálogo do TortoiseSVN checkout ir-se-á iniciar para este URL.

3.7.5. Repositórios Múltiplos

Se utilizas a directiva `SVNParentPath`, então não terás de alterar o ficheiro de configuração do Apache de cada vez que adicionares um novo repositório Subversion. Simplesmente cria o novo repositório na mesma localização do primeiro repositório e está concluído! Na minha empresa, eu tenho acesso directo a essa pasta específica no servidor via SMB (acesso normal ao sistema de ficheiros Windows). Então eu crio apenas uma nova pasta aí, corro o comando TortoiseSVN → Criar repositório aqui... do TortoiseSVN, e o novo projecto tem agora um lar...

Se estás a usar o Subversion 1.3 ou superior, podes usar a directiva `SVNListParentPath on` para permitir ao Apache produzir uma listagem de todos os projectos disponíveis, se apontares o teu navegador web ao caminho pai, em vez de a um repositório específico.

3.7.6. Autorização Baseada-em-Caminhos

O módulo `mod_authz_svn` permite um controlo mais fino das permissões de acesso baseadas nos nomes dos utilizadores e nos caminhos do repositório. Isto está disponível com o servidor Apache, e desde o Subversion 1.3 está também disponível com o `svnserve`.

Um ficheiro de exemplo teria o seguinte aspecto:

```
[groups]
admin = john, kate
devteam1 = john, rachel, sally
devteam2 = kate, peter, mark
docs = bob, jane, mike
training = zak
# Regra de acesso, por defeito, para TODOS os repositórios
# Todos podem ler, admins podem escrever, Dan German está excluído.
[/]
* = r
@admin = rw
dangerman =
# Permite total acesso dos programadores aos repos do seu projecto
[proj1:/]
@devteam1 = rw
[proj2:/]
@devteam2 = rw
[bigproj:/]
@devteam1 = rw
@devteam2 = rw
trevor = rw
# Da acesso de escrita a ao pessoal dos docs a todas as pastas de docs
[/trunk/doc]
@docs = rw
# Dá aos estagiários acesso de escrita apenas ao repositório de treino
[TrainingRepos:/]
@training = rw
```

Observa que verificar cada caminho pode ser uma operação cara, em particular no caso do registo de revisão. O servidor verifica cada caminho alterado em cada revisão, e verifica se está disponível para leitura, o que pode ser consumidor de tempo em revisões que afectam um grande número de ficheiros.

Autenticação e autorização são dois processos separados. SE o utilizador deseja ter acesso a um caminho do repositório, terá de conhecer *ambos*, os requisitos usuais de autenticação e os requisitos de autorização para aceder ao ficheiro.

3.7.7. Autenticação Com um Domínio Windows

Como poderás ter notado, necessitarás de criar uma entrada utilizador/palavra-chave no ficheiro `passwd`, para cada utilizador em separado. E se (por razões de segurança) quiseres que os teus utilizadores modifiquem periodicamente as suas palavras-passe, terás de efectuar manualmente a alteração.

Mas existe uma solução para esse problema - pelo menos se estás a aceder ao repositório a partir da LAN com um controlador de domínio Windows: `mod_auth_sspi`!

O módulo SSPI original foi oferecido pela Syneapps, incluindo o código-fonte. Mas o desenvolvimento foi então cancelado. No entanto não desespere, pois a comunidade pegou nele e melhorou-o. Ele tem agora um novo lar, em *SourceForge* [<http://sourceforge.net/projects/mod-auth-sspi/>].

- Descarrega o módulo que corresponde à tua versão do Apache, e copia o ficheiro `mod_auth_sspi.so` para a pasta dos módulos Apache.
- Edita o ficheiro de config. do Apache: adiciona a linha

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

à secção `LoadModule`. Tem a certeza que inseriste esta linha *antes* da linha

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

- Para fazer com que a localização do Subversion use este tipo de autenticação, terás de alterar a linha

```
AuthType Basic
```

para

```
AuthType SSPI
```

, necessitas também de adicionar

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

dentro do bloco `<Location /svn>`. Se não tiveres um controlador de domínio, deixa o nome do controlador de domínio como `<domaincontroller>`.

Tem atenção que, se te autenticas utilizando o SSPI, não precisas mais então da linha `AuthUserFile` para definir o ficheiro de palavra-passe. O Apache autentica então o teu nome de utilizador e palavra passe com o domínio Windows. Necessitarás de actualizar a tua lista de utilizadores no teu `svnaccessfile` para referenciar também o `DOMAIN\username`.



Importante

A autenticação SSPI só está activada para ligações SSL seguras (https). Se só estás a usar ligações http normais para o teu servidor, isto não irá funcionar.

Para activar o SSL no teu servidor, consulta o capítulo: [Secção 3.7.9, “Proteger o servidor com SSL”](#)



Dica

Os ficheiros `AuthzSVNAccessFile` do Subversion são sensíveis a maiúsculas e minúsculas, em relação aos nomes de utilizadores (`JUser` é diferente de `juser`).

No mundo Microsoft, os domínios Windows e os nomes de utilizador não são sensíveis a maiúsculas e minúsculas. Mesmo assim alguns administradores de redes gostam de criar contas de utilizadores em CamelCase (e.g. `JUser`).

Esta diferença pode-te trair ao usar a autenticação SSPI como domínio do Windows e os nomes dos utilizadores são passados ao Subversion tal como o utilizador os digita na prompt. O Internet Explorer passa com frequência automaticamente o nome de utilizador para o Apache, usando a altura de letra com que foi criada a conta.

O resultado final é que poderás necessitar de pelo menos duas entradas no teu `AuthzSVNAccessFile` para cada utilizador -- uma entrada em letra minúscula e uma entrada com a mesma altura de letra com que o Explorador passa para o Apache. Também irás necessitar de formar os teus utilizadores para também digitar as suas credenciais usando um tipo de letra minúscula, ao aceder aos repositórios via TortoiseSVN.

Os registos de Erro e Acesso do Apache são os teus melhores amigos a decifrar problemas como estes, tal como te irão ajudar a determinar a string do nome do utilizador passada para o módulo `AuthzSVNAccessFile` Subversion. Poderás necessitar de experimentar com o formato exacto da string do utilizador no `svnaccessfile` (e.g. `DOMAIN\user` vs. `DOMAIN//user`) para conseguires por tudo a funcionar.

3.7.8. Múltiplas Fontes de Autenticação

É também possível ter mais do que uma fonte de autenticação para o teu repositório Subversion. Para o fazeres, necessitas fazer com que cada tipo de autenticação seja não-autorizada, para que o Apache verifique nas múltiplas fontes por uma correspondência utilizador/palavra-passe.

Um cenário comum é o uso de autenticação por domínio Windows e por um ficheiro `passwd`, para que possas providenciar acesso a utilizadores fora do domínio Windows.

- Para activar a autenticação a ambos, domínio Windows e ficheiro `passwd`, adiciona as seguintes entradas dentro do bloco `<Location>` do teu ficheiro de config do Apache:

```
AuthBasicAuthoritative Off
SSPIAuthoritative Off
```

Aqui está um exemplo de um ficheiro de configuração do Apache completo para a autenticação combinada, domínio Windows e ficheiro `passwd`:

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

  AuthName "Subversion repositories"
  AuthzSVNAccessFile svnaccessfile.txt

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

# Htpasswd Logins.
AuthType Basic
AuthBasicAuthoritative Off
AuthUserFile passwd

Require valid-user
```

</Location>

3.7.9. Proteger o servidor com SSL

Mesmo que o Apache 2.2.x tenha suporte OpenSSL, este não está activado por defeito. Necessitarás de o activar manualmente.

1. No ficheiro de config do Apache, descomenta as linhas:

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

e no fundo

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

então altera a linha (numa só linha)

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

para

```
SSLMutex default
```

2. A seguir necessitas de criar um certificado SSL. Para o fazeres abre a linha de comandos (caixa-DOS) e muda para a pasta Apache (e.g. C:\program files\apache group\apache2) então digita o seguinte comando:

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

Ir-te-á ser solicitada uma frase-passe. Por favor, não utilizes palavras simples mas sim frases completas, e.g. uma parte de um poema. Quanto mais longa a frase, melhor. Também terás de introduzir o URL do teu servidor. Todas as outras questões são opcionais, mas recomendamos que também as preenchas.

Normalmente o ficheiro `privkey.pem` é criado automaticamente, mas se não o for, necessitas de digitar o seguinte comando para gerá-lo:

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

Next type the commands

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

and (on one line)

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

and then (on one line)

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

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

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

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

3. Reinicia o serviço Apache

4. Aponta o teu navegador para `https://servername/svn/project ...`



SSL e o Internet Explorer

Se estás a proteger o teu servidor com SSL num domínio Windows, irás descobrir que já não consegues navegar no repositório com o Internet Explorer. Não te preocupes - isto é só porque o Internet Explorer não se consegue autenticar. Outros navegadores não têm esse problema, tal como o TortoiseSVN e qualquer outro cliente Subversion, que ainda assim conseguirão se autenticar.

Se ainda assim queres usar o Internet Explorer para navegar no repositório, poderás:

- Definir uma directiva `<Location /path>` separada no ficheiro de conf do Apache, e adiciona `SSPIBasicPreferred On`. Isto irá permitir ao Internet Explorer se autenticar de novo, mas entretanto os outros navegadores e o Subversion não se conseguirão autenticar para essa localização.
- Oferece também a navegação com autenticação não encriptada (sem SSL). Estranhamente o IE não tem quaisquer problemas a autenticar-se, se a ligação não está protegida com SSL.
- Na configuração "padrão" do SSL, existe normalmente a seguinte afirmação no host SSL virtual do Apache :

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

Existem (existiam?) boas razões para esta configuração, consulta http://www.modssl.org/docs/2.8/ssl_faq.html#ToC49 Mas se quiseres autenticação NTLM terás de usar `keepalive`. Se descomentares todo o `SetEnvIf` deverás conseguir autenticar o IE com autenticação Windows sobre SSL para o Apache no Win32 incluindo o `mod_auth_sspi`.



Forçando o acesso SSL

Quando configuraste o SSL para tornar o teu repositório mais seguro, querias desactivar o acesso por via normal não-SSL (http), e só permitir o acesso https. Para o fazeres terás de adicionar uma outra directiva ao bloco `<Location>` do Subversion: `SSLRequireSSL`.

Um exemplo do bloco `<Location>` teria o seguinte aspecto:


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

3.7.10. Usando certificador cliente com hospedeiros SSL virtuais.

Enviado para a lista de correio do TortoiseSVN pelo Nigel Green. Obrigado!

Em algumas configurações do servidor poderás necessitar de configurar um único servidor com dois hospedeiros SSL virtuais. O primeiro para acesso web público, sem requisitos de certificado de cliente. O segundo mais seguro, correndo o servidor Subversion com o certificado de cliente obrigatório.

Adicionando a directiva `SSLVerifyClient Optional` à secção *per-server* da configuração do Apache (i.e. fora de qualquer bloco `VirtualHost` e `Directory`), força o Apache a requisitar um Certificado cliente, na negociação SSL inicial. Devido a um bug no `mod_ssl` é essencial que o certificado seja pedido neste ponto, já que não funcionará se a ligação SSL for pré-negociada.

A solução é adicionar a seguinte directiva à pasta hospedeiro virtual que desejas bloquear para o Subversion:

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

Esta directiva garante acesso à pasta só se o certificado cliente foi recebido e verificado com sucesso.

Para resumir, as linhas relevantes da configuração do Apache são:

```
SSLVerifyClient Optional

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

127.0.0.1:443>
```

Capítulo 4. Guia de Uso Diário

Este documento descreve a utilização diária do cliente TortoiseSVN. *Não* é uma introdução aos sistemas de controlo de versões *nem* uma introdução ao Subversion (SVN), é mais um local a que podes recorrer quando tens uma ideia do que pretendes mas não te lembras como o fazer.

Se necessitas de uma introdução ao controlo de versões com o Subversion, então nós recomendamos a leitura do fantástico livro: *Version Control with Subversion* [<http://svnbook.red-bean.com/>].

Este documento também estará continuamente em actualização, tal como o Tortoise e o Subversion. Se encontrares algumas falhas, relata-os por favor às mailing lists para que possamos actualizar a documentação. Alguns dos screenshots no Guia de Uso Diário (GUD) poderão não reflectir a versão corrente do software. Perdoem-nos por esse facto, pois estamos a trabalhar no TortoiseSVN no nossos tempos livres.

De modo a conseguir o máximo proveito do Guia de Uso Diário:

- Deverás ter o TortoiseSVN já instalado.
- Deverás estar familiarizado com os sistemas de controlo de versões.
- Deverás conhecer os fundamentos do Subversion.
- Deverás ter o servidor configurado e/ou ter acesso a um repositório do Subversion.

4.1. Começando

4.1.1. Sobreposição de Ícones

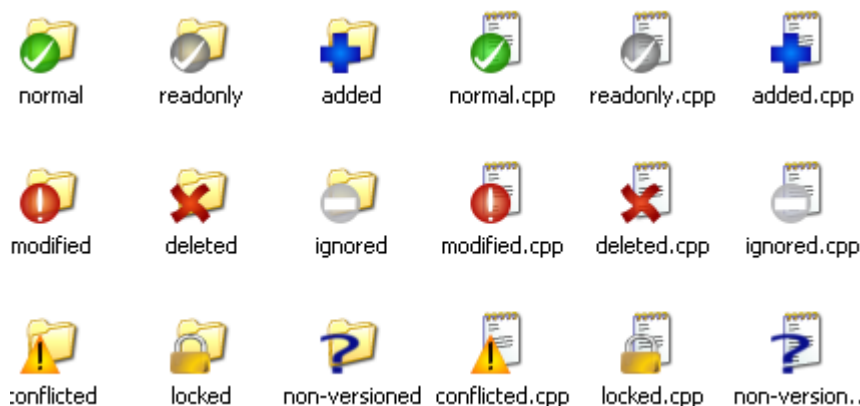


Figura 4.1. O explorador mostrando os ícones sobrepostos

Uma das funcionalidades mais visíveis do TortoiseSVN são os ícones sobrepostos que aparecem nos ficheiros na tua cópia de trabalho. Estes dão-te uma vista geral dos ficheiros que foram modificados. Consultar [Secção 4.7.1, “Sobreposição de Ícones”](#) para saber mais sobre o que cada ícone sobreposto representa.

4.1.2. Menus de Contexto

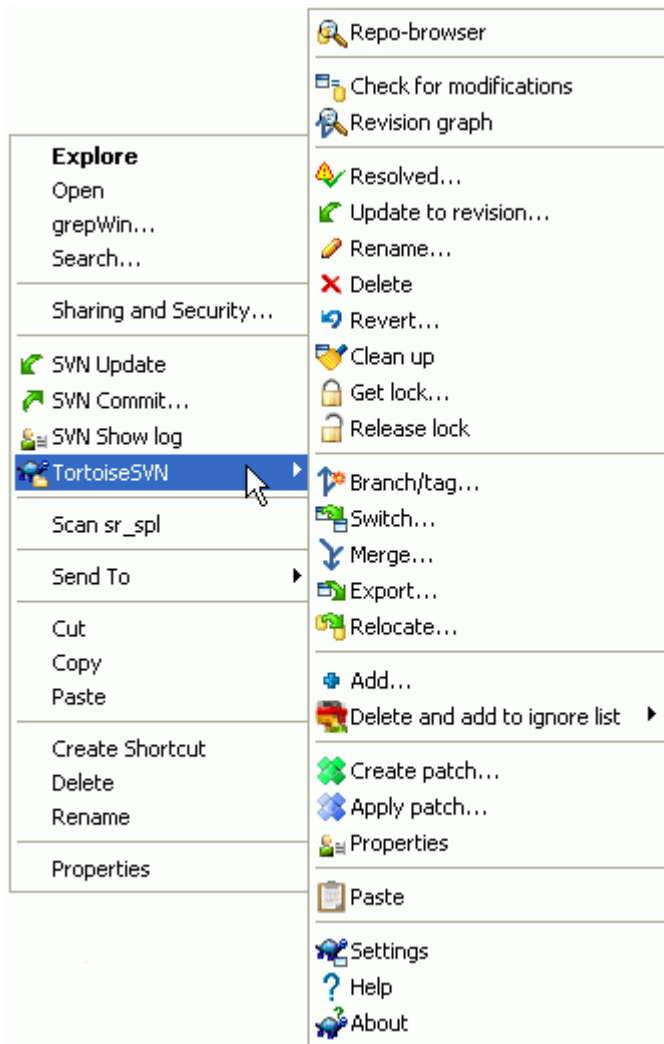


Figura 4.2. Menu de Contexto de uma pasta sob controlo de versões.

Todos os comandos do TortoiseSVN são invocados a partir do menu de contexto no explorador do Windows. A maioria estão directamente visíveis quando clicas com o botão direito numa pasta ou ficheiro. A lista de comandos disponíveis dependem do caso de se tratar de uma pasta ou um ficheiro, ou se o ficheiro pai se encontra sob controlo de versões, ou não. Podes também ver o menu do TortoiseSVN como uma parte do menu ficheiro do Explorador.



Dica

Alguns comandos que são utilizados muito raramente, estão disponíveis apenas no menu de contexto extendido. Para mostrar o menu de contexto extendido, manter premida a tecla **Shift** enquanto clicas com o botão direito do rato.

Em alguns casos verás várias entradas TortoiseSVN, o que não representa um bug!

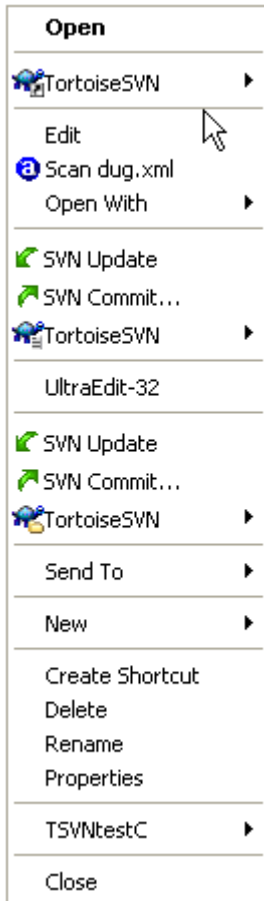


Figura 4.3. Menu de ficheiro do Explorador para um atalho numa pasta versionada.

Neste exemplo está um atalho não versionado dentro de uma pasta versionada, e no menu ficheiro do Explorador estão *três* entradas para o TortoiseSVN. Uma é para a pasta, outra para o atalho em si e a terceira é para o objecto para o qual o atalho aponta. Para ajudar a distinguir entre eles, os ícones têm um indicador no canto inferior direito, para distinguir se a entrada de menu é para um ficheiro, pasta ou atalho para múltiplos itens seleccionados.

Se usas o Windows 2000 verás que os menus de contexto serão mostrados como texto simples, sem os menus de ícones mostrados acima. Estamos conscientes de que em versões anteriores isto estava a funcionar, mas a Microsoft alterou a forma como os seus handlers de ícones funcionam no Vista, forçando-nos a utilizar um método diferente de visualização que infelizmente já não funcionam para o Windows 2000.

4.1.3. Arrastar e Largar

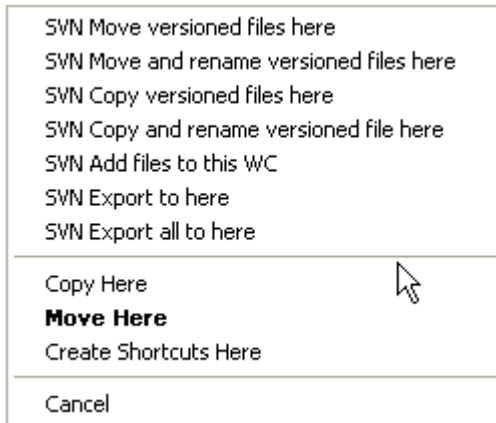


Figura 4.4. Menu arrastar com o botão direito para uma pasta sob controlo de versões.

Outros comandos estão disponíveis como opções de arrasto quando arrastas com o botão direito ficheiros ou pastas para uma nova localização dentro da cópia de trabalho, ou quando arrastas com o botão direito um ficheiro não versionado ou uma pasta para outra pasta sob o controlo de versões.

4.1.4. Atalhos comuns

Algumas operações comuns têm atalhos no Windows, mas não aparecem em botões ou menus. Se não consegues descobrir como fazer alguma coisa óbvia, como refrescar uma vista, vê aqui.

F1

Ajuda, claro.

F5

Refrescar a vista corrente. Esta é provavelmente o comando de tecla única mais útil. Por exemplo ... No Explorador, os ícones sobrepostos da tua cópia de trabalho serão actualizados (refrescados). Na caixa de diálogo submeter, a cópia de trabalho será reanalisada de modo a verificar o que precisa de ser submetido. Na caixa de diálogo Mensagens de Registo, o repositório será contactado de novo para verificar as alterações mais recentes.

Ctrl-A

Seleccionar todos. Poderá ser utilizado em caso de receberes uma mensagem de erro quiseres copiar e colar para um email. Usa Ctrl-A para seleccionar a mensagem de erro e...

Ctrl-C

... Copiar o texto seleccionado.

4.1.5. Autenticação

Se o repositório a que tentas aceder está protegido por palavra-passe, uma caixa de diálogo de autenticação irá surgir.

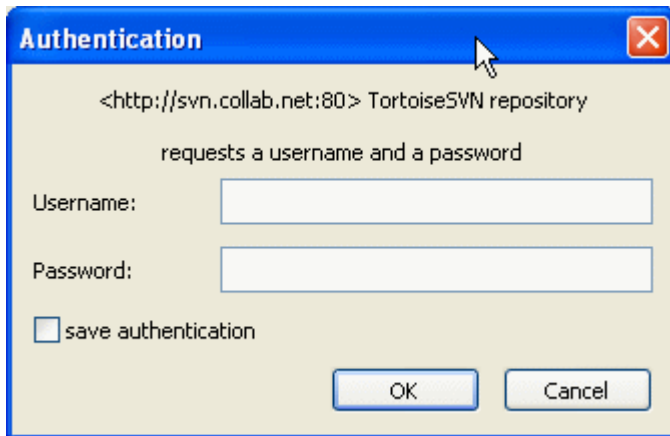


Figura 4.5. Caixa de diálogo de autenticação

Introduzir o teu nome de utilizador e palavra-passe. A caixa de verificação irá fazer com que o TortoiseSVN guarde as credenciais na pasta por defeito do Subversion: %APPDATA%\Subversion\auth em três subpastas:

- `svn.simple` contém as credenciais para autenticação básica (username/password).
- `svn.ssl.server` contém os certificados SSL de servidor.
- `svn.username` contém as credenciais para autenticação apenas por utilizador (sem necessidade de palavra-passe).

Se quiseres limpar a cache de autenticação para todos os servidores, poderás fazê-lo a partir da página **Saved Data** da caixa de diálogo de definições do TortoiseSVN. Esse botão irá limpar todos os dados de autenticação guardados nas pastas `auth` do Subversion, tal como quaisquer dados de autenticação guardados no registo por versões anteriores do TortoiseSVN. Ver [Secção 4.30.6, “Preferências de Dados Guardados”](#).

Algumas pessoas gostam que os dados de autenticação sejam apagados quando terminam a sessão no Windows, ou quando desligam o computador. A maneira de obter isso é através do uso de um script que apague a pasta %APPDATA%\Subversion\auth, ou seja

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

Poderás obter uma descrição de como instalar tal script em [windows-help-central.com](http://www.windows-help-central.com/windows-shutdown-script.html) [http://www.windows-help-central.com/windows-shutdown-script.html].

Para mais informações em como configurar o teu servidor em autenticação e controlo de acessos, consulta [Secção 3.5, “Aceder ao Repositório”](#)

4.1.6. Maxiizando Janelas

Muitas das caixas de diálogos do TortoiseSVN têm muita informação para visualizar, pelo que é normalmente útil maximizar só a altura ou largura, em vez de maximizar para preencher o ecrã. Como conveniência existem atalhos para este caso no botão Maximizar. Utilizar o botão do meio do rato para maximizar verticalmente e o do direito do rato para maximizar horizontalmente

4.2. Importando Dados Para Um Repositório

4.2.1. Importar

If you are importing into an existing repository which already contains some projects, then the repository structure will already have been decided. If are importing data into a new repository then it is worth taking the time to think about how it will be organised. Read [Secção 3.1.5, “Estrutura do Repositório”](#) for further advice.

Esta secção descreve o comando import do Subversion, que é desenhado para importar uma hierarquia de pastas para um repositório de uma vez só. Apesar de executar o trabalho, possui no entanto algumas limitações:

- Não existe modo de seleccionar ficheiros e pastas a incluir, tal como utilizar as configurações do padrão global para ignorar arquivos.
- A pasta importada não se torna uma cópia de trabalho. Terás de efectuar um SVN exportar para copiar os ficheiros de volta do servidor.
- É fácil importar a pasta de nível errado para o repositório.

Por essas razões recomendamos que não use em definitivo o comando import, mas antes, siga o método de dois passos descrito em [Secção 4.2.2, “Importar no local”](#). Mas como estás aqui, eis como funciona o import básico...

Antes de importares o teu projecto para o repositório deverás:

1. Remover todos os ficheiros que não são precisos para construir o projecto (ficheiros temporários, ficheiros gerados pelo compilador, por exemplo, *.obj, binários compilados, ...)
2. Organizar os ficheiros em pastas e subpastas. Apesar de ser possível renomear/remover os ficheiros mais tarde, é altamente recomendado estruturar correctamente o teu projecto antes de o importar!

Agora seleccionar a pasta de topo da estrutura de pastas do teu projecto no windows explorer e, clica c/ o botão direito para abrir o menu de contexto. Selecciona o comando TortoiseSVN → Importar... que mostrará a caixa de diálogo:

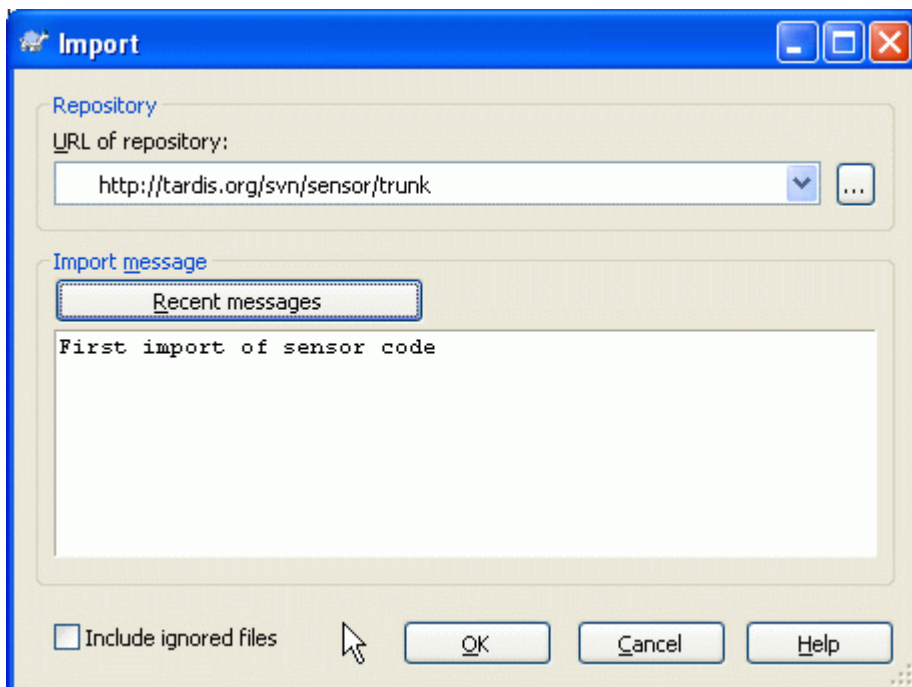


Figura 4.6. Caixa de diálogo Importar

Nesta caixa de diálogo terás de introduzir o URL da localização do repositório, no qual queres importar o teu projecto. É importante ter a noção que a pasta local, que estás a importar, não aparecerá ela própria no repositório, só o seu conteúdo. Por exemplo, se tens uma estrutura:

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

e importas C:\Projects\Widget em `http://mydomain.com/svn/trunk`, então poderás ter a surpresa de descobrir que as tuas subpastas vão direitas para trunk em vez de estarem na subpasta Widget. Terás de especificar a subpasta como parte do URL, `http://mydomain.com/svn/trunk/Widget-X`. De notar que o comando importar irá criar as subpastas automaticamente no repositório, caso elas não existam.

A mensagem de importar é utilizada como mensagem de registo.

Por defeito, ficheiros e pastas que correspondem aos padrões globais para ignorar arquivos *não* são importados. Para substituir este comportamento, poderás usar a caixa de verificação **Incluir arquivos ignorados**. Consultar em [Secção 4.30.1, “Preferências Gerais”](#), para mais informações em como introduzir um padrão de arquivos a ignorar.

Assim que premires OK TortoiseSVN importa a arvore de pastas completa, incluindo todos ficheiros, para o repositório. O projecto está agora armazenado, sob controlo de versões, no repositório. Note-se que a pasta que importaste *NÃO* está sob controlo de versões. Para obter uma *cópia de trabalho* com controle de versões, necessitarás de efectuar um SVN Exportar da versão que acabaste de importar. Ou lê, para descobrir como importar uma pasta no local.

4.2.2. Importar no local

Assumindo que já tens um repositório e queres adicionar uma pasta á sua estrutura, segue estes passos:

1. Usa o navegador de repositório para criar uma nova pasta de projecto, directamente no repositório.
2. SVN exporta a nova pasta acima da pasta que queres importar. Irás receber um aviso que a pasta local não está vazia. Agora tens uma pasta de topo versionada com conteúdo não versionado.
3. Usa TortoiseSVN → Adicionarmenuitem> nesta pasta versionada para adicionar algum ou todo o seu contesvn:ignore em pastas e fazer qualquer outra alteraSubmete a pasta de topo,e ter

4.2.3. Ficheiros Especiais

Às vezes é necessário ter um ficheiro sob o controlo de versões que contém dados específicos, isto é, tens um ficheiro que cada desenvolvedor/utilizador necessita modificar para satisfazer o seu/sua configuração. Mas o versionamento de tal ficheiro é difícil porque, cada utilizador iria, a todo o momento, submeter as suas alterações ao repositório.

Em alguns casos sugerimos o uso de ficheiros *template*. Cria um ficheiro que contém os dados que os teus desenvolvedores precisam, adiciona este ficheiro ao controlo de versões e deixa os desenvolvedores exportar este ficheiro. Então, cada desenvolvedor tem que *fazer uma cópia* desse ficheiro e renomear essa cópia. Após isto, modificar a cópia já não é um problema.

Como um exemplo, podes olhar para o script de construção do TortoiseSVN. Ele chama um ficheiro chamado `TortoiseVars.bat` que não existe no repositório, só o ficheiro `TortoiseVars.tmpl`. O `TortoiseVars.tmpl` é um ficheiro template que cada desenvolvedor tem de criar uma cópia de, e renomeá-lo para `TortoiseVars.bat`. No interior desse ficheiro, adicionámos comentários para que os utilizadores vejam quais as linhas que têm de editar e mudar de acordo com a sua configuração, de forma a faze-la funcionar.

Então, para não perturbar os utilizadores, nós adicionámos também o ficheiro `TortoiseVars.bat` á lista de arquivos a ignorar do sua pasta mãe, i.e colocámos esse ficheiro na propriedade `svn:ignore`. Deste modo não aparecerá como ficheiro não versionado a cada submissão.

4.3. SVN Exportar Para Uma Cópia de Trabalho

Para obter uma cópia de trabalho necessitas de efectuar primeiro um *SVN Exportar* de um repositório.

Seleccionar uma pasta no explorador do windows onde queres colocar a tua cópia de trabalho. Clicar com o botão direito para aparecer o menu de contexto, e selecciona o comando TortoiseSVN → SVN Exportar..., que mostrará a seguinte caixa de diálogo:

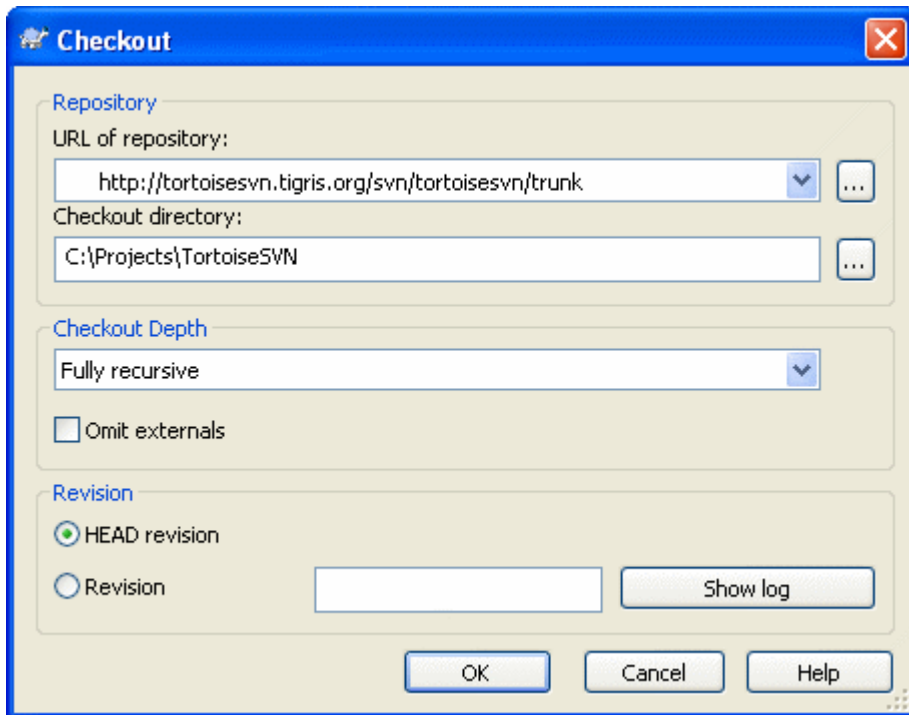


Figura 4.7. A caixa de diálogo Checkout

Se introduzires um nome de pasta que ainda não existe, uma pasta com esse nome será criada.

4.3.1. Profundidade do Checkout

Podes escolher a *profundidade* que queres SVN Exportar, o que te permite especificar a profundidade da recursividade nas pastas filho. Se quiseres apenas algumas secções de uma grande árvore, podes fazer SVN Exportar apenas da pasta de topo e depois actualizar recursivamente pastas seleccionadas.

Totalmente recursiva

Efectuar checkout a árvore inteira, incluindo todas as pastas filho e subpastas.

filhos directos, incluindo pastas

Efectuar checkout a pasta especificada, incluindo todos os ficheiros e pastas filho, mas não povoar as pastas filho.

Só ficheiros filho

Efectuar checkout a pasta especificada, incluindo todos os ficheiros mas não efectuar checkout qualquer pasta filho.

Só este item

Efectuar checkout só a pasta. Não povoa-la com ficheiros ou pastas filho.

Cópia de trabalho

Reter a profundidade especificada na cópia de trabalho. Esta opção não é usada na caixa de diálogo do SVN exportar, mas é a opção por defeito para todas as outras caixas de diálogo que têm a configuração de profundidade.

Excluir

Usado para reduzir a cópia de trabalho, depois de a pasta já ter sido povoada. Esta opção só está disponível na caixa de diálogo **Actualizar para a revisão**.

Se SVN exportares para uma cópia de trabalho parcial (i.e, escolhendo outra opção para além da *totalmente recursiva* relativamente á profundidade de exportação), poderás resgatar subpastas adicionais, utilizando o navegador do repositório ([Secção 4.24, “O Navegador de Repositório”](#)) ou a caixa de diálogo verificar modificações ([Secção 4.7.3, “Estado Remoto e Local”](#)).

No navegador de repositório, clicar com o botão direito na pasta SVN exportada e usar o TortoiseSVN → **Navegador de Repositório** para mostrar o navegador de repositório. Seleccionar a subpasta que gostarias de adicionar à tua cópia de trabalho e utilizar, **Context menu** → **Actualizar item para revisão...** Esse menu só será visível, se o item seleccionado ainda não existir na tua cópia de trabalho mas o item pai já exista.

Na caixa de diálogo verificar alterações, clicar primeiro no botão **Verificar repositório**. A caixa de diálogo mostrará todos os ficheiros e pastas que estão no repositório, mas não foram exportadas como, *adicionada remotamente*. Clicar com o botão direito na pasta(s) que gostarias de adicionar á tua cópia de trabalho e, usa **Menu de contexto** → **Actualizar**.

Esta funcionalidade é muito útil quando só queres SVN exportar partes de uma árvore grande, mas desejas a conveniência de actualizar uma só cópia de trabalho. Supõem que tens uma grande árvore que tem as subpastas `Project01` até `Project99`, e só queres SVN exportar `Project03`, `Project25` e `Project76/SubProj`. Utiliza estes passos:

1. SVN exportar a pasta pai com a profundidade “Só este item”. Agora tens uma pasta de topo vazia.
2. Seleccionar a nova pasta e usar TortoiseSVN → **Navegador de repositório** para visualizar o conteúdo do repositório.
3. Clicar com o botão direito no `Project03` e **Menu contexto** → **Actualizar item para revisão...** Mantém as configurações por defeito e clica em OK. Agora tens essa pasta totalmente povoada.

Repetir o mesmo processo para o `Project25`.

4. Navegar para o `Project76/SubProj` e fazer o mesmo. Desta vez verifica que a pasta `Project76` não tem conteúdo á excepção do `SubProj`, que por si está totalmente povoada. O Subversion criou as pastas intermediárias por ti, mas sem as povoar.



Modificando a profundidade da cópia de trabalho

Uma vez que tenhas SVN exportado uma cópia de trabalho, com uma profundidade particular, poderás modificar a mesma mais tarde para obter maior ou menor conteúdo utilizando **Context menu** → **Actualizar item para revisão...**



Usando um servidor mais antigo

Servidores anteriores à versão 1.5, não compreendem o pedido de profundidade da cópia de trabalho não lidando com eficiência com os pedidos. O comando continuará a funcionar, mas um servidor antigo poderá enviar todos os dados, deixando o cliente filtrar o que não precisa o que poderá gerar muito tráfico de rede. Se possível deverás actualizar o teu servidor para a versão 1.5.

Se o projecto contiver referências a projectos externos, que *não* pretendas SVN exportar ao mesmo tempo, usa a caixa de verificação **Omitir externos**



Importante

Se Omitir externos está verificada, ou se desejas aumentar o valor de profundidade, tens de efectuar actualizações á cópia de trabalho utilizando TortoiseSVN → Actualizar para revisão... em vez de TortoiseSVN → Actualizar. A actualização padrão irá incluir todos os externos e manter a profundidade existente.

É recomendado que SVN exportes só até à parcela trunk da árvore de pastas. Se especificares o caminho pai da árvore de pastas no URL poderás acabar com um disco rígido cheio, visto que, terás uma cópia inteira da árvore do repositório incluíndo cada ramo e etiqueta do teu projecto!



Exportando

Por vezes quererás criar uma cópia local sem nenhuma das pastas .svn, i.e. criar uma tarball zipada do teu código fonte. Lê [Secção 4.26, “Exportar uma Cópia de Trabalho do Subversion”](#) para descobrires como o fazer.

4.4. Submetendo as tuas alterações para o Repositório

Enviar as alterações que fizeste, na tua cópia de trabalho, é conhecida como *submeter* as alterações. Mas antes de as submeteres, tens de ter a certeza que a tua cópia de trabalho se encontra actualizada. Poderás usar directamente TortoiseSVN → Actualizar ou usar primeiro TortoiseSVN → Verificar alterações, para ver que ficheiros foram alterados localmente ou no servidor.

4.4.1. A Caixa de Diálogo Submeter

Se a tua cópia de trabalho se encontra actualizada e não existem conflitos, então estás em condições para submeter as tuas alterações. Seleccionar os ficheiros e/ou pastas que queres submeter e, TortoiseSVN → Submeter....

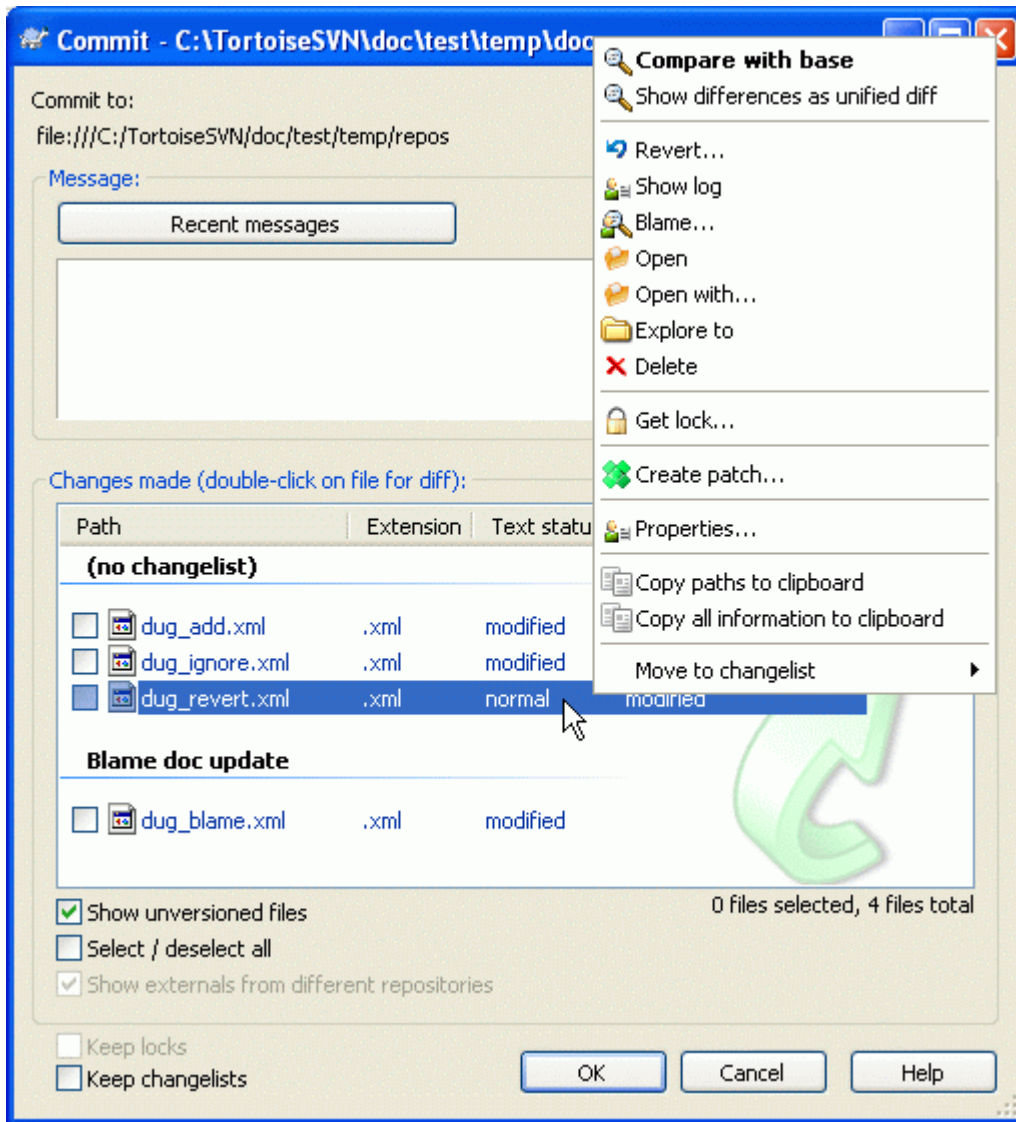


Figura 4.8. A Caixa de Diálogo Submeter

A caixa de diálogo submeter mostra-te cada ficheiro alterado, incluindo os adicionados, apagados e não versionados. Se não queres que um ficheiro alterado seja submetido, limpa a caixa de verificação desse ficheiro. Se quiseres adicionar um ficheiro não versionado, então verifica esse ficheiro, para o adicionar á submissão.

Itens que foram comutados para um repositório diferente são também indicados com o uso do marcador (s). Poderás ter comutado algo enquanto trabalhavas num ramo, e esquecido de comutar de novo para o trunk. Isto é um sinal de aviso!



Submeter ficheiros ou pastas?

Quando submetes ficheiros, a caixa de diálogo de submeter mostrará só os ficheiros seleccionados. Quando submetes uma pasta, a caixa de diálogo submeter irá seleccionar os ficheiros alterados automaticamente. Se te esqueceres de um novo ficheiro criado ao submeter a pasta, ele será encontrado. Submetendo a pasta *não* significa que todos os ficheiros serão marcados como alterados; Apenas significa que será feito mais trabalho por ti, facilitando-te a vida.

Se alteraste ficheiros que foram incluídos a partir de um repositório diferente, usando `svn:externals`, essas alterações não poderão ser incluídas na mesma submissão atómica. Um símbolo de aviso, por baixo do ficheiro, dir-te-á se isto aconteceu. Uma etiqueta de dica explicará que esses ficheiros externos têm de ser submetidos separadamente.



Muitos ficheiro não versionados na caixa de submissão

Se sentires que a caixa de diálogo de submissão mostra muitos ficheiros não versionados (e.g. gerados pelo compilador ou backups do editor), existem várias maneiras para lidar com esta situação. Poderás:

- Adicionar o ficheiro (ou uma extensão de carácter de substituição) à lista de ficheiros a excluir, na página de configurações. Isto irá afectar todas as cópias de trabalho que tens.
- Adicionar o ficheiro à lista `svn:ignore` usando TortoiseSVN → Apaga e adiciona à lista de ignorados. Isto só afectará a pasta onde foi alterada a propriedade `svn:ignore`. Usando a Caixa de Diálogo de Propriedades SVN, poderás alterar a propriedade `svn:ignore` para a pasta.

Ler [Secção 4.13, “Ignorando Ficheiros E Pastas”](#) para mais informação.

O Duplo clique na caixa de diálogo submeter, em qualquer ficheiro alterado, irá lançar a ferramenta externa diff para mostrar as tuas alterações. O menu de contexto irá dar-te mais opções, como pode ser visto no screenshot. Poderás também arrastar ficheiros daqui para outra aplicação, como um editor de texto ou um IDE.

Poderás seleccionar ou desseleccionar items clicando na caixa de verificação à esquerda do item. Para pastas poderás usar **Shift**-Seleccionar para tornar essa acção recursiva.

As colunas mostradas no painel de fundo são customizáveis. Se clicares com o botão direito em qualquer cabeçalho de coluna verás um menu de contexto, que te permitirá seleccionar quais as colunas que serão mostradas. Poderás também alterar a largura da coluna utilizando um indicador de arrasto, que aparecerá quando moveres o rato sobre os limites da coluna. Estas customizações são conservadas para que as vejas os mesmos cabeçalhos da próxima vez.

Por defeito ao submeteres as tuas alterações, são libertados automaticamente quaisquer bloqueios que tiveres sobre ficheiros após o sucesso da submissão. Se desejares manter esses bloqueios tem a certeza que a caixa de verificação **Manter bloqueios**. Por defeito o estado dessa caixa de verificação é retirada a partir da opção `no_unlock`, no ficheiro de configuração do Subversion. Ler [Secção 4.30.1, “Preferências Gerais”](#) para mais informações em como editar o ficheiro de configuração do Subversion.



Arrastar e Largar

Podes arrastar ficheiros de qualquer proveniência para a caixa de diálogo Submeter desde que, as cópias de trabalho estejam exportadas a partir do mesmo repositório. Por exemplo, tu podes ter uma gigantesca cópia de trabalho, com várias janelas de explorador abertas para visualizar distantes pastas na hierarquia. Se desejares evitar submeter a partir da pasta de topo (com uma pesquisa lenta da pasta em procura por alterações), podes abrir a caixa de diálogo de submeter para uma pasta e arrastar itens de outras janelas, para os assim incluir na mesma submissão atómica.

Podes arrastar, para a caixa de diálogo de submeter, ficheiros não versionados que residem na cópia de trabalho e eles serão adicionados automaticamente ao SVN.



Reparando Alterações de Nome Externas

Por vezes, é alterado o nome de alguns ficheiros fora do Subversion e, eles são mostrados na lista de ficheiros como um em falta e outro ficheiro não versionado. Para evitar perder o histórico necessitas de notificar o Subversion a cerca da ligação. Simplesmente selecciona o de nome antigo(em falta) e o de novo nome (não versionado) e para emparelhar os dois ficheiros como uma alteração de nome usa Context Menu → Reparar Movimento

4.4.2. Listas de Alterações

A caixa de diálogo de submissão suporta a funcionalidade listas de alterações, para ajudar a agrupar ficheiros. Sabes mais acerca desta funcionalidade em [Secção 4.8, “Listas de Alterações”](#).

4.4.3. Excluir itens da lista a Submeter

Por vezes, tens ficheiros versionados que são alterados frequentemente mas que não desejas submeter. Por vezes isso indica uma falha no processo de construção - porque são estes ficheiros versionados? devias ter usado ficheiros template? Mas por vezes é inevitável. A razão clássica é a mudança, pelo IDE, da marca temporal do ficheiro de projecto, de cada vez que o constróis. O ficheiro de projecto tem de ser versionado pois inclui todas as configurações de construção, mas no entanto não precisa de ser submetido só porque a sua marca temporal foi alterada.

Para ajudar em casos estranhos como este, temos reservada uma lista de alterações chamada `ignore-on-commit`. Qualquer ficheiro adicionado a esta lista de alterações será automaticamente não seleccionada na caixa de diálogo de submissão. Poderás ainda submeter as alterações mas, terás de as seleccionar manualmente na caixa de diálogo de submissão.

4.4.4. Mensagens de Registo de Submeter

Tem a certeza de, inserir uma mensagem de registo que descreve as alterações que vais submeter. Isto ir-te-á ajudar a visualizar o que aconteceu e quando, enquanto navegas através das mensagens de registo do projecto, numa data posterior. A mensagem pode longa ou breve, conforme o desejares; muitos projectos têm guias para o que deverá ser incluído, a linguagem a usar, e por vezes até um formato rígido.

Podes aplicar um formato simples às tuas mensagens de registo, usando uma convenção similar à usada nos emails. Para aplicar estilo ao texto usar; `*text*` para negrito, `_text_` para sublinhado, e `^text^` para itálico.

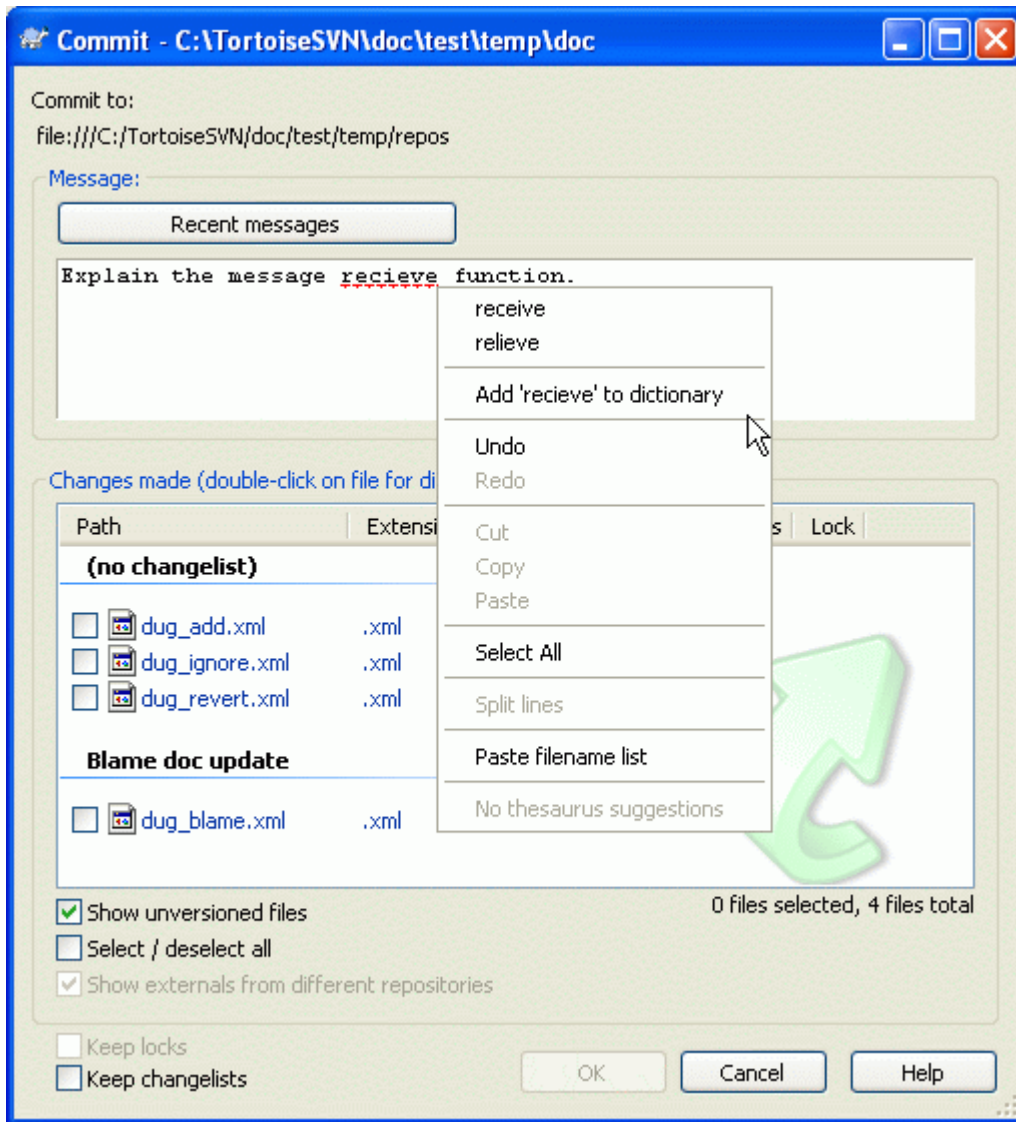


Figura 4.9. A Verificação de Sintaxe na Caixa de Diálogo Submeter

O TortoiseSVN inclui um verificador de sintaxe que te ajuda a conseguir obter mensagens de registo bem escritas. Este irá sobressair qualquer palavra mal escrita. Usa o menu de contexto para aceder às correcções sugeridas. É claro que ele não conhece *todos* os termos técnicos que utilizas, podendo aparecer como erros termos correctamente escritos. Não te preocupes, poderás os adicionar ao teu dicionário pessoal, usando o menu de contexto.

A janela de mensagem de registo também inclui a funcionalidade de auto conclusão de nomes de ficheiros e funções. Esta funcionalidade usa expressões regulares para extrair os nomes de funções e classes, a partir dos ficheiros (de texto) que estás a submeter tal como os nomes dos ficheiros em si. Se a palavra que estás a escrever condiz com qualquer entrada na lista (depois de escreveres pelo menos 3 caracteres, ou premido **Ctrl+Space**), uma lista de termos aparecerá para te permitir seleccionar o nome completo. As expressões regulares fornecidas com o TortoiseSVN estão armazenadas na pasta bin da instalação do TortoiseSVN. Podes também definir as tuas próprias regexes e armazená-las em %APPDATA%\TortoiseSVN\autolist.txt. Está claro que a tua auto lista privada não será reescrita quando actualizas a tua instalação do TortoiseSVN. Se não estás familiarizado com expressões regulares, dá uma vista de olhos na introdução em http://pt.wikipedia.org/wiki/Expressão_regular [http://en.wikipedia.org/wiki/Regular_expression], e o tutorial e documentação online em <http://www.regular-expressions.info/>.

Podes reutilizar mensagens inseridas previamente. clica apenas em **Mensagens recentes** para ver a lista de algumas das últimas mensagens que inseriste, nesta cópia de trabalho. O número de mensagens guardadas pode ser customizável na caixa de diálogo de configuração do TortoiseSVN.

Podes limpar todas a mensagens de registo de submissão armazenadas na página **Dados Guardados** de configurações do TortoiseSVN, ou poderás limpar mensagens individuais a partir da caixa de registo **Mensagens recentes** usando a tecla **Delete**.

Se queres incluir, na mensagem de registo, os caminhos seleccionados, podes usar no controlo de edição o comando **Context Menu de Contexto** → **Colar lista de nomes de arquivos**.

Outra maneira para inserir caminhos na mensagem de registo é simplesmente, arrastar os ficheiros da lista de ficheiros para o controlo de edição.



Propriedades de Pastas Especiais

Existem várias propriedades de pastas especiais que podem ser usadas para dar mais controlo sobre a formatação das mensagens de registo de formatação, e a língua usada para o módulo de verificação ortográfico. Ler [Secção 4.17, “Configurações de Projecto”](#) para mais informação.



Integração com Ferramentas de Controlo de Problemas

Se activaste o sistema de controlo de problemas, podes introduzir um ou mais Problemas na caixa de texto **ID-Problema / Nr-Assunto**:. Múltiplos assuntos devem ser separados por vírgulas. Em alternativa, se estiveres a usar o suporte para controlo de problemas baseado em regex, adiciona apenas as tuas referências de assunto como parte da mensagem de registo. Sabe mais em [Secção 4.28, “Integração com Sistemas de controlo de problemas/ Gestores de Problemas”](#).

4.4.5. Progresso do Submeter

Após premir o **OK**, aparece a caixa de diálogo que mostra o progresso da submissão.

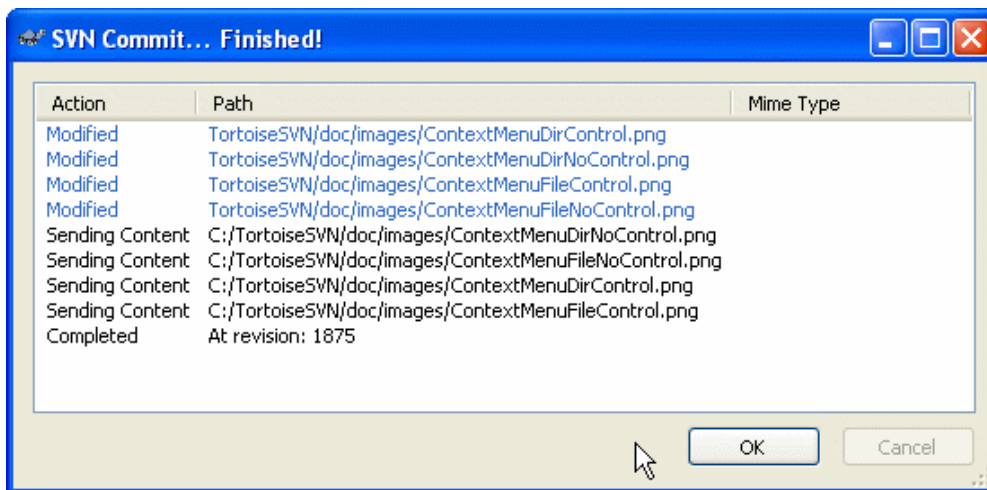


Figura 4.10. A caixa de dialogo de Progresso, mostrando a submissão em progresso

A caixa de diálogo de progresso usa um código de cores para evidenciar as acções de submissão

Azul

Submetendo uma alteração.

Roxo

Submetendo uma nova adição.

Vermelho escuro

Submetendo uma substituição ou apagamento.

Preto

Todos os outros itens.

Este é o esquema de cores por defeito, mas podes alterar essas cores usando a caixa de diálogo configurações. Ler [Secção 4.30.1.4, “Preferências de cor do TortoiseSVN”](#) para mais informação.

4.5. Actualizar a Tua Cópia de Trabalho Com Alterações de Outros

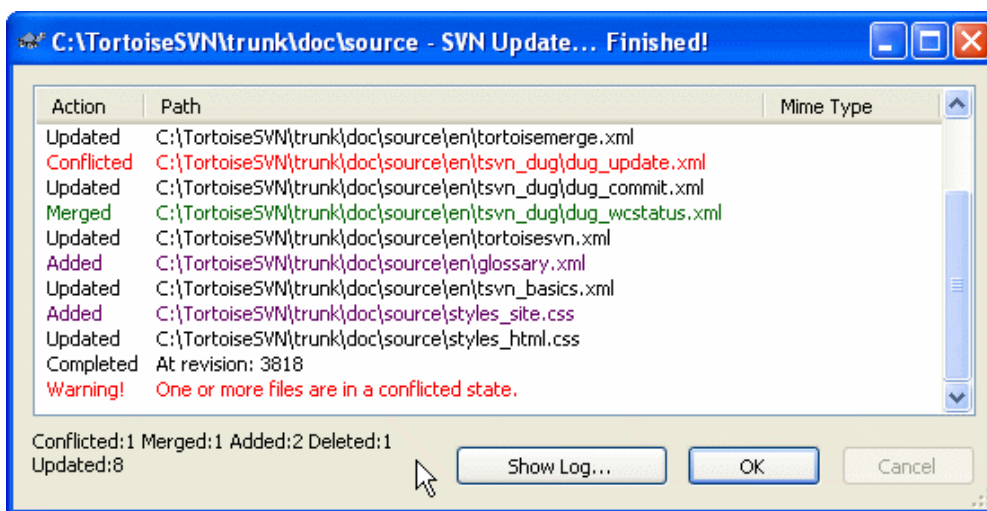


Figura 4.11. Caixa de diálogo de progresso mostrando uma actualização finalizada.

Periodicamente, deverás garantir que as alterações feitas por outros são incorporadas na tua cópia de trabalho local. O processo de obter as alterações do servidor para a cópia local é conhecido como *actualizando*. Actualização pode ser feita em ficheiros isolados, um conjunto de ficheiros seleccionados, recursivamente numa estrutura de hierarquia de pastas. Para actualizar selecciona os ficheiros e/ou pastas que desejas, clica com o botão direito e selecciona TortoiseSVN → Actualizar no menu de contexto do explorador. Uma janela aparecerá mostrando o progresso da actualização, enquanto ela decorre. As alterações feitas por outros serão fundidas com os teus ficheiros, mantendo quaisquer alterações que tenhas feito nos mesmos. O repositório *não* é afectado pela actualização.

A caixa de diálogo de progresso usa um código de cores para evidenciar diferentes acções de actualização

Roxo

Novo item adicionado à tua CT.

Vermelho escuro

Item redundante apagado da tua CT, ou um item em falta substituído na tua CT.

Verde

Alterações do repositório fundidas com sucesso, com tuas alterações locais.

Vermelho brilhante

Alterações do repositório fundidas com as alterações locais, resultando em conflitos que é necessário resolver.

Preto

Item não alterado, na tua CT, actualizado com uma nova versão do repositório.

Este é o esquema de cores por defeito, mas podes alterar essas cores usando a caixa de diálogo configurações. Ler [Secção 4.30.1.4, “Preferências de cor do TortoiseSVN”](#) para mais informação.

Se tiveres alguns *conflitos* durante uma actualização (o que pode acontecer se outros alterarem as mesmas linhas nos mesmos ficheiros que tu alteraste, e essas alterações não condizerem), então a caixa de diálogo mostra esses conflitos a vermelho. Podes efectuar clicar duplo clique nessas linhas para iniciar a ferramenta externa de fusão, para resolver esses conflitos.

Quando a actualização está concluída a caixa de diálogo de progresso mostra, debaixo da lista de ficheiros, o resumo do números de itens actualizados, adicionados, apagados, em conflito, etc. Esta informação de sumário pode ser copiada para o clipboard usando **Ctrl+C**.

O comando de Actualização padrão não tem opções e apenas actualiza a tua cópia de trabalho para a revisão HEAD do repositório, que é o caso de utilização mais comum. Se quiseres mais controlo sobre o processo de actualização deverás usar em alternativa TortoiseSVN → Atualizar para revisão.... Este te permite actualizar a tua cópia de trabalho para uma revisão específica, e não só para a mais recente. Supõem que a tua cópia de trabalho está na revisão 100, mas tu queres que reflecta o estado que tinha na revisão 50 - então simplesmente actualiza para a revisão 50. Na mesma caixa de diálogo também podes escolher a *profundidade* a que deve ser actualizada a pasta corrente. Os termos usados estão descritos em [Secção 4.3.1, “Profundidade do Checkout”](#). A profundidade por defeito é Cópia de trabalho, que preserva a configuração de profundidade existente. Podes também escolher se deves ignorar qualquer projecto externo, na actualização (i.e. projectos referenciados usando `svn:externals`).



Cuidado

Se actualizas um ficheiro ou uma pasta para uma revisão específica não deverás efectuar alterações nos mesmos. Obterás mensagens de erro “desactualizado” quando tentares submete-las! Se quiseres desfazer as alterações efectuadas num ficheiro ou começar de novo a partir de uma revisão anterior, podes rebobinar para uma revisão anterior a partir da caixa de diálogo de registo de revisões. Dá uma vista de olhos em [Secção B.4, “Reverter \(Anular\) revisões no repositório”](#) para mais instruções e métodos alternativos.

Atualizar para revisão pode ocasionalmente ser útil para ver qual o aspecto que tinha o teu projecto num ponto anterior da sua história. Mas na generalidade, actualizar ficheiros individuais para uma versão anterior não é uma boa ideia, pelo que deixa a tua cópia de trabalho num estado inconsistente. Se o ficheiro que estás a actualizar mudou de nome, poderás constatar que o mesmo desaparece da tua cópia de trabalho porque, nenhum ficheiro com esse nome existia na revisão anterior. Deves também constatar que o item mostrará uma sobreposição verde normal, pelo que será indistinguível dos ficheiros que estão actualizados.

Se queres simplesmente uma cópia local de uma versão antiga de um ficheiro é melhor utilizar, a partir da caixa de diálogo do registo, o comando Menu Contexto Guardar revisão em...



Múltiplos Ficheiros/Pastas

Se seleccionares múltiplos ficheiro e pastas no explorador e então seleccionares Actualizar, todos esses ficheiros/pastas são actualizados um por um. O TortoiseSVN garante que todos os ficheiros/pastas que são do mesmo repositório são actualizados exactamente para a mesma revisão! Mesmo que entre essas actualizações outra submissão ocorra.



Ficheiro Local Já Existe

Por vezes quando tentas actualizar, a actualização falha com a mensagem que diz que já existe um ficheiro local com o mesmo nome. Isto tipicamente ocorre quando o Subversion

tenta SVN exportar um ficheiro versionado recentemente e, descobre que um ficheiro, não versionado, com o mesmo nome já existe na pasta de trabalho. O Subversion nunca reescreve um ficheiro não versionado - pode conter algo importante em que estás a trabalhar e que por coincidência, tem o mesmo nome que outro programador utilizou para o seu ficheiro submetido recentemente.

Se obtiveres esta mensagem de erro, a solução é simplesmente renomear o ficheiro local não versionado. Após a conclusão da actualização podes verificar se o ficheiro renomeado é ainda necessário.

Se continuares a obter mensagens de erro, usa então TortoiseSVN → Verificar alterações para listares todos os ficheiros com problemas. Desta maneira poderás tratá-los todos de uma vez só.

4.6. Resolvendo Conflitos

De vez quando irás obter um *conflito* quando actualizares/fundires os teus ficheiros a partir do repositório, ou quando trocares a tua cópia de trabalho para um URL diferente. Existem dois tipos de conflitos:

conflitos de ficheiro

Um conflito de ficheiro ocorre quando dois (ou mais) programadores alteraram o mesmo punhado de linhas do ficheiro.

conflitos de árvore

Um conflito de árvore ocorre quando um programador move/renomeia/apaga um ficheiro ou pasta, que outro programador também moveu/renomeou/apagou ou apenas modificou.

4.6.1. Conflitos de Ficheiro

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:

filename.ext.mine

This is the file that was theEste é o teu ficheiro como existia na cópia de trabalho antes de a teres actualizado - Isto é, sem marcadores de conflito. Este ficheiro tem as tuas últimas alterações a mais nada.

filename.ext.rOLDREV

Este é o ficheiro que foi a revisão BASE antes de actualizares a cópia de trabalho. Isto é, é o ficheiro que tu SVN exportaste antes de efectuares as tuas últimas alterações.

filename.ext.rNEWREV

Este é o ficheiro que o teu cliente Subversion acabou de receber do servidor, quando actualizaste a tua cópia de trabalho. Este ficheiro corresponde à revisão HEAD do repositório.

Podes então lançar uma ferramenta externa de fusão/editor de conflitos com TortoiseSVN → Editar Conflitos ou podes usar qualquer outro editor para resolver o conflito manualmente. Deves decidir a aparência do código, efectuar as alterações necessárias e guardar o ficheiro.

Posteriormente executa o comando TortoiseSVN → Resolvido e submete as tuas modificações para o repositório. Toma nota que o comando Resolvido, na realidade, não resolve o conflito. Apenas remove os ficheiros `filename.ext.mine` and `filename.ext.r*`, para te permitir submeter as tuas alterações.

Se tens conflitos com os teus ficheiros binários, o Subversion não tenta fundir os ficheiros. O ficheiro local permanece intocável (exactamente como o alteraste) e obtens os ficheiros `filename.ext.r*`. Se queres descartar as tuas alterações e manter a versão do repositório, usa o comando Reverter. Se queres manter a tua versão e reescrever a versão do repositório, usa o comando Resolvido e submete a tua versão.

Podes usar o comando Resolvido para ficheiros múltiplos, se clicares na pasta pai e seleccionares TortoiseSVN → Resolvido... Isto irá mostrar a caixa de diálogo listando todos os ficheiros em conflito nessa pasta, e podes seleccionar quais é queres marcar como resolvidos.

4.6.2. Conflitos de Árvore

Um conflito de árvore ocorre quando, um programador moveu/renomeou/apagou um ficheiro ou pasta que outro programador também moveu/renomeou/apagou ou apenas modificou. Existem muitas e variadas situações que podem gerar um conflito de árvore, e todas elas requerem diferentes passos para a sua resolução.

Quando no Subversion, um ficheiro é apagado localmente esse ficheiro também é apagado no sistema de ficheiros local e, mesmo que este seja parte de um conflito de árvore não poderá mostrar um ícone de conflito sobreposto, e tu não poderás clicar nele com o botão direito do rato para resolver o conflito. Usa então a caixa de diálogo Verificar alterações para aceder à opção Editar conflito.

O TortoiseSVN pode ajudar a encontrar o sítio correcto para fundir as alterações mas, poderá ser necessário trabalho adicional para resolver os conflitos. Lembra-te que após a actualização a revisão BASE de trabalho irá conter sempre a revisão de cada item como estava no repositório, na altura da actualização. Se reverteres uma alteração após a actualização esta voltará ao estado do repositório e não, à versão que estava quando começaste a efectuar as tuas alterações locais.

4.6.2.1. Apagar localmente, editar após actualizar

1. O programador A modifica `Foo.c` e submete-o para o repositório
2. O programador B simultaneamente move na sua cópia de trabalho `Foo.c` para `Bar.c`, ou simplesmente apagou `Foo.c`, ou a sua pasta pai.

Uma actualização na cópia de trabalho do programador B resulta num conflito de árvore:

- `Foo.c` foi apagado da cópia de trabalho, mas foi marcado como um conflito de árvore.
- Se o conflito resultou de, uma mudança de nome em vez de um apagar então, o `Bar.c` é marcado como adicionado mas não contém as modificações do programador A.

O programador B tem de escolher se vai manter as alterações do Programador A. No caso de uma mudança no nome do ficheiro, ele pode fundir as alterações com o `Foo.c` no ficheiro renomeado `Bar.c`. Para simples apagamentos de ficheiros e pastas, ele pode escolher manter o item com as alterações do Programador A e descartar o item apagado. Ou, ao colocar o conflito como resolvido, sem efectuar mais nenhuma acção, descarta efectivamente as alterações do Programador A.

A caixa de diálogo de edição de conflitos oferece-se para fundir as alterações, se conseguir encontrar o ficheiro original do renomeado `Bar.c`. Dependendo onde a actualização foi invocada, poderá não ser possível encontrar o ficheiro fonte.

4.6.2.2. Edição local, apagar após actualizar

1. O Programador A move `Foo.c` para `Bar.c` e submete-o para o repositório.
2. O Programador B modifica na sua cópia de trabalho `Foo.c`.

Ou em caso de mover uma pasta...

1. O Programador A move a pasta pai `FooFolder` para `BarFolder` e submete-o para o repositório.
2. O Programador B modifica na sua cópia de trabalho `Foo.c`.

Uma actualização da cópia de trabalho pelo programador B, resulta num conflito de árvore. Para um conflito de ficheiro simples:

- `Bar.c` é adicionado à cópia de trabalho como um ficheiro normal.
- `Foo.c` é marcado como adicionado (com história) e adquire um conflito de árvore.

Para um conflito de pasta:

- `BarFolder` é adicionado à cópia de trabalho como uma pasta normal.
 - `FooFolder` é marcado como adicionado (com história) e adquire um conflito de árvore.
- `Foo.c` é marcado como modificado.

O programador B agora tem de decidir se quer manter a reorganização do programador A e fundir as suas alterações no ficheiro correspondente na nova estrutura, ou simplesmente reverter as alterações de A e manter o ficheiro local.

Para fundir as suas alterações locais com a remodelação, o Programador B deve em primeiro lugar descobrir o ficheiro para o qual `Foo.c` foi renomeado/movido no repositório. Isso pode ser feito usando a caixa de diálogo de registo. As alterações devem então ser fundidas manualmente já que não existe actualmente nenhum método para automatizar ou mesmo simplificar este processo. Uma vez que as alterações tenham sido portadas, o caminho em conflito fica redundante e pode ser apagado. Neste caso usa o botão **Remove** na caixa de diálogo de edição de conflito para efectuar a limpeza e, marcar o conflito como resolvido.

Se o programador B entende que as alterações de A estão erradas, então deve escolher o botão **Manter** na caixa de diálogo de conflito. Isso marca o ficheiro/pasta em conflito como resolvido, mas as alterações de A necessitam de ser removidas à mão. Mais uma vez a caixa de diálogo de registo irá ajudar a verificar o que foi movido.

4.6.2.3. Apagar localmente, apagar após actualizar

1. O Programador A move `Foo.c` para `Bar.c` e submete-o para o repositório.
2. O Programador B move `Foo.c` para `Bix.c`

Uma actualização na cópia de trabalho do programador B resulta num conflito de árvore:

- `Bix.c` é marcado como adicionado com história.
- `Bar.c` é adicionado à cópia de trabalho com o estado 'normal'.
- `Foo.c` é marcado como apagado e adquire um conflito de árvore.

Para resolver este conflito, o Programador B tem de descobrir para que nome o ficheiro em conflito `Foo.c` renomeado/movido no repositório. Isso pode ser feito recorrendo à caixa de diálogo de registo.

Então o programador B tem de decidir qual o novo nome de ficheiro do ficheiro `Foo.c` deve manter - o criado pelo programador A ou o nome criado pelo próprio.

Depois de o programador B ter resolvido o conflito manualmente, o conflito de árvore tem de ser marcado como resolvido, com o botão na caixa de diálogo editor de conflito.

4.6.2.4. Desaparecido localmente, editar após fusão

1. O Programador A trabalhando no trunk modifica então `Foo.c` e submete-o para o repositório.

2. O Programador B trabalhando no ramo move Foo.c para Bar.c e submete-o para o repositório.

A fusão das alterações no trunk do programador A para o ramo, da cópia de trabalho, do programador B resulta num conflito de árvore:

- Bar.c já está na cópia de trabalho com o estado 'normal'.
- Foo.c é marcado como desaparecido, com um conflito de árvore.

Para resolver este conflito, o Programador B tem de marcar o ficheiro como resolvido na caixa de diálogo do editor de conflito, que irá remove-lo da lista de conflitos. Ele então tem de decidir se copia o ficheiro desaparecido Foo.c do repositório para a cópia de trabalho, se funde as alterações do Programador A no ficheiro Foo.c no renomeado Bar.c ou se ignora as alterações ao marcar o conflito como resolvido e não fazendo mais nada.

Tem em atenção que se copias o ficheiro desaparecido do repositório e depois marcas como resolvido, a tua copia será removida novamente. Tens de resolver primeiro o conflito.

4.6.2.5. Edição local, apagar após fusão

1. O Programador A trabalhando no trunk move o Foo.c para Bar.c e submete-o para o repositório

2. O Programador B trabalhando no ramo modifica o Foo.c e submete-o para o repositório.

Existe um caso equivalente para o movimento de pastas, mas não é ainda detectado no Subversion 1.6 ...

1. O Programador A trabalhando no trunk move a pasta pai FooFolder para BarFolder e submete-o para o repositório.

2. O Programador B trabalhando num ramo modifica Foo.c na sua cópia de trabalho

A fusão das alterações no trunk do programador A para o ramo, da cópia de trabalho, do programador B resulta num conflito de árvore:

- Bar.c é marcado como adicionado.
- Foo.c é marcado como modificado com um conflito de árvore.

O programador B agora tem de decidir se quer manter a reorganização do programador A e fundir as suas alterações no ficheiro correspondente na nova estrutura, ou simplesmente reverter as alterações de A e manter o ficheiro local.

Para fundir as suas alterações locais com a remodelação, o Programador B deve em primeiro lugar descobrir para que ficheiro o ficheiro em conflito Foo.c foi renomeado/movido no repositório. Isto pode ser feito usando a caixa de diálogo do registo para a fonte da fusão. O editor de conflito apenas mostra o registo para a cópia de trabalho e não conhece qual o caminho que foi utilizado na fusão, pelo que terás de o descobrir por ti. As alterações têm então ser fundidas à mão, visto não existir nenhuma maneira de automatizar ou mesmo simplificar este processo. Uma vez que as alterações tiverem sido portadas, o caminho em conflito será redundante e pode ser apagado. Neste caso usa o botão **Remove** na caixa de diálogo editor de conflito, para limpar e marcar o conflito como resolvido.

Se o Programador B decide que as alterações do A estão erradas, deverá então escolher o botão **Manter** na caixa de diálogo editor de conflito. Isto marca o ficheiro/pasta em conflito como resolvido, mas as alterações do Programador A necessitam de ser removidas à mão. De novo a caixa de diálogo registo, para a fonte da fusão ajuda a encontrar o rasto ao que foi movido.

4.6.2.6. Apagamento local, apagar após fusão

1. O Programador A trabalhando no trunk move o Foo.c para Bar.c e submete-o para o repositório

2. O Programador B trabalhando num ramo move o Foo.c para Bix.c e submete-o para o repositório

A fusão das alterações no trunk do programador A para o ramo, da cópia de trabalho, do programador B resulta num conflito de árvore:

- `Bix.c` é marcado com o estado normal (não modificado).
- `Bar.c` é marcado como adicionado com história.
- `Foo.c` é marcado como desaparecido e adquire um conflito de árvore.

Para resolver este conflito o Programador B tem de descobrir para que ficheiro o ficheiro em conflito `Foo.c` foi renomeado/movido no repositório. Isto pode ser feito usando a caixa de diálogo registo na fonte da fusão. O editor de conflito apenas mostra o registo para a cópia de trabalho, e não tem conhecimento qual o caminho que foi usado na fusão, então tens de descobri-lo por ti.

Então o programador B tem de decidir qual o novo nome de ficheiro do ficheiro `Foo.c` deve manter - o criado pelo programador A ou o nome criado pelo próprio.

Depois de o programador B ter resolvido o conflito manualmente, o conflito de árvore tem de ser marcado como resolvido, com o botão na caixa de diálogo editor de conflito.

4.7. Obter informação de Estado

Enquanto estás a trabalhar na tua cópia de trabalho necessitas com frequência de saber que ficheiros alteraste/adicionastes/removeste, ou mesmo que ficheiros foram alterados e submetidos por outros.

4.7.1. Sobreposição de Ícones

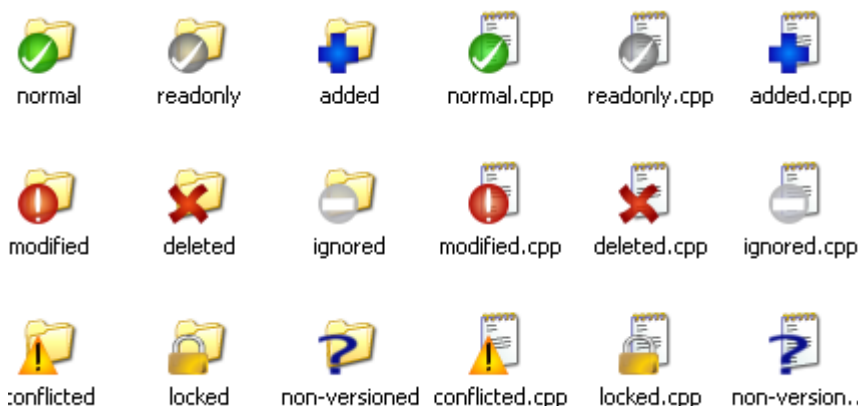


Figura 4.12. O explorador mostrando os ícones sobrepostos

Agora que SVN exportaste uma cópia de trabalho de um repositório Subversion, podes ver que os teus ficheiros com os ícones alterados no explorador do Windows. Esta é uma das razões porque o TortoiseSVN é tão popular. O TortoiseSVN adiciona a chamada sobreposição de ícone a cada ficheiro e que se sobrepõem ao ícone original do mesmo. Dependendo do estado Subversion do ficheiro a sobreposição do ícone será diferente.



Uma cópia de trabalho exportada de fresco possui uma marca de verificação verde como sobreposição. Isto significa que o estado Subversion é *normal*.



Assim que inicias a edição do ficheiro, o estado altera-se para *modificado* e o ícone de sobreposição muda para um ponto de exclamação vermelho. Desse modo podes facilmente ver que ficheiros foram alterados desde a última vez que actualizaste a tua cópia de trabalho e necessitam ser submetidos.



Se durante uma actualização ocorre um *conflito* o ícone de sobreposição altera-se para um ponto de exclamação amarelo.



Se colocaste a propriedade `svn:needs-lock` num ficheiro, o Subversion marca esse ficheiro como apenas-de-leitura até obteres um bloqueio nesse mesmo ficheiro. Ficheiros como esses têm essa sobreposição para indicar que tens de obter um bloqueio antes de editares esse ficheiro.



Se mantiveres um bloqueio num ficheiro e o estado Subversion é *normal*, esta sobreposição de ícone lembra-te que deves libertar o bloqueio se não o estás a usar, para permitir que outros possam submeter as suas alterações para o ficheiro.



Este ícone mostra que alguns ficheiros ou pastas no interior da pasta corrente foram escalonados para serem *apagados* do controlo de versões ou que, um ficheiro sob o controlo de versões está desaparecido na pasta.



O sinal de soma diz-te que um ficheiro ou pasta foi escalonado para ser *adicionado* ao controlo de versões.



O sinal de barra diz que um ficheiro ou pasta é *ignorado* para efeitos de controlo de versão. Esta sobreposição é opcional.



Este ícone mostra os ficheiros e pastas que não estão sob controle de versões, mas não foram ignorados. Esta sobreposição é opcional.

De facto, podes descobrir que nem todos esses ícones são usados no teu sistema. Isto ocorre porque o número de sobreposições permitidas pelo Windows é muito limitada e, se estiveres a utilizar uma versão antiga do TortoiseCVS, não existem suficientes slots para sobreposições disponíveis. O TortoiseSVN tenta ser um “Bom Cidadão (TM)” e limita o seu uso se sobreposições para também dar a oportunidade a outras aplicações.

Agora que existem por aí mais clientes Tortoise (TortoiseCVS, TortoiseHG, ...) o limite de ícones torna-se um verdadeiro problema. Para dar a volta a isto, o projecto TortoiseSVN introduziu um conjunto de ícones comuns e partilhados, carregados como uma DLL, que pode ser usada por todos os clientes Tortoise. Verifica no teu fornecedor de cliente para veres se isto já foi integrado :-)

Para uma descrição de como as sobreposições de ícones correspondem aos estados do Subversion e outros detalhes técnicos, lê [Secção F.1, “Sobreposição de Ícones”](#).

4.7.2. Colunas TortoiseSVN no Explorador do Windows

A mesma informação que está disponível a partir das sobreposições dos ícones (e muito mais) pode ser visualizada, como colunas adicionais, na vista de detalhes do Explorador do Windows.

Simplesmente clica com o botão direito em um dos cabeçalhos de uma coluna, escolhe Mais... a partir do menu de contexto. Uma caixa de diálogo irá aparecer, e poderás especificar as colunas e a sua ordem que serão mostradas na “Vista Detalhada”. Rolar para baixo até aparecer entradas a começar por SVN. Marcar as que gostarias de visualizar e fecha a caixa de diálogo pressionando OK. As colunas serão anexadas à direita das que mostradas correntemente. Podes reordena-las por arrasto e largar, para que satisfaçam as tuas necessidades.



Importante

As colunas adicionais no Explorador do Windows não estão disponíveis no Vista, visto que a Microsoft decidiu não permitir mais esse tipo de colunas para *todos* os ficheiros, mas só para tipos de ficheiros específicos.



Dica

Se queres que a disposição corrente seja visualizada em todas as tuas cópias de trabalho, deves querer usa-la como a tua vista por defeito.

4.7.3. Estado Remoto e Local

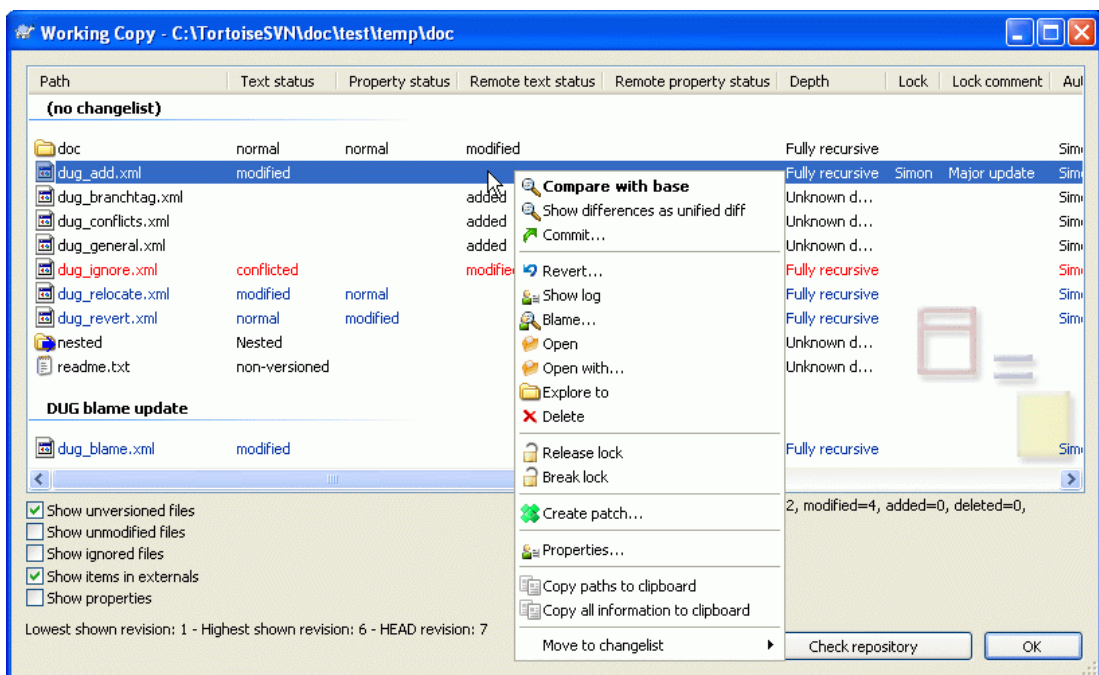


Figura 4.13. Verificar Modificações

É normalmente muito útil conhecer que ficheiros foram alterados e também quais os ficheiros que foram alterados e submetidos por outros. Isto é onde o comando TortoiseSVN → Verificar alterações... vem à mão. Esta caixa de diálogo irá mostrar-te cada ficheiro que foi alterado de algum modo na tua cópia de trabalho, tal como qualquer ficheiro não versionado que lá tiveres.

Se clicares em Verificar repositório podes então também procurar por alterações no repositório. Desse modo podes verificar, antes de uma actualização, se existe um possível conflito. Podes também actualizar ficheiros seleccionados do repositório, sem actualizar a pasta inteira. Por defeito o botão Verificar

repositório só procura o estado remoto com a profundidade de exportação da cópia de trabalho. Se queres ver todos os ficheiros e pastas no repositório, mesmos aqueles que não SVN exportaste, então tens de premir continuamente a tecla **Shift** enquanto clicas no botão **Verificar repositório**.

A caixa de diálogo usa código de cores para evidenciar o estado.

Azul

Itens modificados localmente.

Roxo

Itens adicionados. Itens que foram adicionados com história têm um sinal + na coluna **Estado do texto**, e uma etiqueta mostra de onde foi o item copiado.

Vermelho escuro

Itens apagados ou desaparecidos.

Verde

Itens modificados localmente e no repositório. As alterações serão fundidas aquando da actualização.

Vermelho brilhante

Itens modificados localmente e apagados no repositório, ou modificados no repositório e apagados localmente. Estes *irão* produzir conflitos na actualização.

Preto

Itens não modificados e não versionados.

Este é o esquema de cores por defeito, mas podes alterar essas cores usando a caixa de diálogo configurações. Ler [Secção 4.30.1.4, “Preferências de cor do TortoiseSVN”](#) para mais informação.

Itens que foram comutados para um repositório diferente são também indicados com o uso do marcador (S). Poderás ter comutado algo enquanto trabalhavas num ramo, e esquecido de comutar de novo para o trunk. Isto é um sinal de aviso!

A partir do menu de contexto da caixa de diálogo podes ver uma comparação das alterações. Marca as alterações locais que *tu* fizeste usando **Context Menu** → **Comparar com a Base**. Marca as alterações no repositório feitas por outros usando **Menu Contexto** → **Mostrar diferenças como uma comparação unificada**.

Podes também reverter alterações em ficheiro individuais. Se apagaste um ficheiro acidentalmente, irá aparecer como *Desaparecido* e podes usar *Reverter* para o recuperar.

Ficheiros não versionados e ignorados podem ser enviados daqui para o cesto de reciclagem usando **Menu Contexto** → **Remove**. Se quiseres remover permanentemente os ficheiros (ignorando o cesto de reciclagem) mantém premido a tecla **Shift** enquanto clicas no **Remove**.

Se quiseres examinar um ficheiro em detalhe, podes arrasta-lo para outra aplicação como um editor de texto ou um IDE.

As colunas são costumizáveis. Se clicares com o botão direito em qualquer cabeçalho de coluna verás um menu de contexto, que te permitirá seleccionar quais as colunas a mostrar. Podes também alterar a largura da coluna usando o manípulo de arrasto que aparece quando moves o rato sobre o limite da coluna. Estas customizações são preservados para que vejas os mesmos cabeçalhos da próxima vez.

Se tiveres a trabalhar ao mesmo tempo em várias tarefas, não relacionadas, podes também agrupar os ficheiros em listas de alterações. Ler [Secção 4.4.2, “Listas de Alterações”](#) para mais informação.

No fundo da caixa de diálogo podes ver um sumário do intervalo de revisões do repositório em uso na tua cópia de trabalho. Essas são as revisões *submetidas*, não as revisões de *actualizadas*; Estas representam o intervalo de revisões onde esses ficheiros foram submetidas por último, e não as revisões a que foram actualizadas. De notar que o intervalo de revisões mostrada só se aplica aos itens mostrados, e não, à

totalidade da cópia de trabalho. Se queres ver essa informação para a totalidade da cópia de trabalho, deves verificar a caixa de verificação **Não modificado**.



Dica

Se queres uma vista ampla da tua cópia de trabalho, i.e. mostrando todos os ficheiros e pastas em qualquer nível da hierarquia de pastas, então a caixa de diálogo **Verificar alterações** é a maneira mais fácil para o atingir. Verifica apenas a caixa de verificação **Não modificado** para mostrar todos os ficheiros na tua cópia de trabalho.



Reparando Alterações de Nome Externas

Por vezes, é alterado o nome de alguns ficheiros fora do Subversion e, eles são mostrados na lista de ficheiros como um em falta e outro ficheiro não versionado. Para evitar perder o histórico necessitas de notificar o Subversion a cerca da ligação. Simplesmente selecciona o de nome antigo(em falta) e o de novo nome (não versionado) e para emparelhar os dois ficheiros como uma alteração de nome usa **Context Menu** → **Reparar Movimento**

4.7.4. Vendo diferenças

Muitas vezes tu queres ver os teus ficheiros por dentro para observar o que alteraste. Podes conseguir isso ao seleccionar o ficheiro que foi alterado e, seleccionando **Comparar** a partir do menu de contexto do TortoiseSVN. Isso arranca o visualizador de comparações externo, que então irá comparar o ficheiro corrente com a cópia pura (revisão BASE), que foi armazenada depois da última exportação SVN ou actualização.



Dica

Mesmo quando fora da cópia de trabalho ou, quando tens multiplas versões do ficheiro presentes, podes no entanto ainda mostrar comparações:

Selecciona dois ficheiros que queres comparar no explorador (e.g. usando o **Ctrl** e o rato) e escolhe **Comparar** a partir do menu de contexto do TortoiseSVN. O ficheiro clicado por último (o que tem o focus, i.e. o rectangulo picotado) será considerado o mais recente.

4.8. Listas de Alterações

Num mundo ideal, apenas trabalharias numa única coisa de cada vez, e a tua cópia de trabalho contém só um conjunto de mudanças lógicas. OK, devolta à realidade. Costuma acontecer que tens de trabalhar em várias, e não relacionadas, tarefas ao mesmo tempo, e quando olhas para a caixa de diálogo de submissão, todas as alterações estão misturadas. A funcionalidade da *lista de alterações* ajuda a agrupar os ficheiros, tornando mais fácil ver o que estás a fazer. Está claro que isto só funciona se as alterações não se sobrepuserem. Se duas tarefas diferentes afectarem o mesmo ficheiro, não existe maneira de separar as alterações.



Importante

A funcionalidade de lista de alterações no TortoiseSVN, está só disponível no Windows XP e superior, visto depender de uma capacidade da consola que não está presente no Windows 2000. Desculpa, mas o Win2k é realmente velho, mas por favor não faças queixas.

Podes ver as listas de alterações em vários locais, mas os mais importantes são a caixa de diálogo **submeter** e a caixa de diálogo **verificar alterações**. Vamos começar pela caixa de diálogo **verificar**

alterações, depois de teres trabalhado em várias funcionalidades e muitos ficheiros. Quando primeiro abres a caixa de diálogo, todos os ficheiros alterados são listados em conjunto. Supõem que queres organizar as coisas e agrupa-las de acordo com a funcionalidade.

Selecciona um ou mais ficheiros e usa Context Menu → Move para a lista de alterações para adicionar um item à lista de alterações. Inicialmente não haverá lista de alterações, então a primeira vez que fizeres esta operação, irás criar uma nova lista de alterações. Dá-lhe um nome que descreva qual o objectivo do seu uso e clica OK. A caixa de diálogo ir-se-á alterar para mostrar grupos de itens.

Uma vez que tenhas criado uma lista de alterações, podes arrastar e largar itens para dentro dela, a partir de outra lista de alterações ou a partir do Explorador do Windows. Arrastando do Explorador pode ser útil, pois permite-te adicionar itens à lista de alterações antes do ficheiro ser modificado. Podes fazer isso a partir da caixa de diálogo verificar-alterações, mas apenas ao mostrar todos os ficheiros não modificados

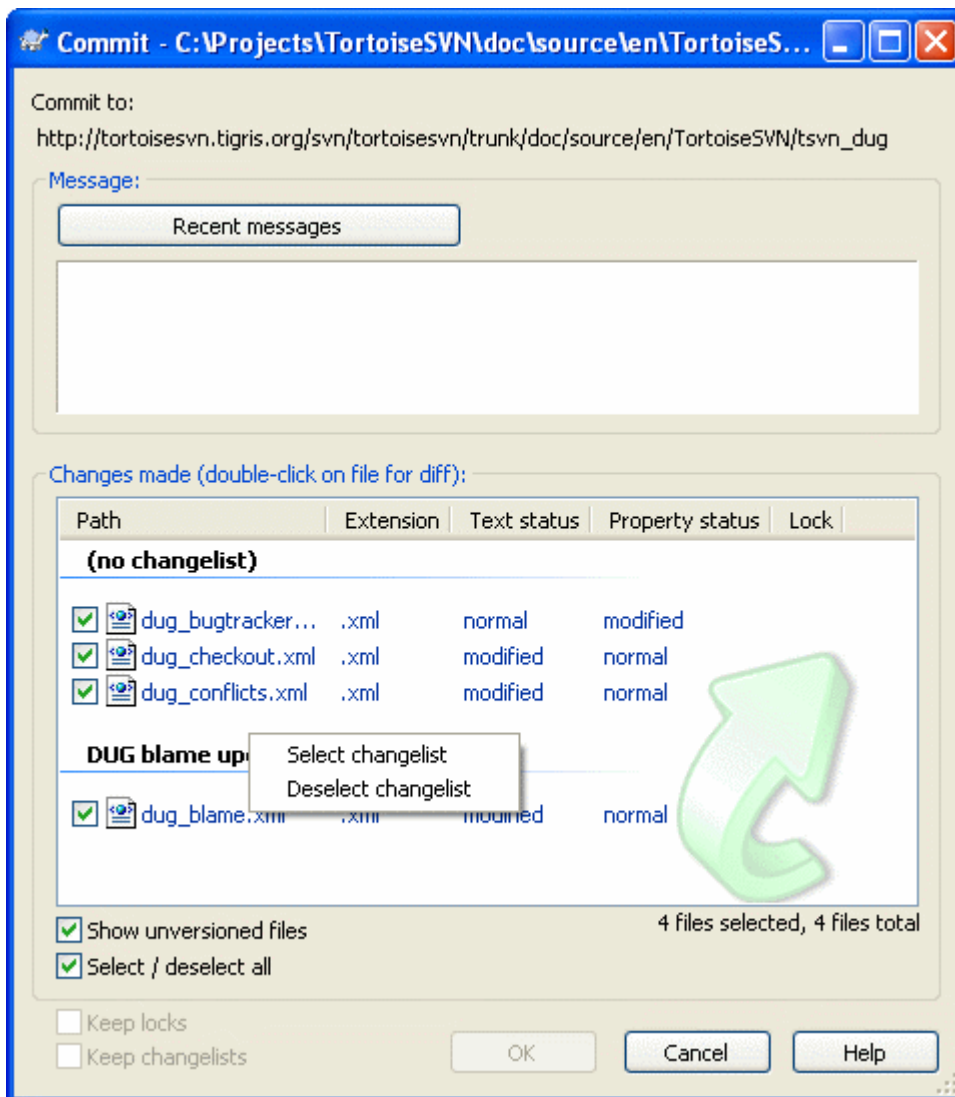


Figura 4.14. Caixa de diálogo Submeter com Listas de Alterações

Na caixa de diálogo submeter podes ver esses mesmos ficheiros agrupados por listas de alterações. À parte de dar uma indicação visual imediata dos grupos, podes também usar os cabeçalhos dos grupos para seleccionar quais os ficheiros a submeter.

No XP há um menu de contexto quando clicas com o botão direito num cabeçalho de grupo, que te dá a escolha para marcar ou desmarcar todas as entradas de grupo. No Vista, no entanto, o menu de contexto

não é necessário. Clica no cabeçalho de grupo para seleccionar todas as entradas, depois marca uma das entradas seleccionadas para os marcar a todos.

O TortoiseSVN reserva um nome de lista de alterações para o seu próprio uso, de nome `ignore-on-commit`. Esta é usada para sinalizar ficheiros versionados que quase nunca queres submeter, mesmo que tenham alterações. Esta funcionalidade é descrita em [Secção 4.4.3, “Excluir itens da lista a Submeter”](#).

Quando submetes ficheiros pertencentes a uma lista de alterações, debes esperar que não seja mais necessário estar associado a uma lista. Então por defeito, os ficheiros são removidos automaticamente ao submete-los. Se pretendes manter um ficheiro na sua lista de alterações, usa a caixa de verificação **Manter listas de alterações no fundo da caixa de diálogo submeter**.



Dica

Listas de alterações são puramente uma funcionalidade local de cliente. Criar e remover lista de alterações não afectará o repositório, nem a cópia de trabalho de outros. Elas são simplesmente uma forma conveniente para organizares os teus ficheiros.

4.9. Caixa de Diálogo Registo de Revisões

Para cada alteração que fazes e submetes, debes providenciar uma mensagem de registo para essa alteração. Dessa maneira, podes mais tarde descobrir as alterações que efectuaste e o porquê, e terás um registo detalhado para o teu processo de desenvolvimento.

A Caixa de Diálogo de Registo de Revisões recupera todas as mensagens de registo e mostra-tas. A visualização é dividida em 3 painéis.

- O painel de topo mostra a lista de revisões onde as alterações ao ficheiro/pasta foram submetidas. Este sumário inclui a data e tempo, a pessoa que submeteu a revisão e o início da mensagem de registo.

Linhas mostradas a azul indicam que algo foi copiado para esta linha de desenvolvimento (se calhar de um ramo).

- O painel do meio mostra a totalidade da mensagem de registo para a revisão seleccionada.
- O painel inferior mostra a lista de todos os ficheiros e pastas que foram alterados, como parte da revisão seleccionada.

Mas faz mais que isso - providencia comandos de menu de contexto, que podes usar para obter ainda mais informação sobre a história do projecto.

4.9.1. Invocando a Caixa de Diálogo Registo de Revisão

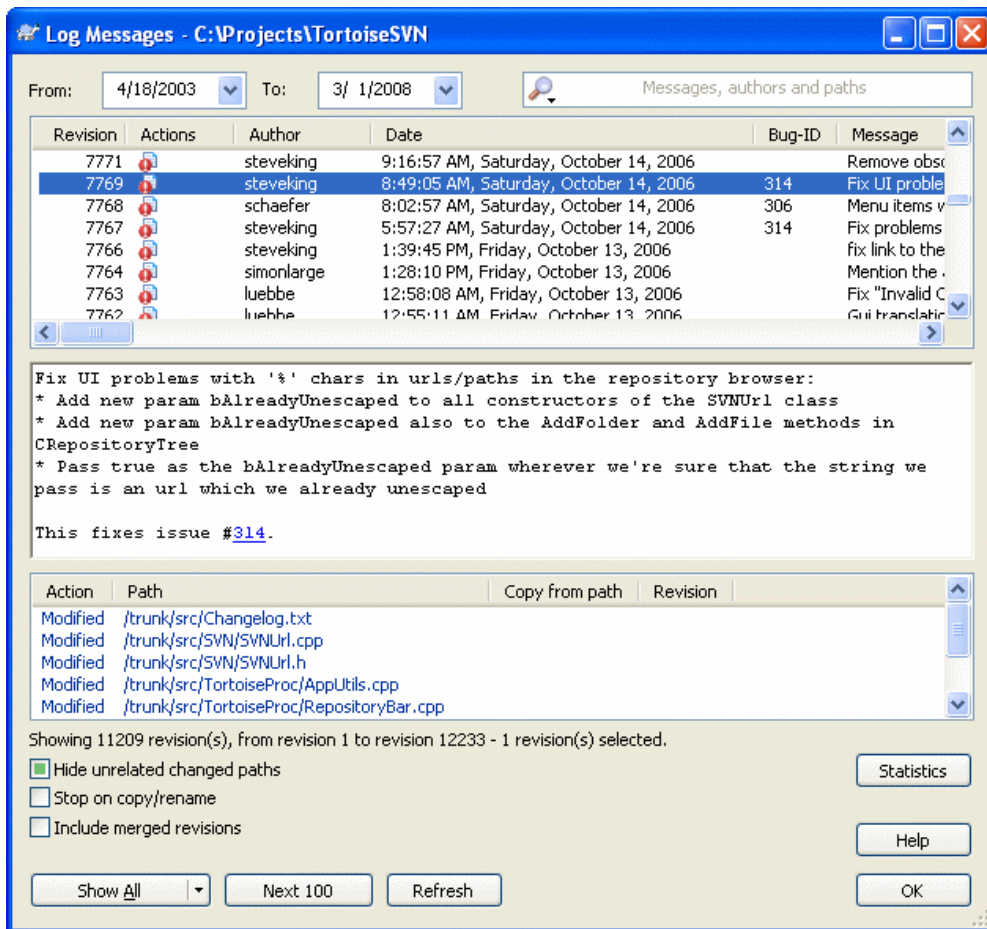


Figura 4.15. A Caixa de Diálogo Registo de Revisão

Existem vários locais a partir dos quais podes mostrar a caixa de diálogo de Registo:

- A partir do submenu de contexto TortoiseSVN
- A partir da página de propriedades
- A partir da caixa de diálogo de Progresso, após o fim de uma actualização. Então a caixa de diálogo de registo só mostrará as revisões que foram alteradas, desde o tua última actualização

Se o repositório não está disponível, verás a caixa de diálogo *Queres ficar fora de linha?* descrita em [Secção 4.9.10, “Modo Fora de Linha”](#).

4.9.2. Acções de Registo de Revisões

O painel de topo tem uma coluna *Acções* contendo ícones que sumarizam o que foi feito na revisão. Existem quatro diferentes ícones, cada mostrada na sua coluna.



Se uma revisão modificou um ficheiro ou pasta, o ícone *modificado* é mostrado na primeira coluna.



Se a revisão adicionou um ficheiro ou pasta, o ícone *adicionado* é mostrado na segunda coluna.



Se a revisão removeu um ficheiro ou pasta, o ícone *removido* é mostrado na terceira coluna.



Se a revisão substituiu um ficheiro ou pasta, o ícone *substituído* é mostrado na quarta coluna.

4.9.3. Obtendo Informação Adicional

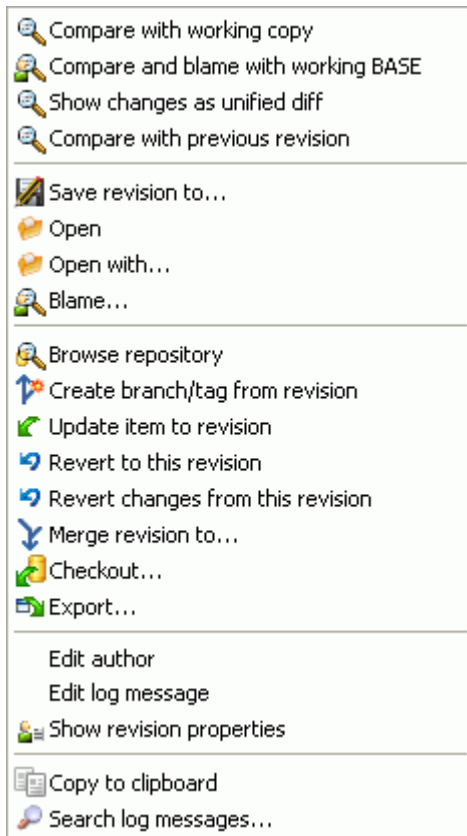


Figura 4.16. O Painel de Topo da Caixa de Diálogo Registo de Revisões com Menu de Contexto

O painel de topo da caixa de diálogo de registo tem um menu de contexto que te permite aceder a muito mais informação. Algumas dessas entradas do menu só aparecem quando o registo é mostrado para um ficheiro, e outras só quando o registo é mostrado para uma pasta.

Comparar com a cópia de trabalho

Compara a revisão seleccionada com a tua cópia de trabalho. A ferramenta, por defeito, de Comparação é o TortoiseMerge que é fornecido com o TortoiseSVN. Se a caixa de diálogo de registo for para uma pasta, então irá mostrar-te uma lista de ficheiros alterados e permitir-te-á rever as alterações feitas a cada ficheiro individualmente.

Compara e responsabiliza com a BASE de trabalho

Responsabiliza a revisão seleccionada e o ficheiro na tua BASE de trabalho, e compara os relatórios de responsabilidade usando a ferramenta de comparação visual. Ler [Secção 4.23.2, “Diferenças de responsabilidade”](#) para mais detalhes. (só ficheiros).

Mostrar diferenças como uma comparação unificada

Vê as alterações feitas na revisão seleccionada como uma comparação unificada (formato de correcção GNU). Isto só mostra as diferenças com algumas linhas de contexto. É mais difícil de ler que uma comparação visual de ficheiros, mas irá mostrar todas as alterações do ficheiro num formato compacto.

Compara com a revisão prévia

Compara a revisão seleccionada com a revisão prévia. Isto funciona de maneira similar à comparação com a cópia de trabalho. Para pastas, esta opção mostrará primeiro a caixa de diálogo de ficheiros alterados, permitindo-te seleccionar os ficheiros a comparar.

Compara e responsabiliza com a revisão anterior

Mostra a caixa de diálogo de ficheiros alterados, permitindo-te seleccionar os ficheiros. Responsabiliza a revisão seleccionada e a revisão prévia, e compara os resultados usando a ferramenta de comparação. (só pastas).

Guardar revisão em...

Guarda a revisão seleccionada para um ficheiro, para que tenhas uma revisão antiga de esse ficheiro (só ficheiros).

Abrir / Abrir com...

Abrir o ficheiro seleccionado com o programa de leitura por defeito, para esse tipo de ficheiro, ou um programa à tua escolha. (só ficheiros).

Responsabilizar...

Responsabiliza o ficheiro até à revisão seleccionada (só ficheiros).

Navegar repositório

Abre o navegador do repositório, para examinar o ficheiro ou pasta seleccionada no repositório, como era na revisão seleccionada.

Criar ramo/etiqueta a partir da revisão

Cria um ramo ou etiqueta a partir da revisão seleccionada. Isto é útil i.e. te esqueceres de criar uma etiqueta, e já submetes-te algumas alterações que não eram supostas ir naquela entrega.

Actualiza um item para uma revisão

Actualiza a tua cópia de trabalho para a revisão seleccionada. Útil se queres que a tua cópia de trabalho que reflecta um tempo passado, ou se ainda houve submissões posteriores para o repositório e tu queres actualizar a tua cópia de trabalho, um passo de cada vez. É melhor actualizar toda a pasta na tua cópia de trabalho, não só um ficheiro, ou de outro modo a tua cópia ficaria inconsistente.

Se queres desfazer permanentemente para uma revisão anterior, usa em substituição **Reverter para esta revisão**.

Reverter para esta revisão

Reverter para uma revisão anterior. Se efectuaste várias alterações e então decides que queres realmente voltar para trás, para o estado em que estavam as coisas na revisão N, este é o comando que precisas. As alterações são desfeitas na tua cópia de trabalho, e este comando *não* afectará o repositório até submeteres as alterações. Tomar atenção que isto irá desfazer *todas* as alterações efectuadas depois da revisão seleccionada, substituindo os ficheiros/pastas com uma revisão anterior.

Se a tua cópia de trabalho está num estado não alterado, depois de efectuares esta operação, a tua cópia de trabalho ir-se-á mostrar como alterada. Se já tens alterações locais, este comando irá integrar as alterações *desfeitas* na tua cópia de trabalho.

O que está acontecer internamente é que, o Subversion efectua uma integração inversa de todas as alterações feitas na revisão seleccionada, desfazendo o efeito de submissões prévias.

Se depois de efectuares esta acção decides que queres *desfazer o desfeito*, e obteres a tua cópia de trabalho de volta para o seu estado não alterado, deves usar TortoiseSVN → Reverter, a partir do Explorador do Windows, que irá descartar as modificações locais efectuadas por esta acção de integração inversa.

Se queres simplesmente ver como um ficheiro ou pasta eram numa revisão anterior, usa em alternativa Actualizar para revisão ou Salvar revisão como....

Reverter alterações desta revisão

Desfazer alterações que foram feitas na revisão seleccionada. As alterações são desfeitas na tua cópia de trabalho, isto é, esta operação *não* afecta de modo algum o repositório! De notar que isto irá desfazer só as alterações feitas nesta revisão; não substitui a tua cópia de trabalho com o ficheiro completo como estava nessa revisão. Isto é muito útil para desfazer uma alteração anterior, quando uma outra alteração não relacionada foi feita depois.

Se a tua cópia de trabalho está num estado não alterado, depois de efectuares esta operação, a tua cópia de trabalho ir-se-á mostrar como alterada. Se já tens alterações locais, este comando irá integrar as alterações *desfeitas* na tua cópia de trabalho.

O que está a acontecer internamente é que, o Subversion executa uma integração inversa dessa única revisão, desfazendo o seu efeito numa submissão anterior.

Podes *desfazer o desfeito* como descrito acima em Reverter para esta revisão.

Integrar revisão para...

Integrar a revisão/revisões seleccionada(s) numa cópia de trabalho diferente. Uma caixa de diálogo de selecção de pasta permite-te escolher a cópia de trabalho para onde integrar, mas depois disso não há caixa de diálogo de confirmação, nem a oportunidade para experimentar uma integração de teste. É uma boa ideia integrar para uma cópia de trabalho não modificada, para que possas reverter as alterações caso não resulte! Isto é uma funcionalidade útil se queres integrar revisões seleccionadas de um ramo para outro.

Checkout...

Fazer um SVN exportar de fresco, da pasta seleccionada na revisão seleccionada. Irá mostrar uma caixa de diálogo de modo a confirmares o URL e a revisão, e seleccionar uma localização para o SVN exportar.

Exportar...

Exportar o ficheiro/pasta seleccionada na revisão seleccionada. Irá aparecer uma caixa de diálogo para confirmares o URL e a revisão, e seleccionares a localização para exportar.

Editar mensagem de autor/registo

Edita a mensagem de registo ou o autor anexado à mensagem de submissão prévia. Consulta [Secção 4.9.7, “Alterando a Mensagem de Registo e Autor”](#) para saberes como isto funciona

Mostrar propriedades de revisão

Vê e edita qualquer propriedade de revisão, não apenas a mensagem de registo e o autor. Consulte [Secção 4.9.7, “Alterando a Mensagem de Registo e Autor”](#).

Copiar para o clipboard

Copia os detalhes do registo das revisões seleccionadas para o clipboard. Isto irá copiar o número de revisão, autor, data, mensagem de registo e a lista de itens alterados para cada revisão.

Pesquisar mensagens de registo...

Pesquisar mensagens de registo com o texto que inseriste. É feita a pesquisa das mensagens de registo que inseriste e também o sumário de acções criada pelo Subversion (mostrado no painel inferior). A pesquisa não é sensível à altura da letra.

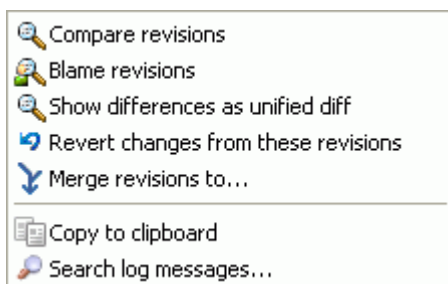


Figura 4.17. Menu de Contexto do Painel Superior para 2 revisões seleccionadas

Se seleccionares duas revisões de uma vez (usando o modificador usual **Ctrl**), o menu de contexto muda e disponibiliza menos opções:

Comparar revisões

Comparar duas revisões seleccionadas usando uma ferramenta de comparação visual. A Ferramenta de Comparação por defeito, é o TortoiseMerge que é fornecida com o TortoiseSVN.

Se seleccionares esta opção para uma pasta, uma caixa de diálogo posterior aparecerá listando os ficheiros alterados e, oferecendo-te ainda mais opções de comparação. Lê mais sobre a caixa de diálogo Comparar Revisões em [Secção 4.10.3, “Comparando Pastas”](#).

Responsabilizar revisões

Responsabiliza as duas revisões e compara os relatórios de responsabilidade usando a ferramenta de comparação visual. Para mais detalhes lê [Secção 4.23.2, “Diferenças de responsabilidade”](#).

Mostrar diferenças como comparação unificada

Vê as diferenças entre as duas revisões seleccionadas como um ficheiro de comparação unificada. Isto funciona para ficheiros e pastas.

Copiar para o clipboard

Copiar as mensagens de registo para o clipboard como descrito acima.

Pesquisar mensagens de registo...

Pesquisar mensagens de registo como descrito acima.

Se seleccionares duas ou mais revisões (usando os modificadores usuais **Ctrl** ou **Shift**), o menu de contexto irá incluir uma entrada para reverter todas as alterações que foram feitas nas revisões seleccionadas. Esta é a maneira mais fácil para rebobinar um grupo de revisões de uma vez só.

Podes também escolher integrar as revisões seleccionadas noutra cópia de trabalho, como descrito acima.

Se todas as revisões seleccionadas têm o mesmo autor, podes editar o autor de todas essas revisões de uma vez só.

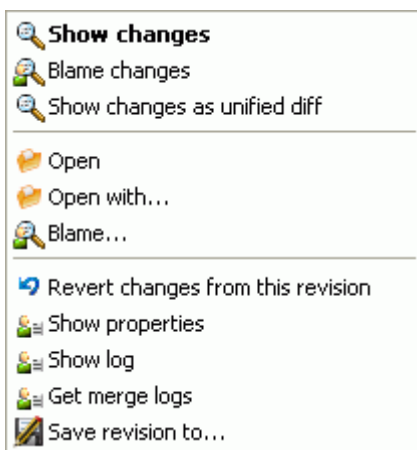


Figura 4.18. O Painel de fundo da Caixa de Diálogo Registo com o Menu de Contexto

O painel inferior da caixa de diálogo Registo também tem um menu de contexto, que te permite

Mostrar alterações

Mostra alterações efectuadas na revisão seleccionada, para o ficheiro seleccionado. Este menu de contexto apenas está disponível para ficheiros visualizados como *modificado*.

Responsabilizar alterações

Responsabilizar a revisão seleccionada e a revisão prévia para o ficheiro seleccionado, e compara os relatórios de responsabilidade usando uma ferramenta de comparação visual. Para mais detalhes ler [Secção 4.23.2, “Diferenças de responsabilidade”](#).

Mostrar como comparação unificada

Mostrar as alterações do ficheiro no formato comparação unificada. Este menu de contexto está só disponível para ficheiros visíveis como *modificado*.

Abrir / Abrir com...

Abrir o ficheiro seleccionado com o leitor por defeito para esse tipo de ficheiro ou, com um programa à tua escolha.

Responsabilizar...

Abre a caixa de diálogo Responsabilizar, permitindo-te responsabilizar até à revisão seleccionada.

Reverter alterações desta revisão

Reverter as alterações feitas no ficheiro seleccionado nesta revisão.

Mostrar propriedades

Ver as propriedades Subversion para o item seleccionado.

Mostrar registo

Mostra o registo de revisão para o único ficheiro seleccionado.

Obter registos de integração

Mostrar o registo de revisão para o único ficheiro seleccionado, incluindo as alterações integradas. Descobre mais em [Secção 4.9.6, “Funcionalidades de Rastreamento de Integração”](#).

Guardar revisão em...

Guardar a revisão seleccionada num ficheiro para que tenhas uma versão antiga do mesmo.



Dica

Deves ter notado que por vezes referimo-nos às alterações e outras vezes a diferenças. Qual a diferença?

O Subversion usa números de revisão para significar 2 coisas diferentes. A revisão, de modo geral, representa o estado do repositório num dado ponto no tempo, mas pode também ser usado para representar o conjunto de alterações que criou a revisão, e.g. “Feito em r1234” significa que as alterações submetidas na r1234 implementa a funcionalidade X. Para tornar mais claro qual o sentido que está a ser utilizado, usamos dois termos diferentes.

Se seleccionares duas revisões N e M, o menu de contexto irá oferecer a visualização da *diferença* entre essas duas revisões. Em termos Subversion isto é `diff -r M:N`.

Se seleccionares uma só revisão N, o menu de contexto irá oferecer a visualização das *alterações* feitas nessa revisão. Em termos Subversion isto é `diff -r N-1:N` ou `diff -c N`.

O painel inferior mostra os ficheiros alterados em todas as revisões seleccionadas, então o menu de contexto irá oferecer sempre a visualização de *alterações*.

4.9.4. Obtendo mais mensagens de registo

A caixa de diálogo de Registo não mostra sempre todas as alterações jamais feitas por um número de razões:

- Para um grande repositório poderão existir centenas ou mesmo milhares de alterações e, ir busca-las a todas pode demorar muito tempo. Normalmente só estarás interessado nas alterações mais recentes.

Por defeito, o número de mensagens de registo trazidas está limitada às 100, mas no entanto podes alterar esse valor no TortoiseSVN → Preferências (Secção 4.30.1.2, “Preferências das Caixas de Diálogo TortoiseSVN 1”),

- Quando a caixa Parar ao copiar/alterar o nome está verificada, o Mostrar Registo irá parar no ponto onde o ficheiro ou pasta seleccionada foi copiado de outro ponto do repositório. Isto pode ser útil ao olhar para ramos (ou etiquetas), já que pára na raiz desse ramo, e dá uma rápida indicação das alterações feitas só nesse ramo.

Normalmente tu quererás deixar esta opção não verificada. O TortoiseSVN lembra-se do estado da caixa de verificação, respeitando desse modo a tua preferência.

Quando a caixa de diálogo Mostrar Registo é invocada de dentro da caixa de diálogo de Integrar, esta caixa é sempre verificada por defeito. Isto porque, integrar é na maioria das vezes olhar para as alterações em ramos e voltar para trás, e ir além da raiz do ramo não fará sentido nessa situação.

De notar que o Subversion actualmente implementa o renomear com o par copiar/apagar, então renomear um ficheiro ou pasta irá provocar, que a visualização do registo pare se esta opção está verificada.

Se queres ver mais mensagens de registo, clica Próximos 100 para trazer as próximas 100 mensagens de registo. Podes repetir este procedimento quantas vezes necessitares.

Próximo a este botão está um botão multifunções que lembra a última opção para que o usaste. Clica na seta, para ver as outras opções disponíveis.

Usa Mostrar Intervalo... se queres ver um intervalo de revisões específica. A caixa de diálogo irá então pedir-te para introduzir a revisão de início e fim.

Usa Mostrar Tudo se queres ver *todas* as mensagens de registo a partir da revisão HEAD de volta até à revisão 1.

4.9.5. Revisão Actual da Cópia de Trabalho

Porque a caixa de diálogo de registo te mostra o registo a partir da HEAD, e não da revisão corrente da cópia de trabalho, acontece frequentemente que são mostradas mensagens de registo para conteúdo que ainda não foi actualizado, na tua cópia de trabalho. Para ajudar a tornar isto mais claro, a mensagem de submissão que corresponde à revisão que tens na tua cópia de trabalho, é mostrada a negrito.

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 Secção 4.30.10, “Configurações do Registo”.

4.9.6. Funcionalidades de Rastreamento de Integração

O Subversion 1.5 e superior mantém o registo das integrações recorrendo ao uso de propriedades. Isto permite-te obter um histórico mais detalhado das alterações integradas. Por exemplo, se desenvolveres uma nova funcionalidade num ramo e a seguir integras esse ramo de volta para o trunk, a funcionalidade desenvolvida irá ser mostrada no registo do trunk, como uma única submissão para a integração, mesmo que tenha havido 1000 submissões durante o desenvolvimento do ramo.

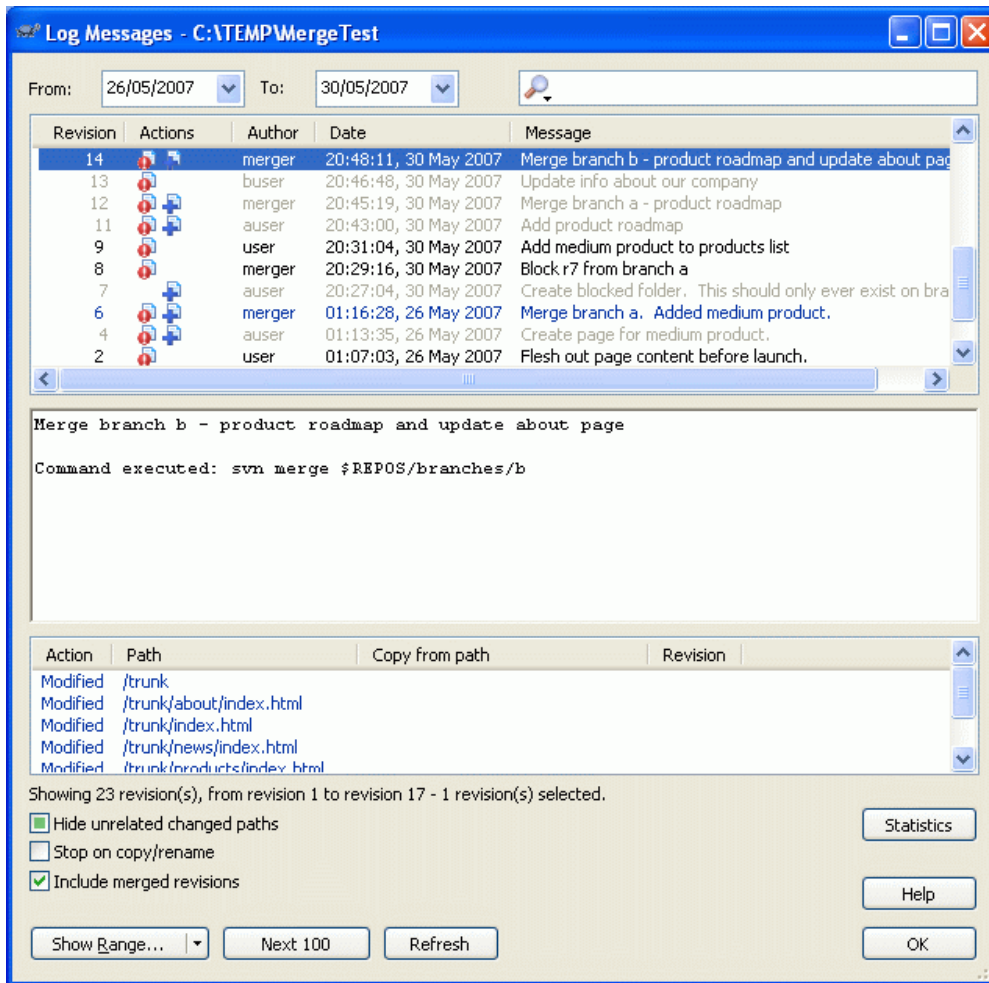


Figura 4.19. A Caixa de Diálogo Registo Mostra Rasto das Revisões de Integração

Se quiseres ver o detalhe de quais revisões que foram integradas como parte dessa submissão, usa a caixa de verificação **Incluir revisões integradas**. Isto irá outra vez buscar as mensagens de registo, mas também irá colocar entrefolhas as mensagens de registo das revisões que foram submetidas. Revisões integradas são mostradas a cinzento, porque representam alterações feitas numa parte diferente da árvore.

É claro que integrar nunca é simples! Durante o desenvolvimento da funcionalidade no ramo, irão provavelmente ocorrer integrações ocasionais de volta para o trunk, de modo a manter o ramo em sincronismo com a linha de código principal.

4.9.7. Alterando a Mensagem de Registo e Autor

As propriedades de revisão são completamente diferentes das propriedades do Subversion de cada item. Revprops são itens descritivos que são associados a um número de revisão específico no repositório, como as mensagens de registo, data de submissão e autor da submissão (autor).

Por vezes poderás querer modificar a mensagem de registo que anteriormente introduziste, talvez porque contém o erro de ortografia, ou queres melhorar a mensagem, ou muda-la por outras razões. Ou queres alterar o autor da submissão porque de esqueces-te de activar a autenticação, ou...

O Subversion deixa-te alterar as propriedades da revisão em qualquer altura que desejes. Mas como essas alterações não podem ser desfeitas (essas alterações não são versionadas) essa funcionalidade está desactivada por defeito. Para activar esta funcionalidade deverás configurar o gancho pre-revprop-change. Consultar o respectivo capítulo no *Hook Scripts* [<http://svnbook.red-bean.com/en/1.5/>]

svn.reposadmin.create.html#svn.reposadmin.create.hooks] no livro do Subversion para detalhes em como o fazer. Ler [Secção 3.3, “Scripts de gancho de servidor”](#) para encontrares mais notas sobre como implementar ganchos numa máquina Windows.

Uma vez que tenhas configurado o teu servidor com os ganchos pretendidos, podes mudar o autor e a mensagem de registo (ou qualquer outra revprop) de qualquer revisão, usando o menu de contexto do painel de topo da caixa de diálogo Registo. Podes também editar a mensagem de registo usando o menu de contexto no painel do meio.



Atenção

Porque as propriedades de revisão do Subversion não são versionadas, fazer alterações a tais propriedades (por exemplo, a propriedade da mensagem de submissão `svn:log`) irá re-escrever o seu valor *para sempre*.

4.9.8. Filtrando Mensagens de Registo

Se queres restringir as mensagens de registo, para mostrar só as que estás interessado em vez de percorreres uma lista de centenas, podes usar os controlos de filtros no topo da caixa de diálogo Registo. Os controlos de início e fim de data permite-te restringir a visualização a um intervalo de datas. A caixa de pesquisa permite-te mostrar só mensagens que contêm uma frase particular.

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`.

De salientar que esses filtros actuam nas mensagens já adquiridas. Eles não controlam o descarregamento de mensagens do repositório.

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

Por vezes as tuas práticas de trabalho iram requerer que as tuas mensagens de registo sigam um formato particular, o que significa que o texto que descreve as alterações não é visível no sumário abreviado mostrado no painel de topo. A propriedade `tsvn:logsummary` pode ser utilizada para extrair uma parte da mensagem de registo para ser mostrada no painel de topo. Ler [Secção 4.17.2, “Propriedades de Projecto TortoiseSVN”](#) para descobrir como usar esta propriedade.



Registos Não Formatados do Navegador de Repositório

Porque a formatação depende do acesso ás propriedades do subversion, tu só verás os seus resultados quando obtiveres uma cópia de trabalho SVN exportada. Buscar remotamente propriedades é uma operação lenta, pelo que não verás esta funcionalidade em acção a partir do navegador de repositório.

4.9.9. Informação Estatística

O botão Estatísticas abre uma caixa que, mostra alguma informação interessante sobre as revisões mostradas na caixa de diálogo Registo. Essa mostra quantos autores estiveram a trabalhar, quantas submissões eles fizeram, o progresso por semana e muito mais. Agora podes ver por alto que trabalhou mais arduamente e quem esteve na balda ;-)

4.9.9.1. Página de Estatísticas

Esta página dá-te todos os números que tenhas em mente em particular, o período e o número de revisões cobertas e alguns valores min/max/média.

4.9.9.2. Página de Submissões por Autor

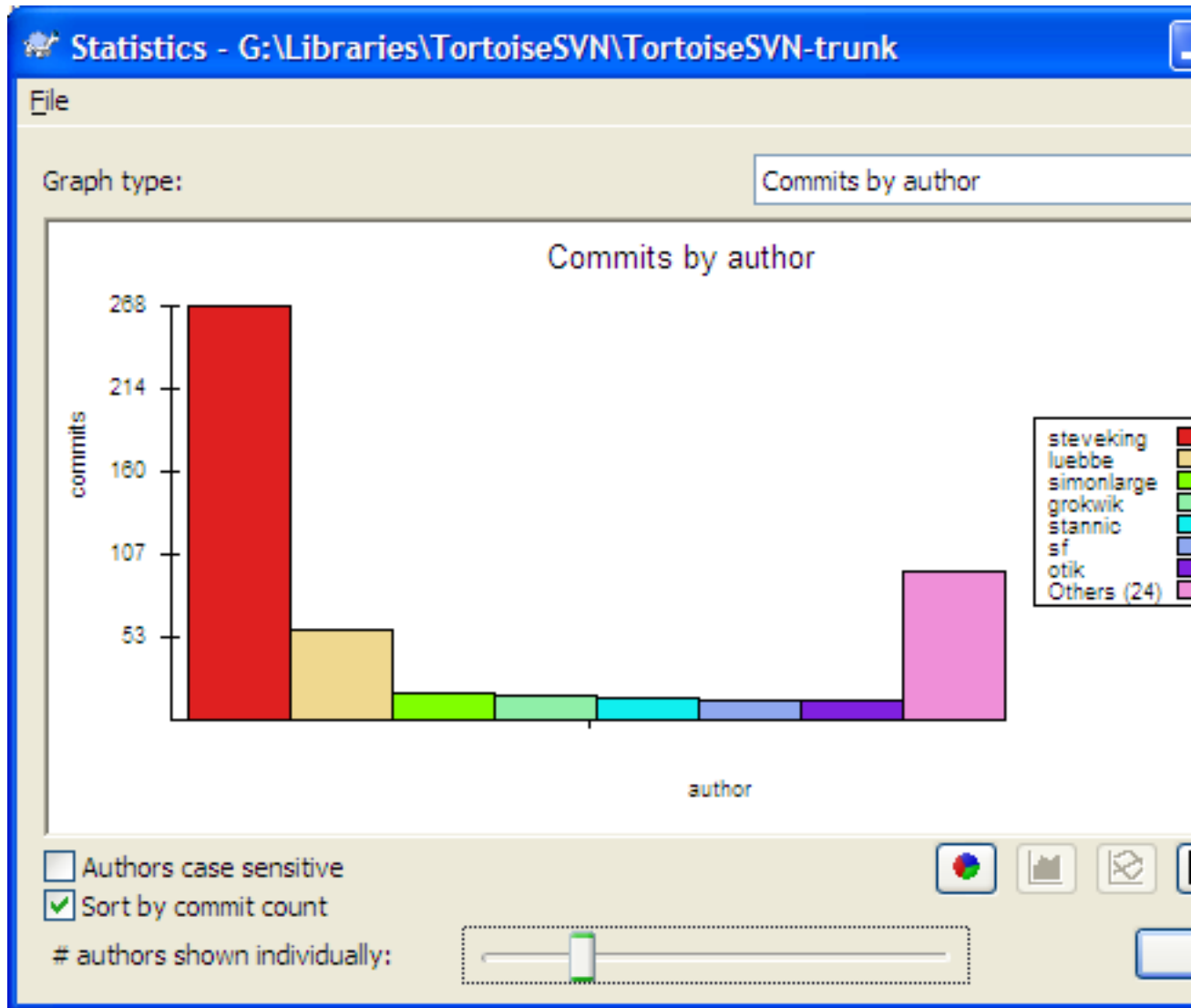


Figura 4.20. Histograma de Submissões-por-Autor

Este gráfico mostra quais os autores que têm estado activos no projecto como um simples histograma, histograma empilhado ou gráfico de queijo.

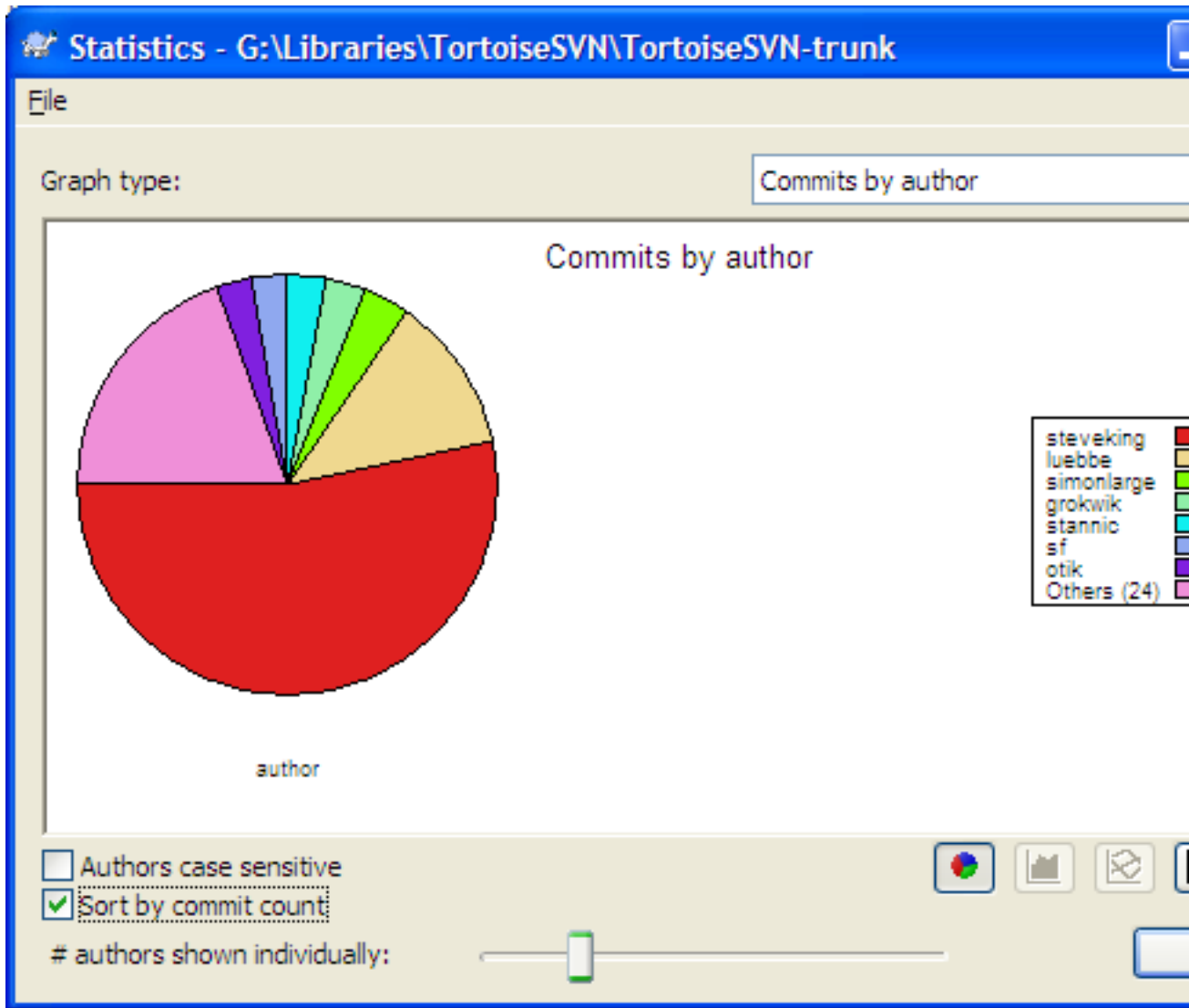


Figura 4.21. Gráfico de queijo Submissões-por-Autor

onde há alguns grandes autores e muitos pequenos contribuidores, o número de pequenos segmentos pode tornar o gráfico mais difícil de ler. O controlo deslizante no fundo permite-te colocar um limite (como percentagem do total de submissões) abaixo do qual, qualquer actividade é agrupado na categoria *Outros*.

4.9.9.3. Página de Submissões por data

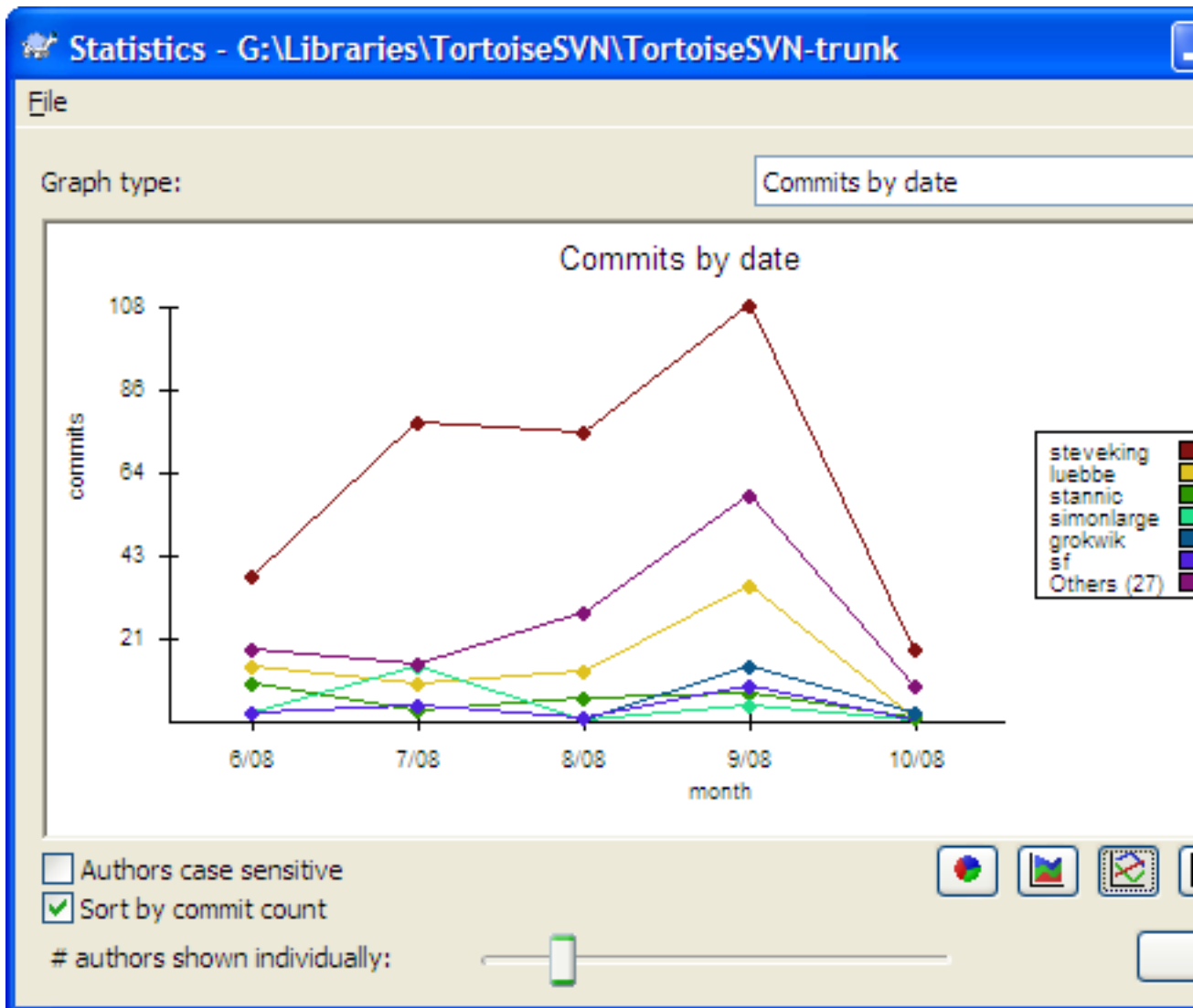


Figura 4.22. Gráfico de Submissões-por-data

Esta página dá-te uma representação gráfica da actividade do projecto em termos de, número de submissões *e* autor. Isto dá alguma ideia de quando um projecto está a ser trabalhado, quem esteve a trabalhar a que altura.

Quando existem vários autores irás obter muitas linhas no gráfico. Existem duas vistas disponíveis aqui: *normal*, onde a actividade de cada autor é relativa à linha base e, *empilhada*, onde a actividade de cada autor é relativa à linha de baixo. A última opção evita o cruzamento de linhas que pode tornar o gráfico mais fácil de ler, mas menos fácil para ver a produção de um determinado autor.

Por defeito, a análise é sensível a maiúsculas e minúsculas, pelo que os utilizadores PeterEgan e PeteRegan são tratados como autores diferentes. No entanto, em muitos casos, os nomes de utilizadores não são sensíveis a maiúsculas e minúsculas, e são por vezes introduzidos inconsistentemente, pelo que poderás querer que o DavidMorgan e o davidmorgan sejam tratados como a mesma pessoa. Usar a caixa de verificação Autores sensíveis a maiúsculas e minúsculas para controlar como este caso é tratado.

Ter em atenção que as estatísticas cobrem o mesmo período que a caixa de diálogo Registo. Se esta só mostra uma revisão então as estatísticas não te dirão muito.

4.9.10. Modo Fora de Linha

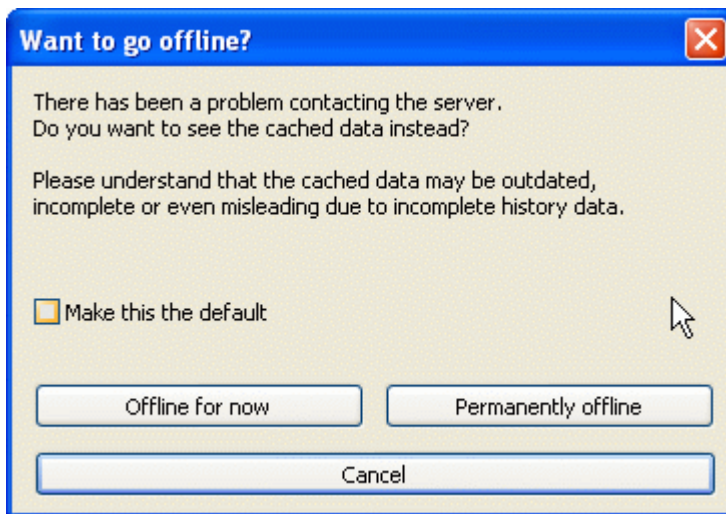


Figura 4.23. Caixa de diálogo Colocar-se em Fora de Linha

Se o servidor não está disponível e tens a cache de registo activada, podes usar a caixa de diálogo e o navegador de revisões no modo fora de linha. Usando dados da cache que te permite continuar a trabalhar, apesar de a informação não estar actualizada ou mesmo completa.

Aqui tens três opções:

Fora-de-linha por agora

Completa a operação corrente no modo fora-de-linha mas, repetir a tentativa de acesso ao repositório da próxima vez que forem requeridos dados do registo.

Fora-de-linha permanentemente

Permanecer em modo fora-de-linha até um pedido de verificação ao repositório for especificamente requerido. Consultar [Secção 4.9.11, “Refrescar a Vista”](#).

Cancelar

Se não queres continuar com a operação com dados possivelmente obsoletos, basta cancelar.

A caixa de verificação Tornar isto por defeito previne esta caixa de diálogo de reaparecer, e escolhe sempre a opção que escolheres a seguir. Podes ainda mudar (ou remover) a opção por defeito, após a teres efectuado, a partir de TortoiseSVN → Preferências.

4.9.11. Refrescar a Vista

Se queres verificar o servidor novamente por novas mensagens de registo, podes simplesmente refrescar a vista usando **F5**. Se tiveres a usar a cache de registo (disponível por defeito), esta irá verificar o repositório por novas mensagens e trazer só as novas. Se a cache de registo estava em modo fora-de-linha esta irá também tentar voltar a estar em linha.

Se está a utilizar a cache de registo e pensas que o conteúdo da mensagem ou o autor tenha mudado, podes usar **Shift-F5** ou **Ctrl-F5** para rebuscar as mensagens mostradas a partir do servidor e actualizar a cache de registo. De notar que isto só afecta as mensagens mostradas correntemente, e não invalida a totalidade da cache para esse repositório.

4.10. Ver Diferenças

Um dos requisitos mais comuns no desenvolvimento de projectos é poder ver o que foi alterado. Tu podes querer ver as diferenças entre duas revisões do mesmo ficheiro, ou as diferenças entre dois ficheiros diferentes. O TortoiseSVN fornece uma ferramenta incorporada para ver as diferenças de ficheiros de texto, chamada TortoiseMerge. Para ver as diferenças de ficheiros de imagem o TortoiseSVN também tem uma ferramenta, chamada TortoiseIDiff. É claro que se quiseres, podes usar o teu programa de comparação favorito.

4.10.1. Diferenças em ficheiros

Alterações locais

Se pretendes ver que alterações *tu* efectuaste na tua cópia de trabalho, usa apenas o menu de contexto do explorador e selecciona TortoiseSVN → Comparar.

Diferenças para outro ramo/etiqueta

Se pretendes ver o que foi alterado no trunk (se estás a trabalhar num ramo) ou num ramo específico (se estás a trabalhar no trunk), podes usar o menu de contexto do explorador. Apenas mantém a tecla **Shift** enquanto clicas com o botão direito no ficheiro. Então selecciona TortoiseSVN → Comparar com URL. Na caixa de diálogo seguinte, especifica o URL no repositório com o qual queres comparar o teu ficheiro.

Também podes usar o navegador de repositório e, seleccionar duas árvores para comparar, talvez duas etiquetas, ou um ramo/etiqueta com o trunk. O menu de contexto aí permite-te compará-las usando Comparar revisões. Ler mais em [Secção 4.10.3, “Comparando Pastas”](#).

Diferenças de uma revisão anterior

Se pretendes ver a diferença entre uma revisão particular e a tua cópia de trabalho, usa a caixa de diálogo Registo de Revisões e selecciona a revisão de interesse, então selecciona a partir do menu de contexto Compara com cópia de trabalho.

Se pretendes ver as diferenças entre a última revisão submetida e a tua cópia de trabalho, assumindo que a tua cópia de trabalho não foi modificada, clica apenas com o botão direito no ficheiro. E selecciona TortoiseSVN → Comparar com revisão anterior. Então será executada uma comparação entra revisão antes da última submissão (como registada na tua cópia de trabalho) e a BASE de trabalho. Isto mostra a última alteração feita nesse ficheiro para o levar ao estado em que se encontra correntemente na tua cópia de trabalho. Não irá mostrar alterações mais recentes que a tua cópia de trabalho.

Diferenças entre duas revisões prévias

Se pretendes ver as diferenças entre duas revisões prévias, que já estão submetidas, usa a caixa de diálogo Registo de Revisões e selecciona as duas revisões que queres comprar (usando o modificador usual **Ctrl**). Então selecciona a partir do menu de contexto Comparar revisões.

Se fizeste isto a partir do registo de revisão de uma pasta, a caixa de diálogo Arquivos Alterados aparecerá mostrando a lista de ficheiros alterados nessa pasta. Ler mais em [Secção 4.10.3, “Comparando Pastas”](#).

Todas as alterações feitas numa submissão

Se pretendes ver as alterações feitas em todos os ficheiros numa revisão particular numa vista única, podes usar o resultado da Comparação-Unificada (formato de remendo GNU). Isto mostra apenas as diferenças com algumas linhas de contexto. É mais difícil de ler que uma comparação visual de ficheiros, mas irá mostrar todas as alterações em conjunto. A partir da caixa de diálogo Registo de Revisões selecciona as revisões de interesse, e selecciona a partir do menu de contexto Mostrar alterações como uma comparação unificada.

Diferenças entre ficheiros

Se pretendes ver as diferenças entre dois ficheiros diferentes, podes fazê-lo directamente no explorador seleccionando ambos os ficheiros (usando o modificador usual **Ctrl**). Então a partir do menu de contexto do explorador, selecciona TortoiseSVN → Comparar.

Diferenças entre ficheiro/pasta da CT e um URL

Se pretendes ver as diferenças entre um ficheiro na tua cópia de trabalho e um ficheiro em qualquer repositório do Subversion, podes fazê-lo directamente no explorador, seleccionando o ficheiro e mantendo premido a tecla **Shift** enquanto clicas com o botão direito para aceder ao menu de contexto.

Selecciona TortoiseSVN → Comparação com URL. Podes fazer a mesma coisa para uma pasta da cópia de trabalho. O TortoiseMerge mostra essas diferenças da mesma maneira que mostra um ficheiro de remendo - uma lista de ficheiros alterados que podes visualizar, um de cada vez.

Diferenças com informação de responsabilidade

Se pretendes ver não só as diferenças, mas também ver o autor, revisão e data dessa alteração, podes combinar os relatórios de comparação e responsabilização a partir da caixa de diálogo registo de revisões. Para mais detalhes consulta [Secção 4.23.2, “Diferenças de responsabilidade”](#).

Diferenças entre pastas

As ferramentas embutidas fornecidas com o TortoiseSVN, não suportam a visualização de diferenças entre hierarquia de pastas. Mas se tiveres uma ferramenta externa que suporte essa funcionalidade, podes usá-la em substituição. Em [Secção 4.10.5, “Ferramentas de Comparação/Integração”](#) nós falamos sobre algumas que temos usado.

Se tens configurada uma ferramenta de comparação de terceiros, podes usar **Shift** quando seleccionares o comando de Comparação, para usar a ferramenta alternativa. Consulta [Secção 4.30.5, “Preferências de Programas Externos”](#) para descobrires como configurar outras ferramentas de comparação.

4.10.2. Opções de Fim-de-Linha e Espaços-Brancos

Por vezes no tempo de vida do projecto poderás alterar os fim-de-linha de CRLF para LF, ou poderás alterar a indentação de uma secção. Infelizmente isto irá marcar um grande número de linhas como alteradas, mesmo que não haja alterações de significado no código. Estas opções te irão ajudar a gerir essas alterações quando for altura de comparar e aplicar as diferenças. Irás ver essas definições nas caixas de diálogo Integrar e Responsabilizar, tal como nas definições para o TortoiseMerge.

Ignorar fins-de-linha exclui alterações que são exclusivamente devidas a diferenças no estilo de fim-de-linha.

Comparar espaços em branco inclui todas as alterações na indentação e espaços-em-branco em linha, como linhas adicionadas/removidas.

Ignorar alterações em espaços em branco exclui alterações que são exclusivamente devidas a uma alteração na quantidade e tipo do espaço em branco, e.g. alterando a indentação ou alterando tabulações para espaços. Adicionando espaços em branco onde nenhum existia, ou remover completamente um espaço em branco, será sempre mostrado como uma alteração.

Ignorar todos os espaços em branco exclui todas as alterações relacionadas só com espaços em branco.

Naturalmente qualquer linha com conteúdo alterado será sempre incluída na comparação.

4.10.3. Comparando Pastas

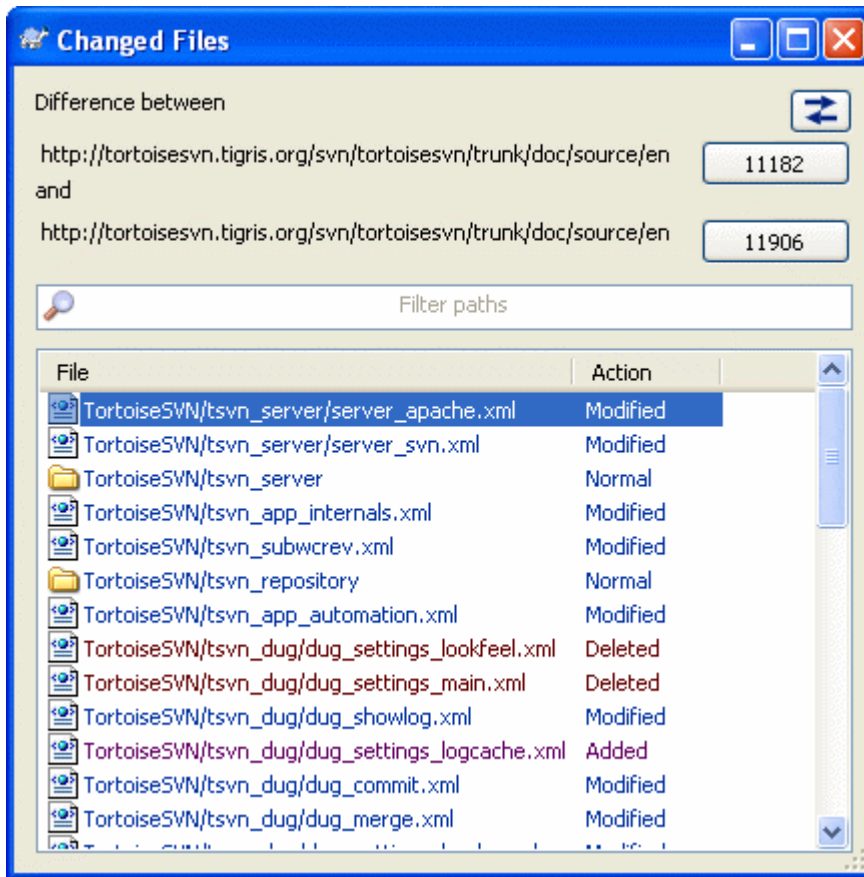


Figura 4.24. A Caixa de Diálogo Comparar Revisões

Quando seleccionas duas árvores no interior do navegador de repositório ou quando seleccionas duas revisões de uma pasta na caixa de diálogo, podes Context menu → Comparar revisões.

Esta caixa de diálogo mostra a lista de todos os ficheiros que foram alterados, permitindo-te compará-los ou responsabiliza-los individualmente, usando o menu de contexto.

Podes exportar uma *árvore de alterações*, que é útil se necessitas de enviar a alguém a estrutura da árvore do teu projecto, mas contendo apenas aqueles ficheiros que foram alterados. Esta operação funciona só em ficheiros alterados, pelo que só precisas de seleccionar os ficheiros de interesse - o que normalmente significa todos - e depois Menu de contexto → Exportar selecção para.... Serás solicitado a introduzir uma localização para guardar a árvore de alterações.

Podes também exportar a *lista* de ficheiros alterados para um ficheiro de texto, usando Menu de contexto → Guardar lista de ficheiros seleccionados para....

Se queres exportar a lista de ficheiros e também as acções (modificada, adicionada,removida), poderás fazê-lo usando Menu de contexto → Copiar selecção para a área de transferência.

O botão no topo permite-te alterar a direcção da comparação. Podes mostrar as alterações necessárias para ir de A a B, ou se preferires, de B a A.

Os botões com os números de revisão podem ser usados para mudar para um intervalo de revisões diferentes. Quando alteras o intervalo, a lista de itens, que diferem entre revisões, será actualizada automaticamente.

Se a lista de nomes de ficheiros é muito extensa podes utilizar a caixa de pesquisa, para assim reduzir a lista para ficheiros cujo nome contém um texto específico. Ter em atenção que é usada uma pesquisa de texto simples, e se queres restringir a lista a ficheiro de fonte de C, deverás introduzir .c em vez de * . c.

4.10.4. Comparando Imagens usando o TortoiseDiff

Existem muitas ferramentas disponíveis para comparar ficheiros de texto, incluindo o nosso TortoiseMerge, mas com frequência estamos também à espera de ver como um ficheiro de imagem foi alterado. É por isso que criámos o TortoiseIDiff.

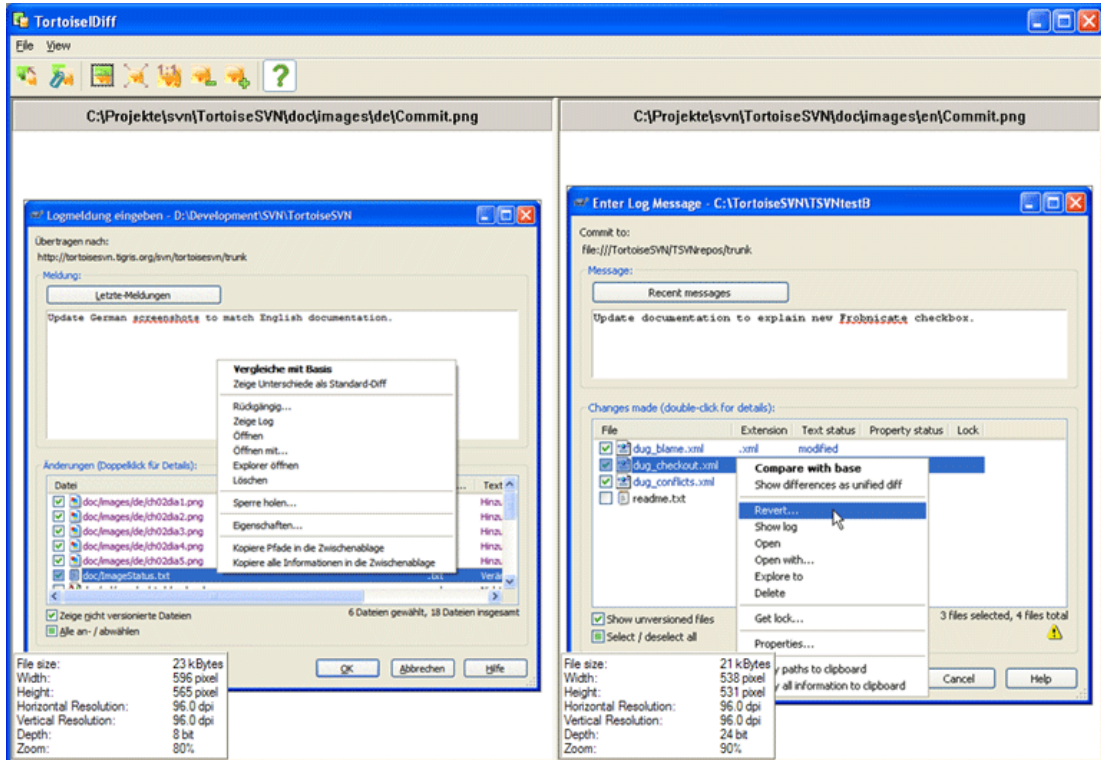


Figura 4.25. O leitor de diferenças de imagem

TortoiseSVN → Comparar para qualquer formato de imagem comum, irá arrancar o TortoiseIDiff para mostrar diferenças de imagem. Por defeito as imagens são mostradas lado-a-lado, mas em vez de, podes usar o menu Vista ou a barra de ferramentas para comutar para uma vista cima-baixo, ou se preferires podes sobrepor as imagens e fingir que estás a usar uma mesa de luz.

Naturalmente que podes aplicar mais ou menos zoom e posicionar a imagem. Podes também posicionar a imagem apenas por arrastá-la com o botão direito do rato. Se seleccionares a opção **Ligar posições das imagens**, os controlos de posicionamento (barras e rolar, roda do rato) ficaram ligados.

Uma caixa de informação de imagem mostra os detalhes sobre a mesma, como o tamanho em píxeis, resolução e profundidade de cor. Se esta caixa está a atrapalhar usa **View → Image Info** para a esconder. Podes obter a mesma informação numa etiqueta, se mantiveres o rato sobre a barra de título da imagem.

Quando as imagens estão sobrepostas, a intensidade relativa das mesmas (mistura alfa) é controlada por um controlo de deslocamento no lado esquerdo. Podes clicar em qualquer ponto do controlo de deslocamento para configurares a mistura directamente, ou podes arrastá-lo para alterares a mistura iterativamente. **Ctrl+Shift-Rodal** para alterar a mistura.

O botão por cima do controlo de deslocamento alterna entre as misturas de 0% e 100%, e se efectuares duplo clique no botão, a mistura alterna automaticamente a cada segundo até clicares no botão novamente. Isto pode ser útil quando olhamos para pequenas alterações múltiplas.

Por vezes queres ver uma diferença em vez de uma mistura. Poderás ter os ficheiros de imagem de duas revisões de um placa de circuito impresso e querer ver quais as pistas que foram alteradas. Se desactivares

o modo mistura alfa, a diferença irá ser mostrada como uma *XOR* dos valores de cor dos píxeis. Áreas inalteradas serão brancas e as alterações coloridas.

4.10.5. Ferramentas de Comparação/Integração

Se as ferramentas fornecidas não fazem o que precisas, experimenta uma das muitas ferramentas comerciais ou open-source disponíveis. Toda a gente tem as suas favoritas e esta lista está longe de estar completa, no entanto, aqui estão algumas que poderás levar em consideração:

WinMerge

WinMerge [<http://winmerge.sourceforge.net/>] é uma grande ferramenta de comparação open-source que também pode lidar com pastas.

Perforce Merge

O Perforce é um SCR comercial no entanto podes descarregar gratuitamente a ferramenta de comparar/integrar. Mais informações em *Perforce* [<http://www.perforce.com/perforce/products/merge.html>].

KDiff3

KDiff3 é uma ferramenta de comparação gratuita que também suporta pastas. Podes descarregá-la em *here* [<http://kdiff3.sf.net/>].

ExamDiff

O ExamDiff padrão é freeware. Esta ferramenta suporta ficheiros mas não pastas. O ExamDiff Pro é shareware e adiciona um número de guloseimas que incluem, comparação de pastas e capacidades de edição. Em ambas edições existe suporte para unicode na versão 3.2 e superior. Podes descarregá-las em *PrestoSoft* [<http://www.prestosoft.com/>].

Beyond Compare

Semelhante ao ExamDiff Pro, esta é uma excelente ferramenta de comparação shareware que suporta comparação de pastas e unicode. Descarrega-a em *Scooter Software* [<http://www.scootersoftware.com/>].

Araxis Merge

O Araxis Merge é uma útil ferramenta comercial para comparar e integrar ficheiros e pastas. Executa uma comparação tripartida em integrações e tem links de sincronização, para uso se alteraste a ordem das funções. Descarrega-a em *Araxis* [<http://www.araxis.com/merge/index.html>].

SciTE

Este editor de texto inclui coloração de sintaxe para comparações unificadas, tornando muito fácil a sua leitura. Descarrega-a em *Scintilla* [<http://www.scintilla.org/SciTEDownload.html>].

Notepad2

O Notepad2 foi desenhado como substituto para o programa padrão Bloco-de-notas, do Windows, sendo baseado no controlo de edição open-source Scintilla. Bem como, sendo bom para ver comparações unificadas é muito melhor que o Bloco-de-notas do Windows para a maioria dos trabalhos. Descarrega-o grátis em *here* [<http://www.flos-freeware.ch/notepad2.html>].

Consultar [Secção 4.30.5, “Preferências de Programas Externos”](#) para mais informações em como configurar o TortoiseSVN para usar essas ferramentas.

4.11. Adicionar Novos Ficheiros e Pastas

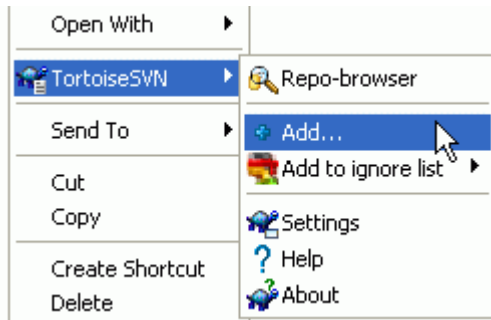


Figura 4.26. Menu de contexto do Explorador para ficheiros não versionados

Se criastes novos ficheiros e/ou pastas durante o teu processo de desenvolvimento, então precisas de adicioná-los também ao controlo de versões. Selecciona o(s) ficheiro(s) e/ou pasta e usa TortoiseSVN → Adicionar...

Após adicionares os ficheiros/pastas ao controlo de versões, o ficheiro aparece com o ícone de sobreposição adicionado, o que significa que terás primeiro de submeter a tua cópia de trabalho para tornar esses ficheiros/pastas disponíveis a outros programadores. Adicionar um ficheiro/pasta *não* afecta o repositório!



Muitas Adições

Podes também usar o comando Adicionar em pastas já versionadas. Neste caso, a caixa de diálogo adicionar irá mostrar todos os ficheiros não versionados dentro dessa mesma pasta versionada. Isto ajuda no caso de teres muitos ficheiros novos, e precisares de os adicionares de uma só vez.

Para adicionar ficheiros, de fora da tua cópia de trabalho, podes usar o manípulo para arrastar-e-largar:

1. Selecciona os ficheiros que queres adicionar
2. Arrasta-os com o botão direito para a nova localização dentro da cópia de trabalho
3. Larga o botão direito do rato
4. Selecciona Menu de Contexto → SVN Copiar e adicionar arquivos a esta Cópia de Trabalho.
Os ficheiros serão então copiados para a cópia de trabalho e adicionados ao controle de versões.

Podes também adicionar ficheiros de dentro da cópia de trabalho, simplesmente arrastando com o botão direito e largando-os na caixa de diálogo submeter.

Se adicionaste um ficheiro ou pasta por engano, podes desfazer a adição, antes de submeteres, usando TortoiseSVN → Desfazer adição....

4.12. Copiando/Movendo/Renomeando Ficheiros e Pastas

Acontece com frequência que já tens os ficheiros que precisas, noutra projecto no teu repositório, e que queres simplesmente copiá-los. Podias simplesmente copiar os ficheiros e adicioná-los como descrito acima, mas isso não te preservaria qualquer história. E se posteriormente tu corriges um problema nos ficheiros originais, só poderás integrar a correcção automaticamente se a nova cópia estiver relacionada com o original no Subversion.

A maneira mais fácil para copiar ficheiros e pastas, dentro da cópia de trabalho, é utilizar o menu de arrasto com o botão direito. Quando tu arrastas-com-botão-direito um ficheiro ou pasta de uma cópia de trabalho para outra, ou mesmo dentro da mesma pasta, aparece o menu de contexto quando libertas o rato.

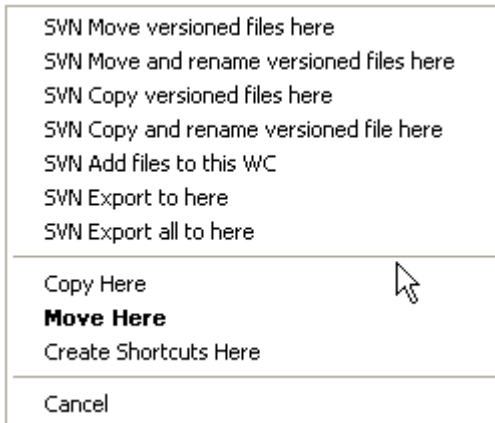


Figura 4.27. Menu arrastar com o botão direito para uma pasta sob controlo de versões.

Agora, podes copiar o conteúdo existente e versionado para uma nova localização, possivelmente renomeando-o ao mesmo tempo.

Tu podes também copiar ou mover ficheiros versionados dentro da cópia de trabalho, ou mesmo entre duas copias de trabalho, usando o familiar metodo cortar-e-colar. Usar o Copiar ou Colar padrão do Windows, para copiar um ou mais itens versionados para a área de transferencia (clipboard). Se a área de transferência contém tais itens versionados, poderás então usar o TortoiseSVN → Colar (nota: não é o Colar padrão do Windows) para copiar ou mover esses itens, para a nova localização da cópia de trabalho.

Podes copiar ficheiros e pastas da tua cópia de trabalho para outra localização no repositório usando TortoiseSVN → Ramo/Etiqueta. Consultar [Secção 4.19.1, “Criando um Ramo ou Etiqueta”](#) para mais detalhes.

Podes localizar uma versão antiga de um ficheiro ou pasta na caixa de diálogo do registo e, copiá-lo para uma nova localização no repositório directamente a partir da caixa de diálogo do registo, usando Menu de contexto → Criar ramo/etiqueta a partir da revisão. Consultar [Secção 4.9.3, “Obtendo Informação Adicional”](#) para mais detalhes.

Podes também usar o navegador de repositório para localizar o conteúdo que pretendes e, copiá-lo para a tua cópia de trabalho directamente do repositório ou copiar entre duas localizações, dentro do repositório. Consultar [Secção 4.24, “O Navegador de Repositório”](#) para mais detalhes.



Não consigo copiar entre repositórios

Whilst you can copy and files and folders *within* a repository, you *cannot* copy or move from one repository to another while preserving history using TortoiseSVN. Not even if the repositories live on the same server. All you can do is copy the content in its current state and add it as new content to the second repository.

Se tens duvidas que dois URLs, no mesmo servidor, se refiram ao mesmo ou a diferentes repositórios, usa o navegador de repositório para abrir um URL e descobrir onde está a raiz do repositório. Se conseguires ver as das localizações numa janela do navegador de repositório, então eles estão no mesmo repositório.

4.13. Ignorando Ficheiros E Pastas

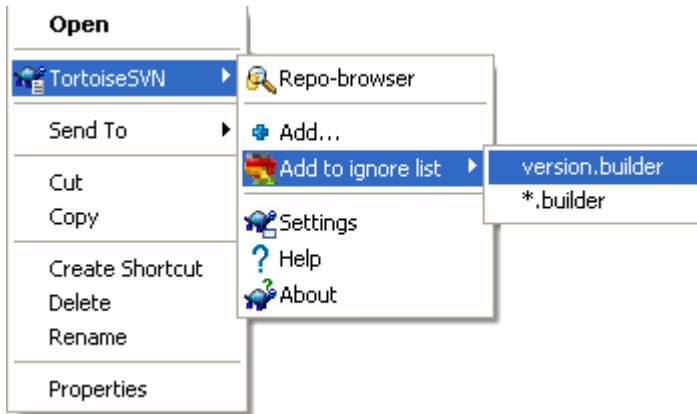


Figura 4.28. Menu de contexto do Explorador para ficheiros não versionados

Na maioria dos projectos terás ficheiros e pastas que não deverão estar sujeitos ao controlo de versões. Esses poderão incluir ficheiros criados pelo compilador, `*.obj`, `*.lst`, talvez uma pasta de produto usada para armazenar o executável. Sempre que submetas alterações o TortoiseSVN mostra os ficheiros não versionados, que enchem a lista de ficheiros na caixa de diálogo submeter. Está claro que podes desligar essa visualização, mas poderás então te esquecer de adicionar um novo ficheiro fonte.

A melhor maneira para evitar esses problemas é adicionar os ficheiros derivados à lista de ignorados do projecto. Desse modo, eles nunca irão aparecer na caixa de diálogo submeter, mas os ficheiros fonte genuinamente não versionados serão ainda assinalados.

Se tu clicares com o botão direito num único ficheiro não versionado e seleccionares, a partir do menu de contexto, o comando TortoiseSVN → Adicionar à lista de ignorados, um submenu aparecerá permitindo-te seleccionar só aquele ficheiro ou todos com a mesma extensão. Se seleccionares múltiplos ficheiros, não haverá submenu e só podes adicionar esses ficheiros/pastas específicos.

Se queres remover um ou mais itens da lista de ignorados, clica com o botão direito nesse itens e selecciona TortoiseSVN → Remover da lista de ignorados. Podes ainda aceder à propriedade `svn:ignore` da pasta. Isso permite-te especificar padrões mais gerais usando caracteres de substituição de nome de ficheiros, descritos na seguinte secção. Consultar [Secção 4.17, “Configurações de Projecto”](#) para mais informações para configurar propriedades directamente. Por favor, ter em atenção que cada padrão para ignorar tem de ser colocado numa linha separada, separá-los por espaços não resulta.



Lista Global de Ignorados

Outra maneira de ignorar ficheiros é adicioná-los à *lista global de ignorados*. A grande diferença aqui é que a lista global de ignorados é uma propriedade local. Aplica-se a *todos* os projectos Subversion, mas apenas no PC local. De modo geral é melhor usar a propriedade `svn:ignore` quando possível, porque pode ser aplicado a um projecto em específico e funciona para todos que SVN exportem o projecto. Consultar [Secção 4.30.1, “Preferências Gerais”](#) para mais detalhes.



Ignorando Itens Versionados

Ficheiros e pastas versionados nunca podem ser ignorados - essa é uma funcionalidade do Subversion. Se versionares um ficheiro por engano, lê em [Secção B.8, “Ignorar ficheiros que já estão versionados”](#) para instruções em como o “desversionar”.

4.13.1. Correspondência de Padrões em Listas de Ignorados

O ignorar padrões no Subversion usa caracteres de substituição do nome de ficheiro, uma técnica usada no Unix para especificar ficheiros recorrendo a meta-caracteres e wildcards (caracteres de substituição). Os seguintes caracteres têm significado especial:

*

Corresponde a qualquer cadeia de caracteres, incluindo a cadeia vazia (sem caracteres).

?

Corresponde a qualquer carácter isolado.

[...]

Corresponde a qualquer um dos caracteres no interior dos parênteses rectos. Dentro dos parênteses rectos, um par de caracteres separados por “-” corresponde a qualquer carácter lexicalmente entre os dois. Por exemplo [AGm-p] corresponde a qualquer um de A, G, m, n, o ou p.

Correspondência de padrões é sensível a maiúsculas e minúsculas, o que pode causar problemas no Windows. Podes forçar a insensibilidade a maiúsculas e minúsculas à força, ao emparelhar caracteres, e.g. para ignorar *.tmp, independentemente ao tamanho da letra, podes usar um padrão como *. [Tt] [Mm] [Pp].

Se queres uma definição oficial para caracteres de substituição de nome de ficheiro, podes encontrá-la nas especificações do IEEE para a linguagem de comando da linha de comandos (shell) *Pattern Matching Notation* [http://www.opengroup.org/onlinepubs/009695399/utilities/xcu_chap02.html#tag_02_13].



Não Usar Caminhos na Lista Global de Ignorados

Não deves incluir informação de caminhos no teu padrão. A correspondência de padrões tem como função ser utilizado contra nomes de ficheiros e nomes de pastas. Se queres ignorar todas as pastas CVS adiciona apenas CVS à lista de ignorados. Não há necessidade de especificar CVS */CVS como fazias em versões anteriores. Se queres ignorar todas as pastas tmp quando elas existem dentro da pasta prog, mas não dentro da pasta doc, deverás usar em alternativa a propriedade svn:ignore. Não há maneira fiável de conseguir isto com recurso a padrões globais de ignorados.

4.14. Removendo, Movendo e Renomeando

Ao contrário do CVS, o Subversion permite renomear e mover ficheiros e pastas. Portanto no submenu TortoiseSVN existem entradas de menu para remover e renomear.

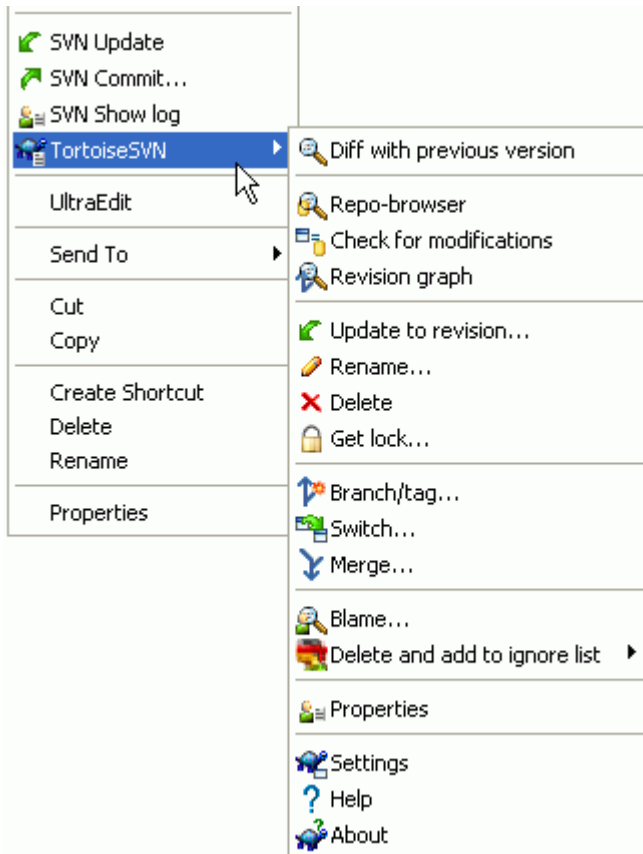


Figura 4.29. Menu de contexto do Explorador para ficheiros versionados

4.14.1. Removendo ficheiros e pastas

Usa TortoiseSVN → Remover para remover ficheiros ou pastas do subversion.

Quando tu executas TortoiseSVN → Remover num ficheiro, este é removido imediatamente da tua cópia de trabalho tal como é marcado para remoção no repositório, na próxima submissão. A pasta pai do ficheiro mostra a sobreposição de ícone “apagado”. Até tu submeteres a alteração, poderás recuperar o ficheiro usando TortoiseSVN → Reverter na pasta pai.

Quando usas o TortoiseSVN → Remover num ficheiro, este permanece na tua cópia de trabalho, mas a sobreposição de ícone muda para indicar que está marcado para remoção. Até submeteres a alteração poderás recuperar a pasta usando o TortoiseSVN → Reverter na própria pasta. Esta diferença de comportamento entre ficheiros e pastas, é parte do Subversion e não do TortoiseSVN.

Se queres remover um tem do repositório, mas mantê-lo localmente como um ficheiro/pasta não versionado, usa Menu de Contexto Estendido → Remover (manter local). Tens de manter premida a tecla **Shift** enquanto clicas com o botão direito no item, no painel lista do explorador (painel direito), de modo a veres isto no menu de contexto estendido.

Se um *ficheiro* é removido via explorador em vez de usar o menu de contexto do TortoiseSVN, a caixa de diálogo submeter mostra esses ficheiros e deixa-te também remove-los do controlo de versões antes da submissão. No entanto se actualizares a tua cópia de trabalho, o Subversion irá detectar o ficheiro em falta e substitui-lo pela última versão no repositório. Se precisas de remover um ficheiro sob controlo de versões, usa sempre TortoiseSVN → Remover para que o Subversion não tenha que adivinhar o que querias realmente fazer.

Se uma *pasta* é removida via explorador em vez de usar o menu de contexto TortoiseSVN, a tua cópia de trabalho ficará quebrada e não poderás efectuar submissões. Se actualizares a tua cópia de trabalho, o Subversion irá substituir a pasta em falta pela última versão do repositório, e então podes apagá-la correctamente usando TortoiseSVN → Delete.



Recuperando um ficheiro ou pasta removido

Se removeste um ficheiro ou pasta e já submeteste essa operação de remoção para o repositório, então o TortoiseSVN → Reverter normal não o pode mais recuperar. Mas o ficheiro ou pasta não está totalmente perdido. Se conheces a revisão em que o ficheiro ou pasta foi removido (se não sabes, usa a caixa de diálogo do registo para a encontrares), abre o navegador de repositório e comuta para essa revisão. Selecciona o ficheiro ou pasta que removeste, clica com o botão direito e selecciona Menu de Contexto → Copiar para..., como destino para essa operação de cópia, selecciona o caminho para a tua cópia de trabalho.

4.14.2. Movendo ficheiros e pastas

Se queres fazer um simples renomear no lugar de um ficheiro ou pasta, usa Context Menu → Alterar nome... Insere o novo nome para o item e está feito.

Se queres mover ficheiros dentro da tua cópia de trabalho, talvez para uma subpasta diferente, usa o manípulo de arrastar-e-largar do botão direito do rato:

1. selecciona os ficheiros ou pastas que queres mover
2. Arrasta-os com o botão direito para a nova localização dentro da cópia de trabalho
3. Larga o botão direito do rato
4. No menu que surge selecciona Menu de Contexto → SVN Mover para aqui item(s) com controlo de versões



Submeter a pasta pai

Visto que renomeações e movimentos são executados como uma remoção seguida por uma adição, deverás submeter a pasta pai do ficheiro renomeado/movido para que a parte removida do renomeado/movido seja apresentada na caixa de diálogo submeter. Se não submeteres a parte removida do renomeado/movido irá ficar para trás no repositório, e quando os teus colegas actualizarem, o ficheiro antigo não será removido. i.e. eles terão *ambos*, o antigo e as novas cópias.

Tu *deves* submeter a pasta renomeada antes de alterar qualquer dos ficheiros dentro dessa pasta, senão a tua cópia de trabalho pode ficar realmente baralhada.

Podes também usar o navegador de repositório para mover itens no mesmo. Consultar [Secção 4.24, “O Navegador de Repositório”](#) para saberes mais.



Não SVN Mover Externos

Não deverás usar os comandos TortoiseSVN Mover ou Renomear numa pasta que foi criada usando `svn:externals`. Esta acção teria como causaria a remoção do item externo do seu repositório pai, provavelmente irritando muitas outras pessoas. Se necessitas de mover uma pasta de externos, deverás usar um vulgar mover de linha de comando (shell), e então ajustar as propriedades `svn:externals` das pastas pai de origem e destino.

4.14.3. Mudando maiúsculas e minúsculas no nome do ficheiro

Efectuando alterações apenas nas maiúsculas e minúsculas no nome de um ficheiro é complicado com o Subversion no Windows, porque durante um curto espaço de tempo durante a mudança de nome, ambos os nomes de ficheiros terão de existir. Como o Windows tem um sistema de ficheiros insensível a maiúsculas e minúsculas, isto não funciona usando o comando Alterar-nome usual.

Afortunadamente existem (pelo menos) dois métodos possíveis para renomear um ficheiro sem perder a sua história de registo. É importante renomeá-lo dentro do Subversion. Apenas renomeá-lo no explorador irá corromper a tua cópia de trabalho!

Solução A) (recomendada)

1. Submeter as alterações na tua cópia de trabalho.
2. Renomear o ficheiro directamente no repositório, de letraMAIÚSCULA para LETRAMaiúscula, usando o navegador de repositório.
3. Actualizar a tua cópia de trabalho.

Solução B)

1. Renomear de letraMAIÚSCULA para letraMAIÚSCULA_ com o comando alterar-nome do submenu do TortoiseSVN.
2. Submeter as alterações.
3. Renomear de letraMAIÚSCULA_ para letraMAIÚSCULA.
4. Submeter as alterações.

4.14.4. Lidando com conflitos de maiúsculas e minúsculas no nome do ficheiro

Se o repositório já contém dois ficheiros com o mesmo nome, diferindo apenas nas miúsculas e minúsculas (e.g. TEST.TXT e test.txt), não conseguirás actualizar ou SVN exportar a pasta pai num cliente Windows. Enquanto o Subversion suporta nomes de ficheiros sensíveis a miúsculas e minúsculas, o Windows não.

Isto por vezes acontece quando duas pessoas submetem, a partir de cópias de trabalho separadas, ficheiros que por acaso têm o mesmo nome mas com uma diferença de maiúsculas e minúsculas. Pode também acontecer quando os ficheiros são submetidos a partir de um sistema, com um sistema de ficheiros sensível a maiúsculas e minúsculas, como o Linux.

Nesse caso tens de decidir qual o que queres manter e remover (ou alterar o nome) do outro no repositório.



Prevenindo dois ficheiros com o mesmo nome

Existe um gancho de servidor disponível em: <http://svn.collab.net/repos/svn/trunk/contrib/hook-scripts/> que previne submissões que resultem em conflitos de maiúsculas e minúsculas.

4.14.5. Reparando Renomeações de Ficheiros

Por vezes o teu IDE amigável irá renomear ficheiros por ti como parte de um processo de refactorização, e claro não irá avisar o Subversion. Se tentares submeter as tuas alterações, o Subversion irá ver o velho ficheiro como desaparecido e o novo como ficheiro não versionado. Poderias apenas adicionar o ficheiro ao controlo de versões, mas assim perderias a sua história, já que o Subversion não sabe que ambos os ficheiros estão relacionados.

Uma melhor maneira será, notificar o Subversion de que esta alteração é na realidade uma alteração de nome, e poderás fazê-lo dentro das caixas de diálogo **Submeter** e **Verificar alterações**. Simplesmente selecciona ambos os ficheiros, o com nome antigo (em falta) e o com o novo nome (não versionado), e usa **Context Menu** → **Reparar movimento** para emparelhar os dois ficheiros como uma renomeação.

4.14.6. Removendo Ficheiros Não Versionados

Normalmente tu configuras a tua lista de ignorados tal que todos os ficheiros gerados sejam ignorados pelo Subversion. Mas se queres limpar todos esse itens ignorados de modo a produzir uma versão limpa? Por norma tu inserias isso na tua makefile, mas se quiseres depurar a tua makefile ou alterar o sistema de build, será útil ter uma maneira de limpar a casa.

O TortoiseSVN disponibiliza tal opção usando **Menu de Contexto Estendido** → **Remover itens sem controle de versão...** Tens de manter premida a tecla **Shift** enquanto clicas com o botão direito na pasta, no painel de lista do explorador (painel direito) de modo a veres isto no menu de contexto estendido. Esta acção irá abrir uma caixa de diálogo que listará todos os ficheiros não versionados em qualquer ponto da tua cópia de trabalho. Podes então seleccionar ou desseleccionar os itens a remover.

Quando tais itens são removidos é usando o cesto de reciclagem para que no caso de te enganares e removeres um ficheiro que deveria estar versionado, poderes ainda recuperá-lo.

4.15. Desfazer Alterações

Se pretendes desfazer todas as alterações que efectuaste num ficheiro desde a última actualização, precisas de seleccionar o ficheiro, clicar com o botão direito para aparecer o menu de contexto e então seleccionar o comando **TortoiseSVN** → **Reverter** Uma caixa de diálogo irá aparecer, mostrando-te os ficheiros que alteraste e podes reverter. Seleccionar os que queres reverter e clica no **OK**.

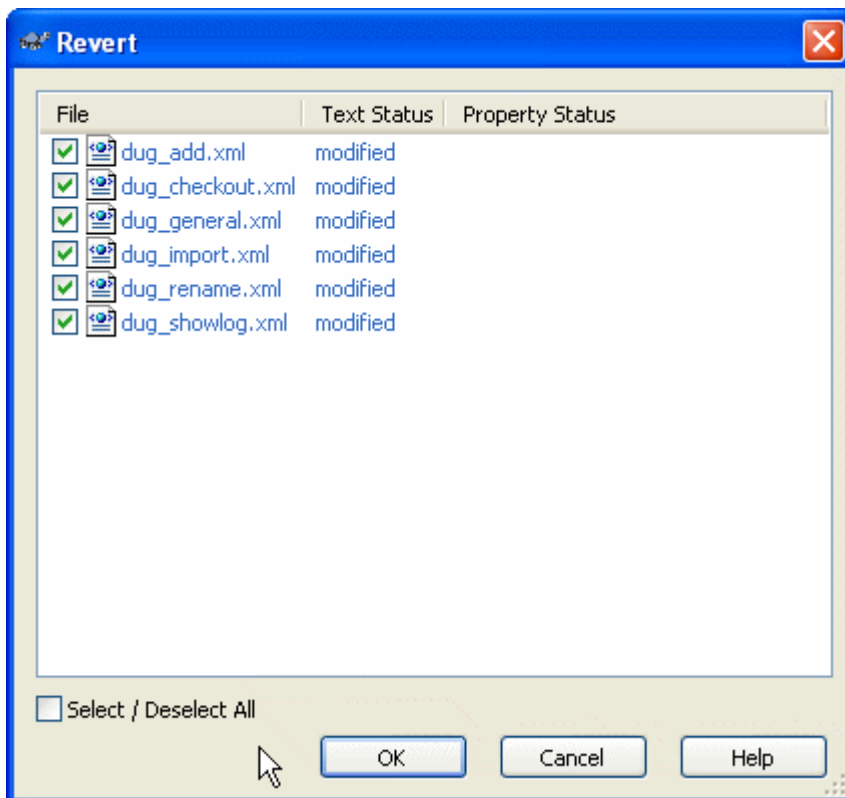


Figura 4.30. Caixa de Diálogo Reverter

Se queres desfazer uma remoção ou renomeação, precisas de usar o Reverter na pasta pai já que o item removido não existirá, para poderes clicar sobre ele com o botão direito.

Se pretendes desfazer a adição de um item, executado no menu de contexto como TortoiseSVN → Desfazer adição.... Na realidade isto é também um reverter, mas neste caso o nome foi alterado para se tornar mais óbvio.

As colunas nesta caixa de diálogo podem ser customizadas da mesma maneira que as colunas na caixa de diálogo Verificar alterações. Consulta [Secção 4.7.3, “Estado Remoto e Local”](#) para mais detalhes.



Desfazer Alterações que já foram Submetidas

Reverter apenas desfaz as tuas alterações locais. *Não* desfaz quaisquer alterações que tenham sido já submetidas. Se pretendes desfazer todas as alterações que já foram submetidas numa revisão particular, consulta [Secção 4.9, “Caixa de Diálogo Registo de Revisões”](#) para mais informação.



Reverter é Lento

Quando reverteres alterações, podes descobrir que essa operação demora muito mais tempo do que estavas à espera. Isto acontece porque a versão modificada do ficheiro é enviado para o cesto de reciclagem, para que possas recuperar as tuas alterações no caso de reverteres por engano. No entanto, se o teu cesto de reciclagem estiver cheio, o Windows tomará mais tempo a encontrar um local para colocar o ficheiro. A solução é simples: ou esvazia o cesto de reciclagem ou desactiva a caixa Usar a reciclagem enquanto revertendo nas preferências do TortoiseSVN.

4.16. Limpar

Se um comando do Subversion não pode concluir com sucesso, talvez devido a problemas de servidor, pode deixar a tua cópia de trabalho num estado inconsistente. Neste caso precisas de usar TortoiseSVN → Limpar na pasta. É uma boa ideia fazer isto no nível de topo da tua cópia de trabalho.

A limpeza tem outro útil efeito secundário. Se a data de um ficheiro foi alterada mas não o seu conteúdo, o Subversion não conseguirá distinguir se foi realmente alterado, à excepção da execução de uma comparação byte a byte com a cópia pura. Se tens muitos ficheiros neste estado, isto torna o adquirir o estado muito lento, que fará com que muitas caixas de diálogo respondam lentamente. Executando a Limpeza na tua cópia de trabalho irá reparar essas marcas de tempo “quebradas” restaurando a verificação de estado a toda a velocidade.



Usar Marcas de Tempo de Submissão

Algumas versões anteriores do Subversion eram afectadas por um problema que provocava um desencontro das marcas de tempo, quando SVN exportavas com a opção verificada Usar marcas de tempo de submeter. Usa o comando de limpeza para acelerar essas cópias de trabalho.

4.17. Configurações de Projecto

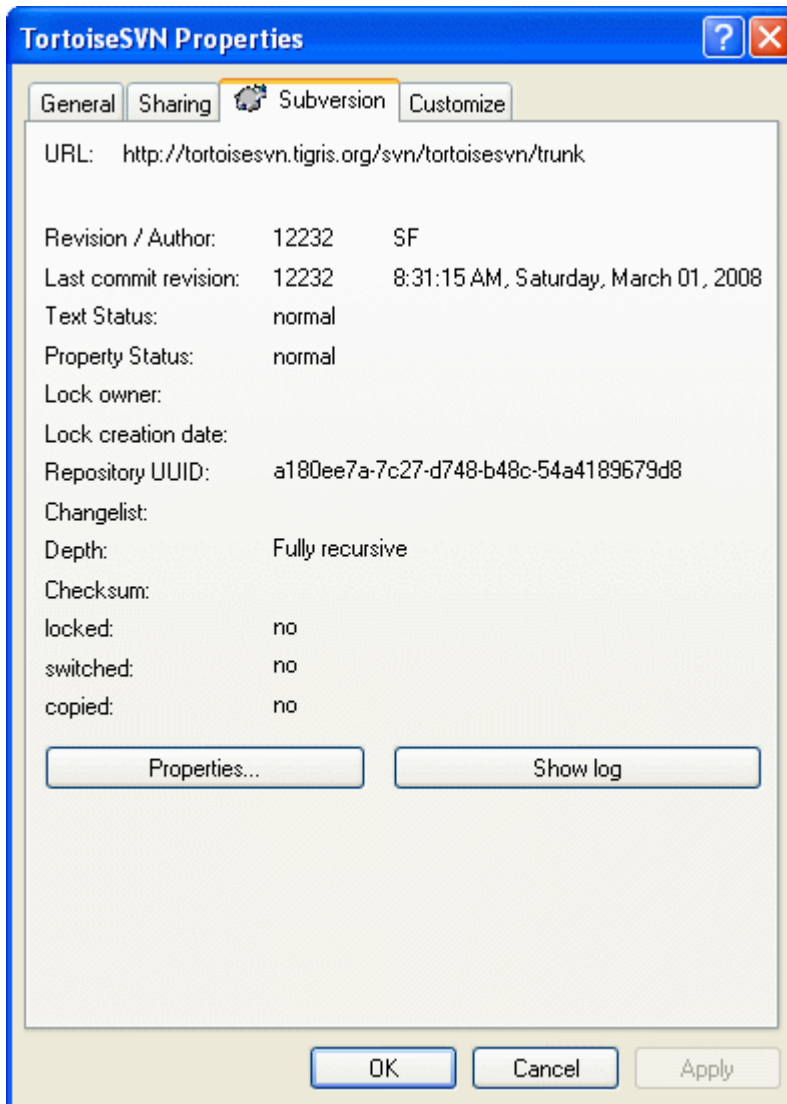


Figura 4.31. Página de propriedades do explorador, aba do Subversion

Por vezes tu queres obter informação mais detalhada sobre um ficheiro/pasta do que a da sobreposição do ícone. Podes obter toda a informação que o Subversion providencia, na caixa de diálogo de propriedades do explorador. Apenas selecciona o ficheiro ou pasta e selecciona no menu de contexto **Menu do Windows** → **propriedades** (nota: esta é a entrada normal do menu de propriedades que o explorador providência, e não a do submenu TortoiseSVN!). Na caixa de diálogo de propriedades o TortoiseSVN adicionou uma nova página de propriedades para ficheiros/pastas, sob o controlo do Subversion, onde podes ver toda a informação relevante sobre o ficheiro/pasta.

4.17.1. Propriedades Subversion

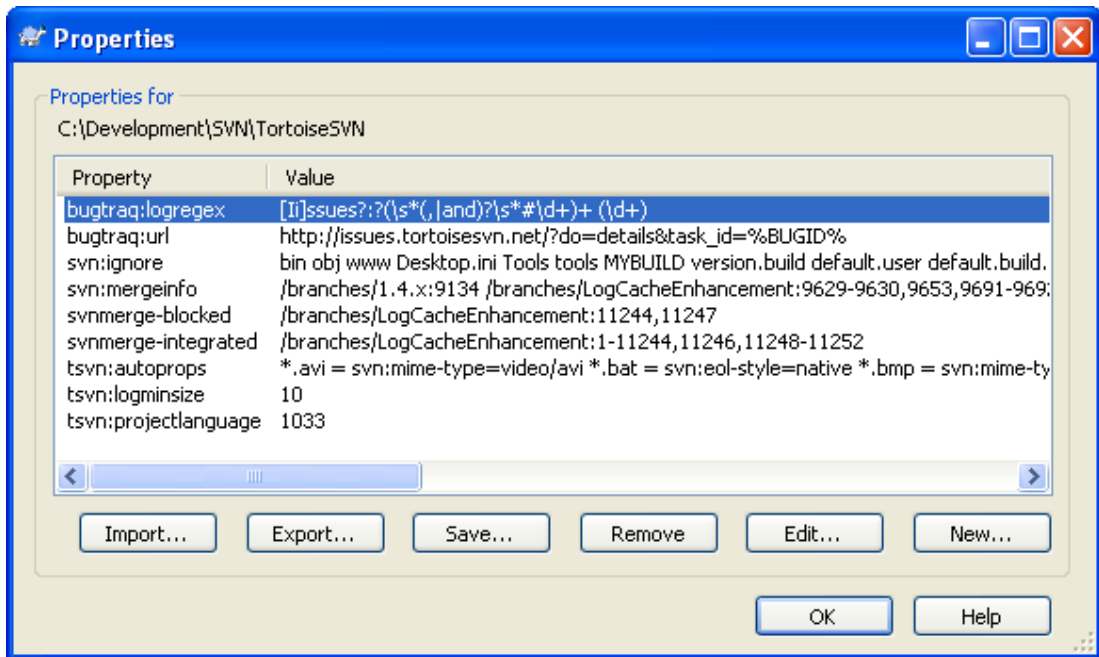


Figura 4.32. Página de propriedades do Subversion

Podes ler e acertar as propriedades do Subversion a partir da caixa de diálogo das propriedades do Windows, mas também a partir de TortoiseSVN → propriedades e do interior das listas de estado do TortoiseSVN, ou seja Menu de contexto → propriedades.

Podes adicionar as tuas próprias propriedades, ou algumas propriedades com significado especial para o Subversion. Essas começam com `svn:`. `svn:externals` é uma dessas propriedades; Descobre como lidar com externos em [Secção 4.18, “Itens Externos”](#).

4.17.1.1. `svn:keywords`

O Subversion suporta expansão de palavras-chave do tipo CVS, que podem ser usadas para embeber informação de revisão e nome de ficheiro dentro do próprio ficheiro. As palavras-chave actualmente suportadas são:

`$Date$`

Data da última submissão conhecida. Isto é baseado em informação obtida quando actualizas a tua cópia de trabalho. *Não* verifica o repositório para tentar encontrar alterações mais recentes.

`$Revision$`

Revisão da última submissão conhecida.

`$Author$`

Autor que fez a última submissão conhecida.

`$HeadURL$`

O URL completo deste ficheiro no repositório.

`Id`

A combinação comprimida das quatro palavras-chave prévias.

Para descobrir como usar estas palavras-chave, consulta o [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] no livro do Subversion, que te dará uma descrição completa dessas palavras-chave e em como activa-las e usá-las.

Para mais informação sobre as propriedades no Subversion, consulta [Special Properties](http://svnbook.red-bean.com/en/1.5/svn.advanced.props.html) [http://svnbook.red-bean.com/en/1.5/svn.advanced.props.html].

4.17.1.2. Adicionando e Editando Propriedades

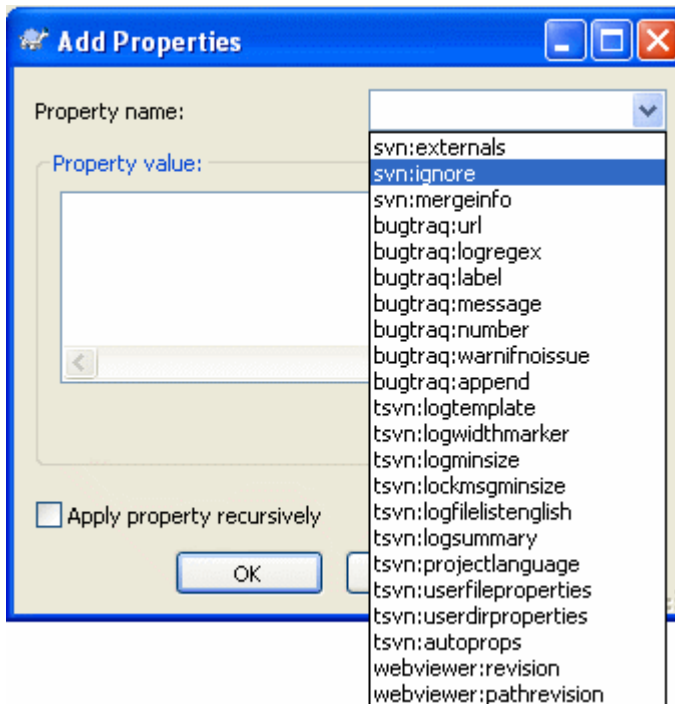


Figura 4.33. Adicionando propriedades

Para adicionar uma nova propriedade clica primeiro em Adicionar.... Selecciona o nome da propriedade pretendida a partir da combo box, ou escreve um nome da tua escolha, e insere um valor na caixa em baixo. As propriedades que requerem valores múltiplos, como uma lista de ignorados, podem ser introduzidos em várias linhas. Clicar em OK para adicionar essa propriedade à lista.

Se queres aplicar a propriedade a muitos itens de uma só vez, selecciona os ficheiros/pastas no explorador e selecciona Menu de contexto → propriedades

Se queres aplicar a propriedade a *cada* ficheiro e pasta na hierarquia abaixo da pasta corrente, verifica a caixa de verificação Recursiva.

Algumas propriedades, como por exemplo `svn:needs-lock`, só podem ser aplicadas em ficheiros, então o nome da propriedade não aparece na lista drop down para pastas. Podes, no entanto, ainda aplicar tal propriedade recursivamente a todos os ficheiros na hierarquia, mas terás de escrever o nome da propriedade manualmente.

Se desejas editar uma propriedade que já exista, selecciona essa propriedade a partir da lista das propriedades existentes e clica em Editar....

Se desejas remover uma propriedade que já exista, selecciona essa propriedade da lista de propriedades existentes e clica em Remover.

A propriedade `svn:externals` pode ser usada para puxar outros projectos do mesmo repositório ou de um repositório completamente diferente. Para mais informação, consultar [Secção 4.18, "Itens Externos"](#).

4.17.1.3. Exportando e Importando Propriedades

Com frequência te encontras a aplicar o mesmo conjunto de propriedades inúmeras vezes, por exemplo `bugtraq:logregex`. Para simplificar o processo de copiar propriedades de um projecto para outro, podes usar a funcionalidade Exportar/Importar.

A partir do ficheiro ou pasta onde as propriedades estão já acertadas, usa TortoiseSVN → propriedades, selecciona as propriedades que desejas exportar e clica em Exportar.... Serás chamado a introduzir o nome do ficheiro onde serão guardadas os nomes das propriedades.

A partir da pasta(s) onde desejas aplicar essas propriedades, usa TortoiseSVN → propriedades e clica em Importar....Serás chamado a introduzir o nome do ficheiro de onde importar, então navega para a localização onde guardaste o ficheiro exportado previamente e selecciona-o. As propriedades serão adicionadas às pastas não recursivamente.

Se queres adicionar propriedades recursivamente a uma árvore segue os passo acima e, então na caixa de diálogo propriedades selecciona cada propriedade à vez e, clica em Editar..., verifica a caixa Aplicar propriedade recursivamente e clica no OK.

O formato do ficheiro Import é binário e proprietário do TortoiseSVN. O seu único propósito é transferir propriedades usando o Import e Export, pelo que não há necessidade de editar esses ficheiros.

4.17.1.4. Propriedades Binárias

O TortoiseSVN pode lidar com valores de propriedades binárias usando ficheiros. Para ler um valor de propriedade binária Guarda... para um ficheiro. Para acertar um valor binário usa um editor hex ou outra ferramenta apropriada para criar o ficheiro com o conteúdo requerido, e Carrega... a partir desse ficheiro.

Apesar das propriedades binárias não serem usadas com frequência, elas podem ser úteis em certas aplicações. Por exemplo, se estiveres a armazenar enormes ficheiros gráficos, ou se a aplicação usada para carregar o ficheiro é enorme, tu poderás querer guardar uma miniatura da imagem como uma propriedade, para que possas obter uma rápida pré-visualização.

4.17.1.5. Configuração automática de propriedades

Tu podes configurar o Subversion e o TortoiseSVN para automaticamente acertar propriedades em ficheiros e pastas, quando são adicionadas ao repositório. Existem duas maneiras de o fazer.

Podes editar o ficheiro de configuração do subversion para activar esta funcionalidade no teu cliente. A página Geral da caixa de diálogo preferências do TortoiseSVN tem um botão editar que te leva lá directamente. O ficheiro de config é um ficheiro de texto simples que controla algum do funcionamento do Subversion. Necessitas de alterar duas coisas: primeiro na secção intitulada `miscellany` descomenta a linha `enable-auto-props = yes`. Segundo precisas de editar a secção abaixo para definires quais as propriedades que queres adicionadas a que tipo de ficheiros. Este método é uma funcionalidade padrão do Subversion e funciona com qualquer cliente. No entanto tem de ser definida individualmente para cada cliente - não existe maneira de propagar essas configurações a partir do repositório.

Um método alternativo é ajustar a propriedade `tsvn:autoprops` em pastas, como descrito na secção seguinte. Este método só funciona para clientes TortoiseSVN, mas é propagado para todas as cópias de trabalho aquando da actualização.

Qualquer que seja o método que escolhas, deverás ter em atenção que o auto-props só é aplicado a ficheiros na altura em que eles são adicionados ao repositório. O auto-props nunca alterará as propriedades dos ficheiros que já estão versionados.

Se queres ter a certeza absoluta que os ficheiros novos têm aplicadas as propriedades correctas, deverás colocar um gancho de pre-commit, de modo a rejeitar submissões onde as propriedades requeridas não estão colocadas.



Submeter as propriedades

As propriedades do Subversion são versionadas. Depois de alterares ou adicionares uma propriedade, terás de submeter essa alteração.



Conflitos em propriedades

Se houver um conflito ao submeter as alterações, porque outro utilizador alterou a mesma propriedade, o Subversion gera o ficheiro `.prej`. Apaga este ficheiro após resolveres o conflito.

4.17.2. Propriedades de Projecto TortoiseSVN

O TortoiseSVN tem algumas propriedades próprias, e essas começam por `tsvn:`.

- `tsvn:logminsize` fixa o tamanho mínimo da mensagem de registo para a submissão. Se inserires uma mensagem mais curta do que a fixada, a submissão é desactivada. Esta funcionalidade é muito útil para te recordar de colocares uma mensagem correcta e descritiva para cada submissão. Se esta propriedade não for fixada, ou o valor for zero, mensagens de registo vazias serão permitidas.

`tsvn:lockmsgminsize` fixa o tamanho mínimo de uma mensagem de bloqueio. Se introduzires uma mensagem mais curta do que aqui especificada, o bloqueio é desactivado. Esta funcionalidade é muito útil para te lembrar de colocar uma mensagem correcta e descritiva para cada bloqueio que obteres. Se esta propriedade não for fixada, ou o valor for zero, mensagens de bloqueio vazias serão permitidas.

- `tsvn:logwidthmarker` é usada por projectos que requerem que as mensagens de registo sejam formatadas com um comprimento máximo (tipicamente 80 caracteres) antes de uma quebra de linha. Fixando esta propriedade para um valor diferente de zero irá fazer duas coisas à caixa de diálogo de introdução de mensagens de registo: coloca um marcador para indicar o comprimento máximo, e desactiva o ajuste de palavras na visualização, para que possas ver se o texto que inseriste é muito longo. Nota: esta funcionalidade só funcionará correctamente se tu tiveres uma fonte de comprimento fixo seleccionada para as mensagens de registo.
- `tsvn:logtemplate` é usado por projectos que têm regras acerca da formatação das mensagens de registo. A propriedade mantém uma cadeia de caracteres multi-linha que será inserida na caixa de mensagens de submissão quando inicias a submissão. Podes então editá-la para incluir a informação requerida. Nota: Se estiveres também a usar a `tsvn:logminsize`, tem a certeza de fixares um comprimento maior que o do template ou perderás o mecanismo de protecção.
- O Subversion permite que tu fixes “autoprops” que serão aplicados a ficheiros recentemente adicionados ou importados, baseado na extensão do ficheiro. Isto depende de cada cliente ter fixado os autoprops apropriados no seu ficheiro de configuração do Subversion. `tsvn:autoprops` pode ser fixado em pastas e essas serão integradas com os autoprops locais quando algum ficheiro for adicionado ou importado. O formato é o mesmo dos autoprops do Subversion, e.g. `*.sh = svn:eol-style=native;svn:executable` fixa duas propriedades em ficheiros com a extensão `.sh`.

Se houver um conflito entre autoprops locais e `tsvn:autoprops`, as configurações de projecto tomam precedência porque são específicas a esse projecto.

- Na caixa de diálogo Submeter tens a opção de colar a lista de ficheiros alterados, incluindo o estado de cada ficheiro (adicionado, modificado, etc). `tsvn:logfilelistenglish` define se o estado do ficheiro é inserido em Inglês ou na língua localizada. Se a propriedade não é fixada, o valor por defeito é `true`.
- O TortoiseSVN pode usar módulos verificadores de ortografia que também são usados pelo OpenOffice e Mozilla. Se tens estes correctamente instalados, esta propriedade irá determinar qual o verificador de ortografia a usar, i.e. em que linguagem as mensagens de registo do teu projecto deverão ser escritas. `tsvn:projectlanguage` fixa o módulo de linguagem que o motor de verificação ortográfica deverá usar quando inseres uma mensagem de registo. Podes encontrar valores para a tua lingua nesta página: [MSDN: Language Identifiers](http://msdn2.microsoft.com/en-us/library/ms776260.aspx) [http://msdn2.microsoft.com/en-us/library/ms776260.aspx].

Podes inserir este valor em decimal ou em hexadecimal se prefixado com 0x. Por exemplo Inglês (US) pode ser introduzido como 0x0409 ou 1033.

- A propriedade `tsvn:logsummary` é usada para extrair a parte da mensagem de registo que será mostrada, na caixa de diálogo de registo, como o seu sumário.

O valor da propriedade `tsvn:logsummary` deverá ser inserida como uma linha de cadeia de caracteres regex que contém um grupo regex. O que corresponder a esse grupo será usado como resumo.

Um exemplo: `\[SUMMARY\]:\s+(.*)` Irá apanhar tudo depois de “[SUMMARY]”, na mensagem de registo, e usa-a como o sumário.

- Quando queres adicionar uma nova propriedade, podes escolher uma da lista da combo box, ou podes inserir um nome de propriedade que desejes. Se o teu projecto usa algumas propriedades customizadas, e se queres que essas propriedades apareçam na lista da combo box (para evitar erros de ortografia quando inserires o nome da propriedade), podes criar uma lista das tuas propriedades customizadas usando `tsvn:userfileproperties` e `tsvn:userdirproperties`. Aplica essas propriedades a uma pasta. Quando tu fores editar as propriedades de qualquer item filho, as tuas propriedades costumizadas irão aparecer na lista de nomes de propriedades pré-definidas.

Algumas propriedades `tsvn:` requerem um valor `true/false`. O TortoiseSVN também entende `yes` como sinónimo para `true` e `no` como sinonimo para `false`.

O TortoiseSVN pode-se integrar com algumas ferramentas de controlo de problemas. Isto usa as propriedades de projecto que começam com `bugtraq:`. Consulta [Secção 4.28, “Integração com Sistemas de controlo de problemas/Gestores de Problemas”](#) para mais informação.

Podes ainda integrar com alguns navegadores de repositório web-based, usando as propriedades de projecto que começam por `webviewer:`. Consulta [Secção 4.29, “Integração com visualizadores de repositório Web-based”](#) para mais informação.



Fixar propriedades de projecto em pastas

Essas propriedades de projecto especiais devem ser fixadas em *pastas* para o sistema funcionar. Quando submetes um ficheiro ou pasta, as propriedades são lidas dessa pasta. Se as propriedades não são encontradas aí, o TortoiseSVN irá procurar na direcção ascendente através da árvore de ficheiros de modo ás encontrar, até encontrar uma pasta não versionada, ou a raiz da árvore (e.g. C:\). Se tens a certeza que cada utilizador só SVN exporta a partir e.g. `trunk/` e não de alguma subpasta, então é suficiente fixar as propriedades no `trunk/`. Se não tens a certeza, deverás fixar as propriedades recursivamente em cada subpasta. A colocação mais profunda da propriedade na hierarquia do projecto sobrepõem-se ás colocações em níveis superiores (mais perto do `trunk/`).

Para *apenas* propriedades de projecto, podes usar a caixa de verificação **Recursivo** para aplicar a propriedade a todas as subpastas na hierarquia, sem também a aplicares a todos os ficheiros.

Quando adicionas novas sub-pastas com o TortoiseSVN, quaisquer propriedades de projecto presentes na pasta pai serão também automaticamente adicionadas à pasta filho.



Cuidado

Apesar de as propriedades de projecto do TortoiseSVN serem extremamente úteis, elas só funcionam com o TortoiseSVN, e algumas não funcionaram com versões recentes. Se as pessoas que trabalham no teu projecto usam uma variedade de clientes Subversion,

ou possivelmente têm versões antigas do TortoiseSVN, poderás querer usar ganchos de repositório para impor as políticas de projecto. Propriedades de projecto só podem ajudar a implementar uma política, nunca podem impô-la.

4.18. Itens Externos

Por vezes é útil construir uma cópia de trabalho a partir de vários SVN exportar diferentes. Por exemplo, poderás querer que diferentes ficheiros ou subpastas venham de diferentes localizações do repositório, ou talvez, até de diferentes repositórios. Se queres que cada utilizador tenha a mesma disposição, podes definir as propriedades `svn:externals` para puxar o recurso especificado para a localização onde será necessário.

4.18.1. Pastas Externas

Digamos que SVN exportas uma cópia de trabalho do `/project1` para `D:\dev\project1`. Seleccionas a pasta `D:\dev\project1`, clicas com o botão direito e escolhes **Menu do Windows** → **Propriedades**, a partir do menu de contexto. Aparecerá a caixa de diálogo **Propriedades**. Em seguida vai à aba **Subversion**, e aí podes fixar as propriedades. Clica **Adicionar...** Selecciona a propriedade `svn:externals` a partir da combobox e, escreve na caixa de edição o URL do repositório, no formato `url folder`, ou se quiseres especificar uma revisão particular, `-rREV url folder`. Podes adicionar múltiplos projectos externos, 1 por linha. Supõem que fixaste essas propriedades no `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
```

Agora clica **Fixar** e submete as tuas alterações. Quando tu (ou outro utilizador qualquer) actualizar a cópia de trabalho, o Subversion irá criar uma subpasta `D:\dev\project1\sounds` e SVN exporta o projecto `sounds`, outra subpasta `D:\dev\project1\quick_graphs`, contendo o projecto `graphics`, e finalmente uma subpasta encadeada `D:\dev\project1\skins\toolkit`, contendo a revisão 21 do projecto `skin-maker`.

Os URLs têm de ser convenientemente escapados, ou não irão funcionar, e.g. deverás substituir cada espaço com `%20`, como mostrado no segundo exemplo, em cima.

Se queres que o caminho local inclua espaços ou outro carácter especial, tens rodeá-lo em aspas, ou usar o carácter `\` (barra invertida), como um carácter de escape, ao estilo da linha de comandos do linux, precedendo cada carácter especial. É claro que isto também significa que tens de necessariamente usar `/` (barra), como delimitador de caminho. De notar que este comportamento é novo no Subversion 1.6 e, não funcionará com clientes mais antigos.



Usa números de revisão explícitos

Deverás considerar seriamente, o uso de números de revisão explícitos em todas as tuas definições externas, como descrito acima. Fazendo-o significa que, terás de decidir a altura de puxar uma diferente fotografia da informação externa, e exactamente qual a fotografia a puxar. Aparte do senso comum de não ser surpreendido por alterações de repositórios de terceiros, que não possas controlar, usar números de revisões explícitos, também significa que se reverts a tua cópia de trabalho para uma revisão anterior, as tuas definições externas irão também reverter para o estado que tinham nessa revisão prévia. O que por sua vez significa que, as cópias de trabalho externas serão actualizadas para corresponder o estado que *elas* tinham, quando o teu repositório estava nessa revisão prévia. Para projectos de software isto pode ser a diferença entre uma construção (build) falhada ou com sucesso, de uma fotografia antiga da tua complexa base de código.



Definições antigas do svn:externals

O formato mostrado aqui foi introduzido no Subversion 1.5. Podes também ver o formato antigo, que tem a mesma informação, numa pasta diferente. O novo formato é preferível, já que suporta várias e úteis funcionalidades descritas abaixo, mas não irá funcionar em clientes antigos. As diferenças são mostradas em *Subversion Book* [<http://svnbook.red-bean.com/en/1.5/svn.advanced.externals.html>].

Se o projecto externo está no mesmo repositório, quaisquer alterações que faças aí, serão incluídas na lista de submissões, quando submeteres o teu projecto principal.

Se o projecto externo está num repositório diferente, qualquer alteração que fizeres no mesmo, será notificada quando submeteres o teu projecto principal, no entanto, terás de submeter essas alterações externas separadamente.

Se usares URLs absolutos nas definições `svn:externals`, e tiveres de transferir a tua cópia de trabalho (i.e., se o URL do teu repositório se alterar), então os teus externos não se alterarão e poderão não mais funcionar.

Para evitar tais problemas, o cliente Subversion, na versão 1.5 ou superior, suporta URLs externos relativos. São suportados quatro diferentes métodos de especificar um URL relativo. Nos exemplos seguintes assume-se que temos dois repositórios: um em `http://example.com/svn/repos-1` e outro em `http://example.com/svn/repos-2`. Temos um SVN exportado do `http://example.com/svn/repos-1/project/trunk` em `C:\Working` e a propriedade `svn:externals` está colocada no trunk.

Relativo à pasta pai

Esses URLs começam sempre com a cadeia de caracteres `../`, por exemplo:

```
../../widgets/foo common/foo-widget
```

. Isto irá extrair `http://example.com/svn/repos-1/widgets/foo` em `C:\Working\common\foo-widget`.

De notar que, o URL é relativo ao URL da pasta com a propriedade `svn:externals` e não, à pasta onde o externo está escrito no disco.

Relativo ao repositório raiz

Esses URLs começam sempre com a cadeia de caracteres `^/`, por exemplo:

```
^/widgets/foo common/foo-widget
```

. Isto irá extrair `http://example.com/svn/repos-1/widgets/foo` em `C:\Working\common\foo-widget`.

Poderás facilmente referir-te a outros repositórios com o mesmo `SVNParentPath` (uma pasta comum contendo vários repositórios). Por exemplo:

```
^/../../repos-2/hammers/claw common/claw-hammer
```

Isto irá extrair `http://example.com/svn/repos-2/hammers/claw` em `C:\Working\common\claw-hammer`.

Relativo ao esquema

URLs a começar com a cadeia de caracteres `//` copiam só a parte do esquema do URL. Isto é útil quando a mesma máquina necessita de ser acedida com esquemas diferentes, dependendo da

localização na rede; e.g. clientes com acesso de intranet usam `http://`, enquanto clientes externos usam `svn+ssh://`. Por exemplo:

```
//example.com/svn/repos-1/widgets/foo common/foo-widget
```

. Será extraído como `http://example.com/svn/repos-1/widgets/foo` ou `svn+ssh://example.com/svn/repos-1/widgets/foo`, dependendo do método usado para SVN exportar o `C:\Working`.

Relativo ao nome do servidor

URLs a começar com a cadeia de caracteres `/`, copiam a parte do URL do esquema e do nome da máquina, por exemplo:

```
/svn/repos-1/widgets/foo common/foo-widget
```

. Extrairá `http://example.com/svn/repos-1/widgets/foo` em `C:\Working\common\foo-widget`. Mas se tu SVN exportares a tua cópia de trabalho a partir de outro servidor em `svn+ssh://another.mirror.net/svn/repos-1/project1/trunk`, então a referência externa irá extrair `svn+ssh://another.mirror.net/svn/repos-1/widgets/foo`.

Podes também especificar uma revisão cavilha depois do URL, se necessário; e.g. `http://sounds.red-bean.com/repos@19`.

Se necessitas de mais informação em como o TortoiseSVN lida com Propriedades, consulta [Secção 4.17, “Configurações de Projecto”](#).

Para saberes mais sobre os diferentes métodos de aceder a subprojectos comuns, consulta [Secção B.6, “Incluir um subprojecto comum”](#).

4.18.2. Ficheiros Externos

A partir do Subversion 1.6 podes adicionar um ficheiro único externo, à tua cópia de trabalho, usando a mesma sintaxe que as pastas. No entanto, existem algumas restrições.

- O caminho para o ficheiro externo deverá colocar o ficheiro numa pasta versionada já existente. No geral faz mais sentido colocar o ficheiro directamente na pasta que tem colocado `svn:externals`, mas poderá ser numa subpasta versionada, se necessário. Por contraste, pastas externas irão automaticamente criar qualquer pasta intermédia não versionada, se necessário.
- O URL para um ficheiro externo deverá estar no mesmo repositório que o URL em que o ficheiro exteno será inserido; ficheiros externos inter-repositórios não são suportados.

Um ficheiro externo comporta-se apenas como qualquer outro ficheiro versionado em vários aspectos, mas eles não podem ser movidos ou apagados, usando os comandos normais; a propriedade `svn:externals` terá de ser então modificada.



Supporte incompleto de ficheiros externos em Subversion 1.6

No Subversion 1.6 não é possível remover um ficheiro externo da tua cópia de trabalho, uma vez que o tenhas adicionado, mesmo que removas também a propriedade `svn:externals`. Terás de SVN exportar uma nova cópia de trabalho para remover o ficheiro.

4.19. Ramificando/Etiquetando

Uma das funcionalidades dos sistemas de controlo de versões, é a sua capacidade para isolar alterações em linhas de desenvolvimento separadas. Esta linha é conhecida como um *ramo*. Ramos são com frequência usados para experimentar novas funcionalidades, sem perturbar a linha principal de desenvolvimento com erros de compilador e bugs. Assim que a nova funcionalidade está suficientemente estável, então o ramo de desenvolvimento é *integrado* de volta para o ramo principal (trunk).

Outra funcionalidade dos sistemas de controlo de versões, é a sua capacidade para marcar revisões particulares (e.g. uma versão de entrega), para que possas a qualquer altura recrear uma certa construção ou ambiente. Este processo é conhecido como *etiquetagem*.

O Subversion não tem comandos especiais para ramificar ou etiquetar, mas em alternativa usa a chamada “cópia barata”. Cópias baratas são semelhantes aos hard links no linux, o que significa que, em vez de efectuar uma cópia completa no repositório, uma ligação interna é criada apontando para a árvore/revisão específica. Como resultado ramos e etiquetas são rápidas a criar e não ocupam praticamente nenhum espaço extra no repositório.

4.19.1. Criando um Ramo ou Etiqueta

Se importaste o teu projecto com a estrutura de pastas recomendada, criar uma versão ramo ou etiqueta é muito simples:

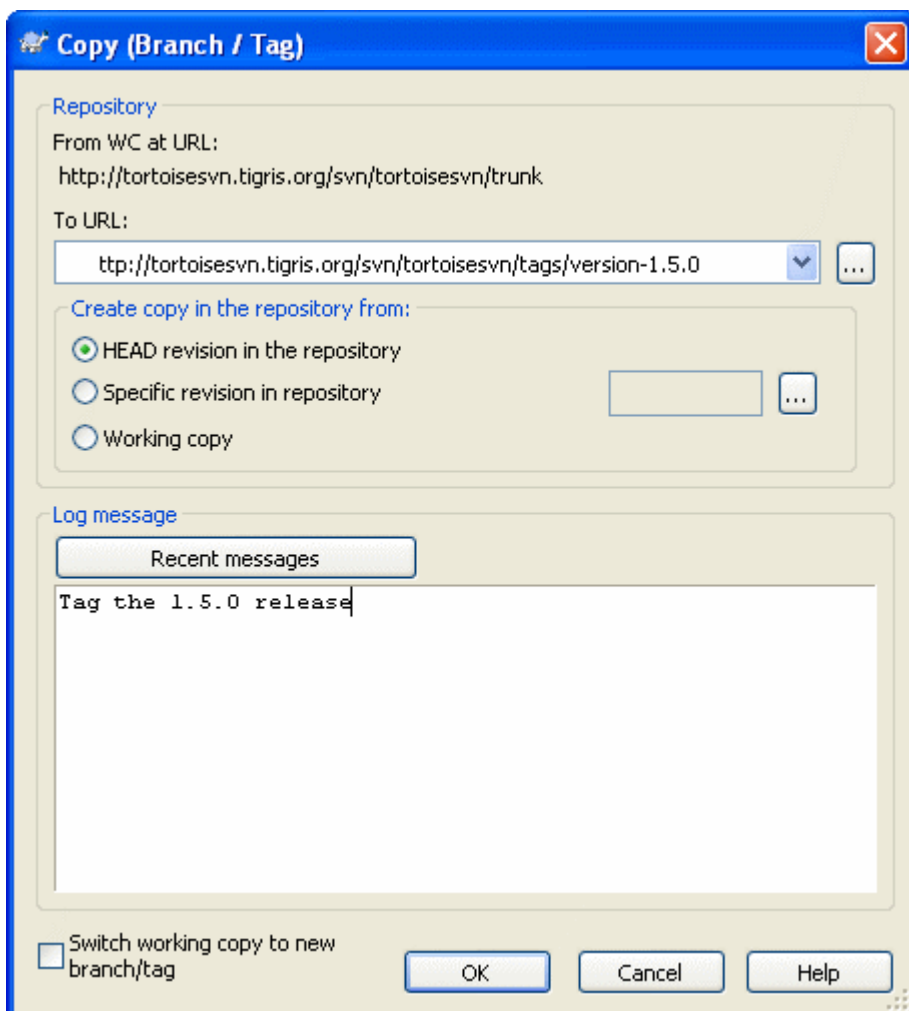


Figura 4.34. A Caixa de Diálogo de Ramificar/Etiquetar

Selecciona na tua cópia de trabalho a pasta para que queres copiar o ramo ou etiqueta, então selecciona o comando TortoiseSVN → Ramificar/Etiquetar...

O URL de destino, por defeito, para o novo ramo será o URL fonte no qual a tua cópia de trabalho está baseada. Necessitarás de editar esse URL de modo a obteres o novo caminho para o teu ramo/etiqueta. Sendo assim, em vez de

```
http://svn.collab.net/repos/ProjectName/trunk
```

poderás querer algo como

```
http://svn.collab.net/repos/ProjectName/tags/Release_1.10
```

Se não te recordas da convenção de nomes que usaste na última vez, clica no botão à direita para abrir o navegador de repositório, para que possas visualizar a estrutura do repositório.

Agora tens de seleccionar a fonte da cópia. Aqui tens três opções:

A revisão HEAD do repositório

O novo ramo é copiado directamente, no repositório, a partir da revisão HEAD. Não é necessário transferir nenhuns dados da tua cópia de trabalho e, o ramo é criado muito rapidamente.

Revisão específica no repositório

O novo ramo é copiado, directamente no repositório, mas podes escolher uma revisão anterior. Isto é útil se te esqueceste de fazer uma etiqueta quando fizeste uma entrega do teu projecto na semana anterior. Se não te lembras do número da revisão, clica no botão à direita para mostrar o registo de versões, e selecciona uma revisão a partir daí. Mais uma vez nenhuns dados são transferidos da tua cópia de trabalho, e o ramo é criado muito rapidamente.

Cópia de trabalho

O novo ramo é uma cópia idêntica da tua cópia de trabalho local. Se actualizaste alguns ficheiros numa versão anterior da tua CT, ou se efectuaste alterações locais. Isso é exactamente o que vai na cópia. Naturalmente este tipo de etiquetas complexas pode envolver transferências de dados da tua CT para o repositório, se este não já lá existir.

Se queres que a tua cópia de trabalho seja automaticamente trocada para o ramo recentemente criado, usa a caixa de verificação **Trocar a cópia de trabalho para o novo ramo/etiqueta**. Mas se o fizeres, tem primeiro a certeza de que a tua cópia de trabalho não contém alterações. Se tiver, essas alterações serão fundidas na CT do ramo, quando fizeres a troca.

Prime OK para submeter a nova cópia para o repositório. Não te esqueças de introduzir uma mensagem de registo. De notar que a cópia é criada *dentro do repositório*.

Ter em atenção que, a não ser que troques a tua cópia de trabalho para o ramo recentemente criado, a criação de um ramo ou etiqueta *não* afecta a tua cópia de trabalho. Mesmo que cries o ramo a partir da tua cópia de trabalho, essas alterações são submetidas para o novo ramo, e não para o trunk, no entanto a tua CT será ainda marcada como modificada em relação ao trunk.

4.19.2. SVN Exportar ou Trocar...

...essa não é (realmente) a questão. Enquanto um SVN exportar descarrega tudo a partir de um ramo do repositório para a tua cópia de trabalho, TortoiseSVN → Trocar... apenas transfere os dados alterados, para a tua cópia de trabalho. Ótimo para a carga na tua rede e bom para a tua paciência. :-)

Para poderes trabalhar com o teu ramo ou etiqueta criado de fresco tens várias opções:

- TortoiseSVN → SVN Exportar para criares uma versão exportada de fresco numa pasta vazia. Podes SVN exportar para qualquer localização no teu disco e, podes criar quantas cópias de trabalho, a partir do teu repositório, que desejes.

- Trocar a tua cópia de trabalho corrente para a cópia recentemente criada no repositório. Mais uma vez selecciona a pasta de topo do teu projecto e usa TortoiseSVN → Trocar... a partir do menu de contexto.

Na próxima caixa de diálogo, insere o URL do ramo que acabaste de criar. Selecciona a botão de rádio Revisão Head e clica no OK. A tua cópia de trabalho é trocada para o novo ramo/etiqueta.

O Trocar funciona tal como Actualizar, na forma em que nunca descarta as tuas alterações locais. Quaisquer alterações que tenhas feito na tua cópia de trabalho, que nunca tenhas submetido, serão integradas quando efectuares a troca. Se não quiseres que isto aconteça, deverás submeter as alterações antes da troca ou reverter a tua cópia de trabalho para uma versão já submetida (HEAD tipicamente).

- Se queres trabalhar no trunk e no ramo, mas não queres o custo de uma versão SVN exportada de fresco, poderás usar o Explorador do Windows para efectuar a cópia do teu trunk exportado para outra pasta e, TortoiseSVN → Trocar... essa cópia para o teu novo ramo.

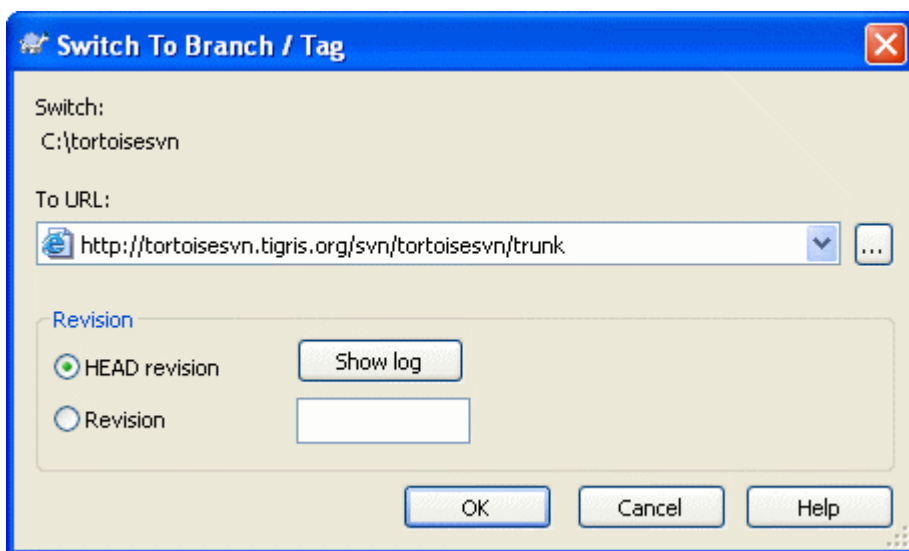


Figura 4.35. A Caixa de Diálogo Trocar

Apesar de o próprio Subversion não fazer distinção entre etiquetas e ramos, a maneira como eles tipicamente são usados difere um pouco.

- Tags are typically used to create a static snapshot of the project at a particular stage. As such they not normally used for development - that's what branches are for, which is the reason we recommended the `/trunk /branches /tags` repository structure in the first place. Working on a tag revision is *not a good idea*, but because your local files are not write protected there is nothing to stop you doing this by mistake. However, if you try to commit to a path in the repository which contains `/tags/`, TortoiseSVN will warn you.
- Poderá ser que necessites de efectuar alterações posteriores numa entrega que tenha sido já etiquetada. A maneira correcta para lidar com isto é criar um novo ramo a partir da etiqueta e submeter o mesmo. Efectua as tuas Alterações nesse ramo e então cria uma nova etiqueta a partir deste ramo, e.g. `Version_1.0.1`.
- Se modificares a cópia de trabalho criada a partir de um ramo, e submeteres, então todas as alterações irão para o novo ramo e *não* para o trunk. Só as modificações são armazenadas, o resto permanece como cópia barata.

4.20. Integrar

Quando ramos são utilizados para manter linhas de desenvolvimento separadas, em alguma altura tu quererás integrar as alterações, que efectuaste num ramo, de volta para o trunk, ou vice-versa.

É importante perceber como funciona no Subversion o ramificar e integrar antes de começares a usá-los, pelo que se poderá tornar algo complexo. É altamente recomendado que leias o capítulo [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] no livro do Subversion, que te dará uma descrição completa, e com muitos exemplos, de como é usado.

O próximo ponto a anotar é que integrar processa-se *sempre* na tua cópia de trabalho. Se queres integrar alterações *num* ramo, tens de ter uma cópia de trabalho nesse ramo e invocar o assistente de integração a partir dessa cópia de trabalho, usando TortoiseSVN → Integrar....

De modo geral é uma boa ideia executar uma integração numa cópia de trabalho não modificada. Se tens outras alterações na tua CT, submete-as primeiro. Se a integração não correr como esperavas, poderás ter de reverter-la, e o comando **Reverter** irá descartar *todas* as alterações incluindo qualquer uma que tenhas feito antes da integração.

Existem três casos de utilização comuns para integrar, que são lidados de maneira ligeiramente diferente, como descrito abaixo. A primeira página do assistente de integração questiona-te a selecciona o método que precisas.

Integrar um intervalo de revisões

Este método cobre o caso em que criaste uma ou mais revisões num ramo (ou no trunk) e queres portar essas alterações para um ramo diferente.

O que estás a pedir ao Subversion para fazer é o seguinte: “Calcula as alterações necessárias para ir [DA] revisão 1 do ramo A [ATÉ] revisão 7 do ramo A, e aplica essas alterações à minha cópia de trabalho (do trunk ou ramo B).”

Reintegrar um ramo

Este método cobre o caso em que criaste um ramo de funcionalidade, como descrito no livro do Subversion. Todas as alterações do trunk foram portadas para o ramo de funcionalidade, semana a semana, e agora que a funcionalidade está concluída, desejas integrá-la de volta no trunk. Porque tens mantido o ramo de funcionalidade sincronizado com o trunk, as últimas versões do ramo e trunk serão absolutamente idênticas à excepção das alterações do teu ramo.

Este é um caso especial da integração de árvore, descrita abaixo, e apenas requer o URL de origem para integrar do (usualmente) teu ramo de desenvolvimento. Ele usa as funcionalidades de rastreamento-de-integração do Subversion, para calcular o intervalo correcto de revisões a usar e executar verificações adicionais, de modo a garantir que o ramo foi totalmente actualizado com as alterações do trunk. Isto garante que tu, por acidente, não desfaças o trabalho que outros tenham submetido para o trunk desde as tuas últimas alterações sincronizadas.

Após a integração, todo o ramo de desenvolvimento foi completamente integrado de volta para a principal linha de desenvolvimento. O ramo é agora redundante e poderá ser removido.

Após teres executado uma reintegração de um ramo, não deverás continuar a usá-lo para desenvolvimento. A razão para tal é a seguinte; se tentares mais tarde re-sincronizar o teu ramo já existente, o seguidor de integração verá a tua reintegração como uma alteração do trunk que não foi ainda integrada no ramo, e irá tentar integrar a integração ramo-para-trunk de volta para o ramo! A solução para isto é criar um novo ramo a partir do trunk para continuar a próxima fase do teu desenvolvimento.

Integrar duas árvores diferentes

Este é um caso mais geral do método de reintegração. O que estás a pedir ao Subversion para fazer é: “Calcula as alterações necessárias para ir [DA] revisão head do trunk [PARA] a revisão head do ramo, e aplica essas alterações à minha cópia de trabalho (do trunk).”. O resultado líquido será que o trunk fica agora exactamente como o ramo.

Se o teu servidor/repositório não suporta o rastreamento-de-integração então esta é a única maneira para integrar um ramo de volta para o trunk. Outro caso ocorre quando tu estás a usar ramos de vendedores e, necessitas de integrar as alterações a seguir a um novo gotejar do vendedor no teu código do trunk. Para mais informação, consulta o capítulo *vendor branches* [<http://svnbook.red-bean.com/en/1.5/svn.advanced.vendorbr.html>] no livro do Subversion.

4.20.1. A Integrar Um Intervalo de Revisões

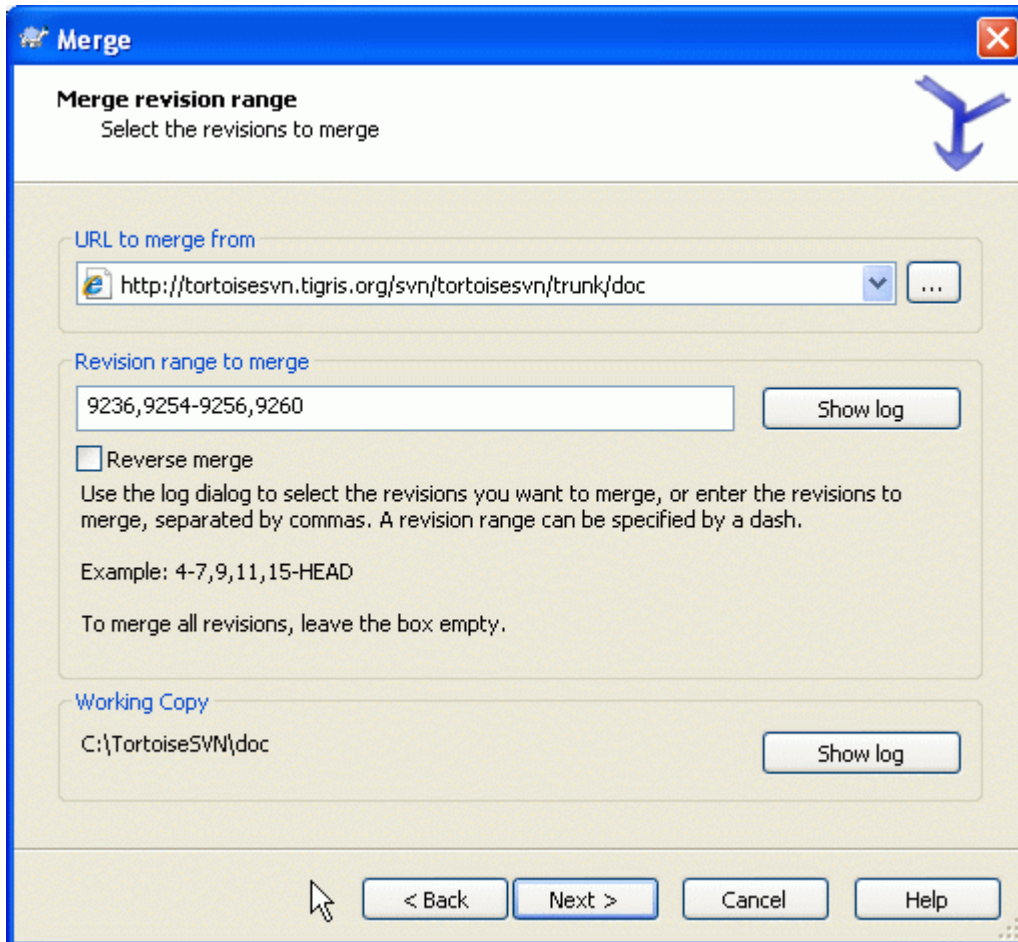


Figura 4.36. O Assistente de Integração - Seleciona o Intervalo de Revisões

No campo **De:** introduz o URL completo da pasta, do ramo ou etiqueta que contém as alterações que queres portar para a tua cópia de trabalho. Podes também clicar ... para navegar no repositório e encontrar o ramo desejado. Se já integraste anteriormente deste ramo, usa apenas a lista drop down que mostra o histórico dos URLs previamente utilizados.

No campo **Intervalo de revisões a integrar** introduz a lista de revisões que desejás integrar. poderá ser uma única revisão, uma lista de revisões específicas separadas por vírgulas, ou um intervalo de revisões separadas por um hífen, ou uma qualquer combinação destes.



Importante

Existe uma diferença importante na maneira como o intervalo de revisões é especificado no TortoiseSVN, comparado com o cliente de linha de comando. A maneira mais fácil para o visualizar é pensar numa vedação com postes e painéis de vedação.

Com o cliente de linha de comando, tu especificas as alterações a integrar usando duas revisões “postes de vedação”, que especificam os pontos *antes* e *depois*.

Com o TortoiseSVN tu especificas uma lista-de-alterações para integrar utilizando “painéis de vedação”. A razão para isto torna-se clara quando usas a caixa de diálogo de registo para especificar as revisões a integrar, onde cada revisão aparece como uma lista-de-alterações.

Se integrares revisões em pedaços, o método descrito no livro do subversion mostra-te a integrar 100-200 desta vez e 200-300 da próxima. Com o TortoiseSVN terias de integrar 100-200 desta vez e 201-300 da próxima.

Esta diferença gerou muita discussão acalorada nas listas de correio. Nós reconhecemos que existe uma diferença para o cliente da linha de comandos, mas cremos que para a maioria dos utilizadores do GUI é mais fácil de compreender o método que implementámos.

A maneira mais fácil para seleccionar o intervalo de revisões que precisas é clicar no **Mostrar registo**, já que este irá listar as alterações mais recentes com os teus comentários de registo. Se queres integrar as alterações a partir de uma única revisão, selecciona apenas essa revisão. Se queres integrar as alterações de várias revisões então, selecciona esse intervalo (usando o modificador usual **Shift**). Clica no **OK** e o números da lista de revisões a integrar serão preenchidas por ti.

Se queres remover alterações integradas, da tua cópia de trabalho, para reverter uma alteração que já foi submetida, selecciona as revisões a reverter e tem a certeza que a caixa de verificação **Reverter integração** está verificada.

Se já integraste algumas alterações deste ramo, espera-se que tenhas introduzido, uma nota da última revisão integrada, na mensagem de registo quando submeteste a alteração. Nesse caso podes usar o **Mostrar Registo** na cópia de trabalho para seguir essa mensagem de registo. Relembrando que estamos a pensar nas revisões como listas-de-alterações, deverás usar a revisão depois do ponto de chegada da última integração, como ponto de partida para esta integração. Por exemplo, se integraste as revisões de 37 a 39 da última vez, então o ponto de partida para esta integração deverá ser a revisão 40.

Se estás a usar as funcionalidades de rastreamento de integração, do Subversion, não precisas de te lembrares quais as revisões que já foram integradas - o Subversion irá recordar-se disso por ti. Se deixas o intervalo de revisões em branco, todas as revisões que não foram ainda integradas serão incluídas. Consulta [Secção 4.20.6, “Rastreamento de Integração”](#) para saberes mais.

Se podem estar outras pessoas a submeter alterações então tem cuidado no uso da revisão HEAD. Poderá não se referir à revisão que estás a pensar se alguém efectuou uma submissão após a tua última actualização.

Clica **Seguinte** e segue para [Secção 4.20.4, “Opções de Integração”](#)

4.20.2. Reintegrar um ramo

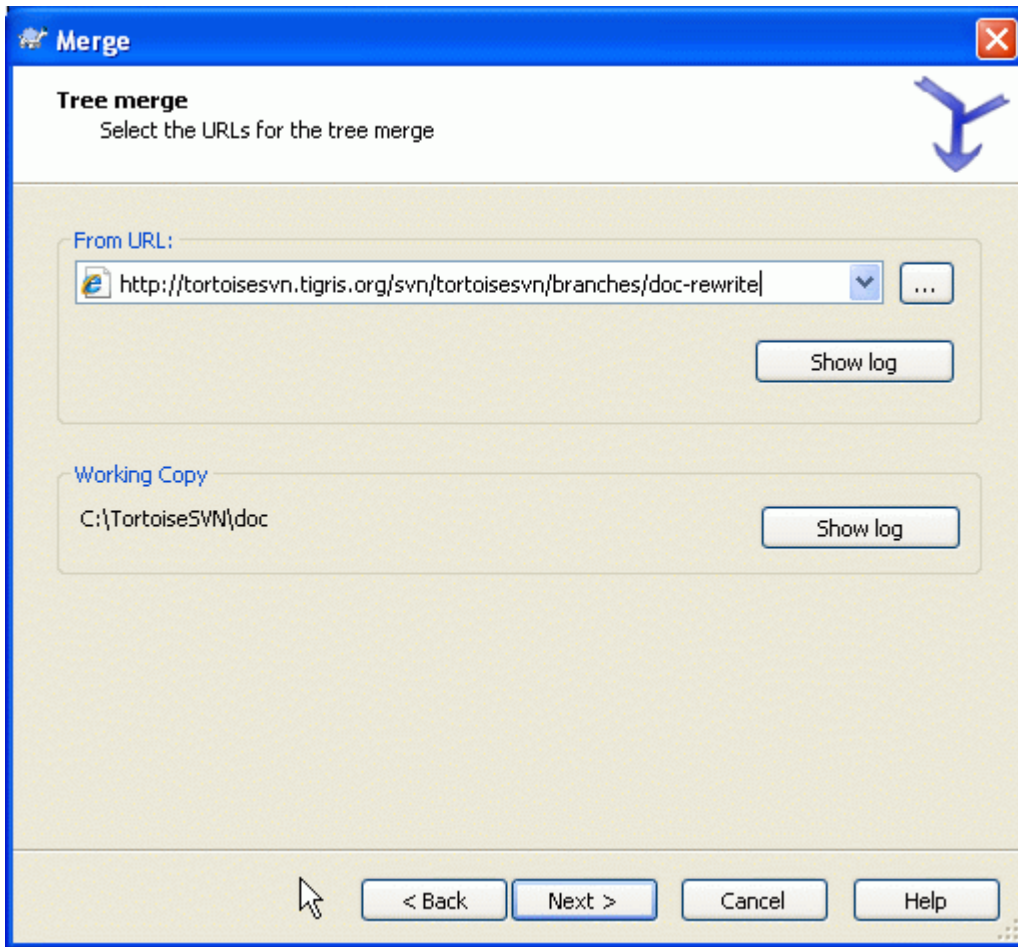


Figura 4.37. O Assistente de Integração - Integração para Reintegrar

Para integrar um ramo de funcionalidade de volta para o trunk, deverá arrancar o assistente de integração de dentro da cópia de trabalho do trunk.

No campo Do URL: introduz o URL completo da pasta do ramo que queres integrar de volta. Podes também clicar ... para navegar o repositório.

Existem algumas condições que se aplicam a uma integração de reintegração. Primeiramente, o servidor deverá suportar o rastreamento de integração. A cópia de trabalho deverá ser de profundidade infinita (sem exportações SVN dispersas), e não deverá ter nenhuma alteração, itens trocados ou itens que foram actualizadas para revisões diferentes da HEAD. Todas as alterações para o trunk feitas durante o desenvolvimento do ramo, deverão ter sido já integradas transversalmente para o ramo (ou marcadas como tendo sido feitas). O intervalo de revisões a integrar será então calculado automaticamente.

4.20.3. A Integrar Duas Árvores Diferentes

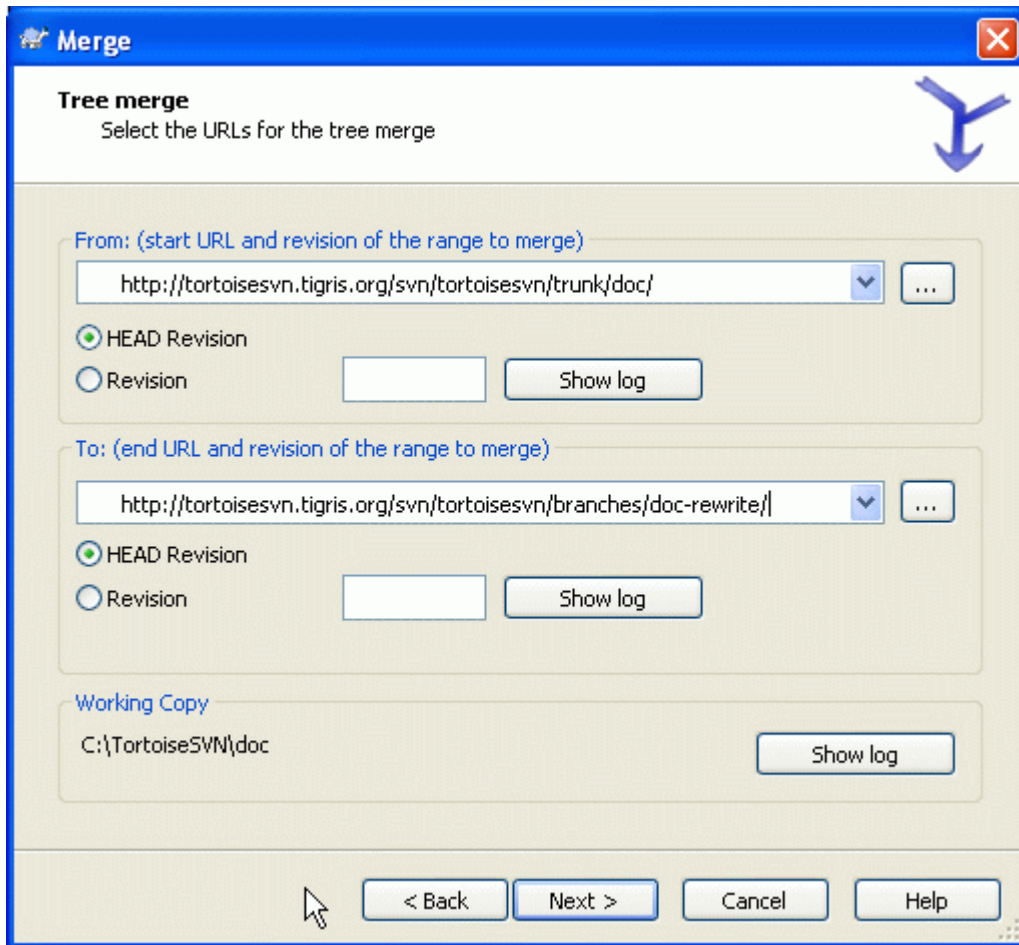


Figura 4.38. O Assistente de Integração - Integração de Árvores

Se estás a usar este método para integrar um ramo de funcionalidade de volta para o trunk, necessitas de arrancar o assistente de integração de dentro da cópia de trabalho do trunk.

No campo De: introduz o URL completo da pasta *trunk*. Isto pode soar errado, mas lembra-te que o trunk é o ponto de partida para o qual queres adicionar as alterações do ramo. Podes também clicar ... para navegar o repositório.

No campo Para: introduz o URL completo da pasta do ramo de desenvolvimento.

Em ambos os campos Da Revisão e Para Revisão, introduz o número da última revisão em que as duas árvores estiveram sincronizadas. Se tens a certeza que ninguém está a fazer submissões, podes usar a revisão HEAD em ambos os casos. Se existe uma hipótese de alguém ter feito uma submissão desde essa sincronização, debes usar números de revisões específicas para evitar perder submissões mais recentes.

Podes também usar o Mostrar Registo para seleccionar a revisão.

4.20.4. Opções de Integração

Esta página do assistente deixa-te especificar opções avançadas antes de começar o processo de integração. Na maior parte do tempo tu podes apenas usar as opções por defeito.

Tu podes especificar a profundidade a usar na integração, i.e. até que ponto deverá a integração penetrar na tua cópia de trabalho. Os termos de profundidade usados estão descritos em [Secção 4.3.1, “Profundidade do Checkout”](#). A profundidade por defeito é a Cópia de trabalho, e é quase sempre o que precisas.

Na maior parte do tempo tu queres que a integração tome conta do histórico do ficheiro, para as que alterações relativas a um antepassado comum sejam integradas. Por vezes tu queres integrar ficheiros que são talvez relacionados, mas não no repositório. Por exemplo tu podes ter versões importadas 1 e 2 de

uma biblioteca de terceiros, em duas pastas separadas. Apesar de eles estarem logicamente relacionadas, o Subversion não tem conhecimento disto porque o que apenas vê são tarballs que importaste. Se tu tentares integrar as diferenças entre essas duas árvores irás ver uma remoção completa seguida por uma adição completa. Para levar o Subversion a usar só as diferenças baseadas no caminho, em vez das diferenças baseadas na história, verifica a caixa **Ignorar ascendência**. Consultar mais sobre este tópico no livro do Subversion, *Noticing or Ignoring Ancestry* [<http://svnbook.red-bean.com/en/1.5/svn.branchmerge.advanced.html#svn.branchmerge.advanced.ancestry>]

Tu podes especificar a forma como os fins-de-linha e espaços-em-branco são lidados. Essas opções são descritas em **Secção 4.10.2, “Opções de Fim-de-Linha e Espaços-Branco”**. O comportamento por defeito é tratar todos os espaços-em-branco e os fins-de-linha como alterações a serem integradas.

Se estás a usar o rastreamento de integração, e queres marcar a revisão como tendo sido integrada, sem actualmente efectuar essa integração aqui, verifica a caixa de verificação **Só regista a integração**. Existem duas razões possíveis para que tu queiras fazer isto. Poderá ser que a integração seja muito complicada para os algoritmos de integração, pelo que inseres as alterações manualmente, e então marcas como integrada, para que o rastreador de integração conheça esse facto. Ou queiras evitar que uma revisão particular seja integrada. Considerando-a como já integrada, irá evitar que a integração ocorra com clientes sensíveis ao seguimento.

Now everything is set up, all you have to do is click on the **Merge** button. If you want to preview the results **Test Merge** performs the merge operation, but does *not* modify the working copy at all. It shows you a list of the files that will be changed by a real merge, and notes those areas where conflicts will occur.

A caixa de diálogo progresso de integração, mostra cada estágio da integração com os intervalos das revisões envolvidas. Isto pode indicar uma revisão a mais do que estavas à espera. Por exemplo se tu pedes para integrar a revisão 123, a caixa de diálogo de progresso irá reportar “Integrar revisões 122 até 123”. Para compreender isto tu precisas de te recordar que a Integração está intimamente relacionada com a Comparação. O processo de integração funciona através da geração de listas de diferenças entre dois pontos no repositório, e então aplicar essas diferenças à tua cópia de trabalho. A caixa de diálogo de progresso simplesmente mostra o ponto de início e fim para a comparação.

4.20.5. Rever os Resultados de Integração

A integração está agora completa. É uma boa ideia dar uma vista de olhos na integração e ver se está como o esperado. Integrar é normalmente algo complicado. Os conflitos surgem com frequência se o ramo divergiu bastante do trunk.

Para clientes e servidores Subversion, anteriores à versão 1.5, nenhuma informação sobre integração é armazenada e as revisões tem de ser seguidas manualmente. Quando tiveres testado as alterações e fores então submeter esta revisão, a tua mensagem de registo deverá incluir *sempre* os números das revisões que foram portadas na integração. Se queres aplicar posteriormente outra integração, necessitarás de conhecer o que já integraste, já que não queres portar uma alteração mais do que uma vez. Para mais informação, consulta *Best Practices for Merging* [<http://svnbook.red-bean.com/en/1.4/svn.branchmerge.copychanges.html#svn.branchmerge.copychanges.bestprac>] no livro do Subversion.

Se o teu servidor e todos os clientes estão a correr o Subversion 1.5, ou versão superior, a funcionalidade de rastreamento de integração irá registar as revisões integradas e evitar que uma revisão seja integrada mais que uma vez. Isto torna a tua vida muito mais simples, já que podes simplesmente integrar o intervalo completo de revisões de cada vez, e saber que só as novas revisões serão realmente integradas.

A gestão de ramos é importante. Se queres manter este ramo actualizado com o trunk deverás ter o cuidado de integrar com frequência, para que o ramo e o trunk não divirjam muito. É claro que deverás evitar ainda a integração repetida de alterações, como descrito acima.



Dica

Se acabaste de integrar um ramo de funcionalidade de volta para o trunk, este agora contém todo o código da nova funcionalidade, tornando-se o ramo obsoleto. Podes então removê-lo do repositório, se for requerido.



Importante

O Subversion não pode integrar um ficheiro com uma pasta e vice-versa - só pastas com pastas e ficheiros com ficheiros. Se clicas num ficheiro e abres a caixa de diálogo de integração, então tens de dar o caminho para um ficheiro nessa caixa de diálogo. Se seleccionas uma pasta e abrires a caixa de diálogo, então terás de especificar um URL de uma pasta para a integração.

4.20.6. Rastreamento de Integração

O Subversion 1.5 introduz facilidades para rastreamento da integração. Quando tu integras alterações de uma árvore para outra, os números de revisões integradas são armazenadas e esta informação pode ser usada para vários propósitos diferentes.

- Podes evitar o perigo de integrar a mesma revisão duas vezes (problema da integração repetida). De cada vez que uma revisão for marcada como tendo sido integrada, integrações futuras, que incluam essa revisão no seu intervalo, irão ignorá-la.
- Quando integras um ramo de volta para o trunk, a caixa de diálogo de registo pode-te mostrar as submissões no ramo como parte do registo do trunk, dando-te melhor rastreamento das alterações.
- Quando mostras a caixa de diálogo de registo de dentro da caixa de diálogo de integração, as revisões já integradas são mostradas a cinzento.
- Quando mostrada a informação de responsabilidade para um ficheiro, podes então escolher mostrar o autor original das revisões integradas, em vez de, a pessoa que efectuou a integração.
- Tu podes marcar revisões como *não integrar* incluindo-as na lista de revisões integradas sem realmente efectuar a integração.

A informação de rastreamento de integração é armazenada na propriedade `svn:mergeinfo` pelo cliente, quando executa uma integração. Quando a integração é submetida, o servidor armazena essa informação numa base de dados, e quando tu requisitas a informação de integração, registo ou responsabilidade, o servidor possa responder de acordo. Para o sistema funcionar correctamente deves assegurar que o servidor, repositório e todos os clientes estejam actualizados. Clientes antigos não irão armazenar a propriedade `svn:mergeinfo` e servidores antigos não irão fornecer essa mesma informação, requisitada pelos novos clientes.

Sabe mais sobre o rastreamento de integração através do Subversion em [Merge tracking documentation](http://subversion.tigris.org/merge-tracking/documentation) [<http://subversion.tigris.org/merge-tracking/index.html>].

4.20.7. Lidando com Conflitos durante a Integração

A integração não corre sempre com suavidade. Por vezes existe um conflito, e se estás a integrar múltiplos intervalos, geralmente irás querer resolver o conflito antes de começares a integrar o próximo intervalo. O TortoiseSVN ajuda-te através deste processo, mostrando-te a caixa de diálogo *conflitos de integração*.

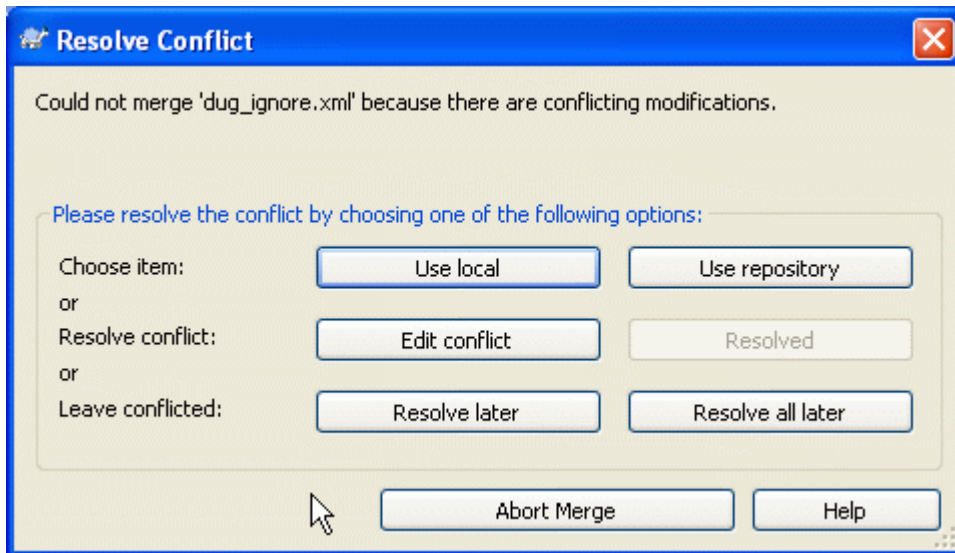


Figura 4.39. A Caixa de Diálogo Conflitos de Integração

Quando o conflito ocorre durante uma integração, tens três maneiras de lidar com ele.

1. You may decide that your local changes are much more important, so you want to discard the version from the repository and keep your local version. Or you might discard your local changes in favour of the repository version. Either way, no attempt is made to merge the changes - you choose one or the other.
2. Normalmente queres observar os conflitos e resolvê-los. Nesse caso, escolhe o **Editar Conflito** que arrancará a ferramenta de integração. Quando estiveres satisfeito com o resultado, clica **Resolvido**.
3. A última opção é adiar a resolução e continuar com a integração. Tu podes escolher fazer isso para o ficheiro correntemente em conflito, ou para todos os ficheiros no resto da integração. No entanto, se existirem mais alterações nesse ficheiro, não será possível completar a integração.

Se não quiseres usar este processo iterativo, existe uma caixa de verificação na caixa de diálogo de progresso de integração **Integrar não-iterativamente**. Se esta está verificada e a integração pode resultar num conflito, o ficheiro é marcado como em conflito e a integração continua. Terás de resolver os conflitos após a conclusão da totalidade da integração. Se esta não está verificada, então antes do ficheiro ficar marcado como em conflito, terás a oportunidade par resolver o mesmo. Isto tem a vantagem de, se um ficheiro sofre múltiplas integrações (múltiplas revisões aplicam uma alteração a esse ficheiro), integrações subseqente irão ter sucesso, dependendo de que linhas são afectadas. Mas, está claro, não te podes afastar para beber um café, enquanto a integração está a decorrer ;)

4.20.8. Integrar um Ramo Completo

Se queres integrar todas as alterações de uma ramo de funcionalidade de volta para o trunk, então podes usar o TortoiseSVN → **Reintegrar integração...** a partir do menu de contexto extendido (mantém permido a tecla **Shift** enquanto clicas com o botão direito no ficheiro).

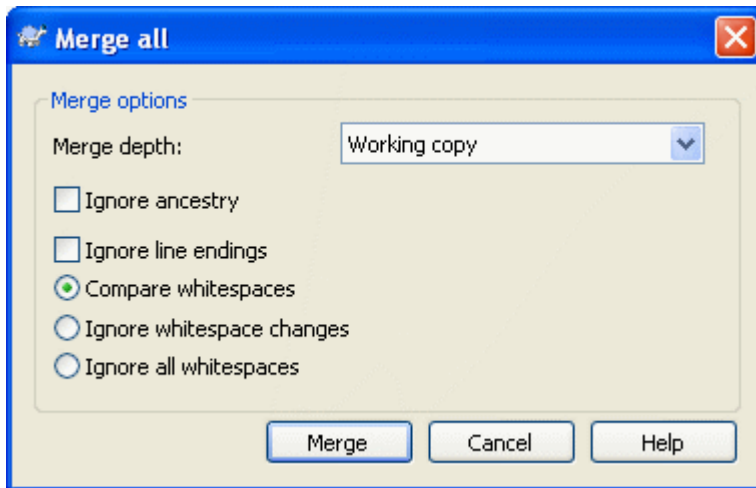


Figura 4.40. Caixa de Diálogo de reintegração de Integração

Esta caixa de diálogo é muito simples. Tudo o que tens de fazer é configurar as opções para a integração, como descrito em [Secção 4.20.4, “Opções de Integração”](#). O resto é feito automaticamente pelo TortoiseSVN usando o rastreamento de integração.

4.20.9. Manutenção do Ramo de Funcionalidade

Quando desenvolves uma nova funcionalidade num ramo separado, é uma boa ideia definir uma política para reintegração quando a funcionalidade estiver completa. Se outro trabalho está a decorrer no `trunk` ao mesmo tempo, poderás descobrir que as diferenças tornar-se-ão significativas ao longo do tempo, e integrar de volta pode-se tornar um pesadelo.

Se a funcionalidade é relativamente simples e o desenvolvimento não será muito longo, então poderás adoptar uma abordagem simples, que é manter o ramo inteiramente separado até a funcionalidade estar completa, então integras as alterações do ramo de volta para o `trunk`. No assistente de integração isto deverá ser um simples **Integrar um intervalo de revisões**, com o intervalo de revisões sendo o intervalo completo de revisões do ramo.

Se a funcionalidade irá demorar mais tempo e tu precisas de ter em conta todas as alterações no `trunk`, então tu necessitas de manter o ramo sincronizado. Isto significa simplesmente que periodicamente integras as alterações do `trunk` no ramo, para que o ramo contenha todas as alterações do `trunk` *mais* a nova funcionalidade. O processo de sincronização usa **Integrar um intervalo de revisões**. Quando a funcionalidade estiver completa, então podes integrá-la de volta para o `trunk` usando **Reintegrar um ramo** or **Integrar duas árvores diferentes**.

4.21. A bloquear

Geralmente o Subversion funciona melhor sem bloqueios usando os métodos “Copiar-Modificar-Integrar” descritos anteriormente em [Secção 2.2.3, “A solução Copiar-Modificar-Integrar”](#). No entanto existem algumas circunstâncias em que poderás necessitar de implementar alguma tipo de política de bloqueio.

- Se estás a usar ficheiros “não integráveis”, como por exemplo, ficheiros de gráficos. Se duas pessoas alteram o mesmo ficheiro, e não é possível a integração, só um dos dois irá perder as suas alterações.
- A tua empresa sempre usou no passado um sistema de controlo de revisões bloqueante, e houve uma decisão da administração de que “o melhor é bloquear”.

Em primeiro lugar necessitas de garantir que o servidor Subversion está actualizado para, pelo menos, a versão 1.2. Versões anteriores não suportam de todo bloqueios. Se está a usar um acesso `file://`, então está claro que só o teu cliente necessita de ser actualizado.

4.21.1. Como Funciona o Sistema de Bloqueio no Subversion

Por defeito nada é bloqueado, e qualquer um que tenha acesso para submeter, pode submeter alterações em qualquer ficheiro a qualquer altura. Outros irão actualizar a sua cópia de trabalho periodicamente e as alterações no repositório serão integradas com as alterações locais.

Se tu *Obteres o bloqueio* num ficheiro, então só podes submeter esse ficheiro. Submissões de outros utilizadores serão bloqueados até libertares o bloqueio. Um ficheiro bloqueado não pode ser modificado de forma alguma no repositório, pelo que não poderá ser apagado ou também renomeado, à excepção do autor do bloqueio.

No entanto outros utilizadores não terão necessariamente conhecimento de que tu obtiveste um bloqueio. A não ser que verifiquem o estado de bloqueio regularmente, a primeira vez que terão conhecimento será quando ao submeterem a operação falhar, o que na maior parte dos casos não será muito útil. Para tornar mais fácil a gestão de bloqueios, existe uma propriedade nova no Subversion `svn:needs-lock`. Quando esta propriedade está configurada num ficheiro (para qualquer valor), quando o ficheiro é SVN exportado ou actualizado a cópia local é colocada em estado de só de leitura, *a não ser que* essa cópia tenha o bloqueio desse ficheiro. Isto actua como um aviso que não deverás alterar esse ficheiro a não ser que primeiro obtenhas o seu bloqueio. Os ficheiros que são, versionados e só de leitura, são marcados com uma sobreposição especial do TortoiseSVN, que indica que necessitas de obter um bloqueio antes de editares.

Os bloqueios são registados por localização da cópia de trabalho tal como por autor. Se tiveres várias cópias de trabalho (em casa, no trabalho) então só poderás obter um bloqueio *numa* dessas cópias de trabalho.

Se um dos teus colegas adquire um bloqueio e vai de férias sem o libertar, o que farás? O Subversion providência um meio de forçar bloqueios. Libertar um bloqueio detido por outro é referido como *Quebrar* o bloqueio, e forçar aquisição de um bloqueio detido por outrém é referido como *Roubar* o bloqueio. Naturalmente essas são acções que não deverás efectuar de ânimo leve, se queres manter a amizade com os teus colegas de trabalho.

Os bloqueios são registados no repositório e um testemunho de bloqueio é criado na tua cópia de trabalho. Se existe uma discrepância, por exemplo se alguém quebrou o bloqueio, o testemunho local do bloqueio fica inválido. O repositório é sempre a referência definitiva.

4.21.2. Obter um Bloqueio

Selecciona o ficheiro(s) na tua cópia de trabalho, para o qual queres adquirir um bloqueio, e então selecciona o comando TortoiseSVN → Obter Bloqueio....

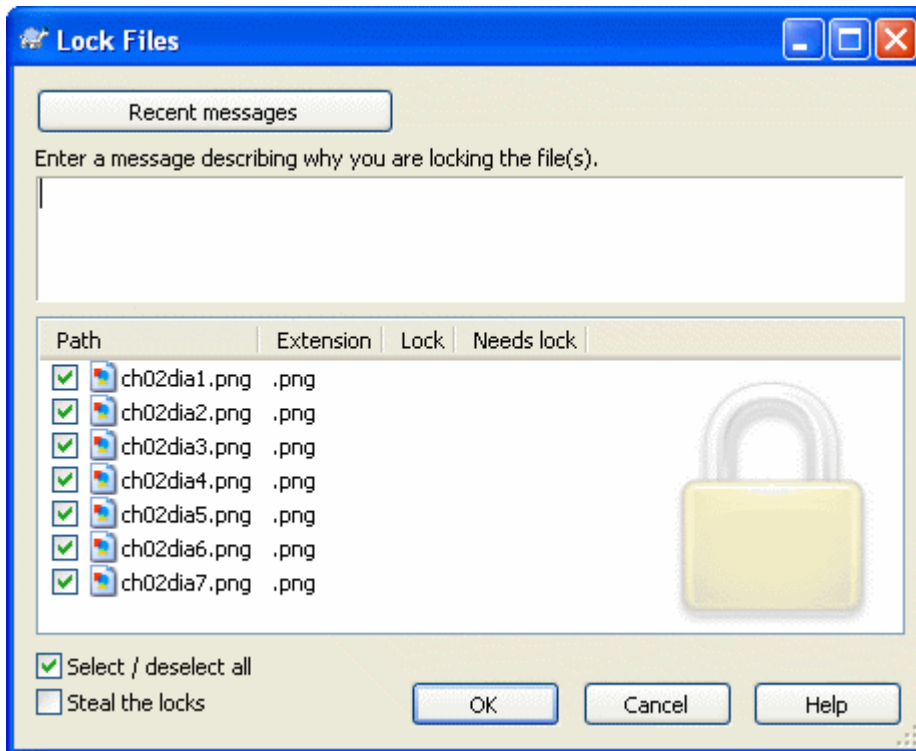


Figura 4.41. A Caixa de Diálogo Bloquear

Uma caixa de diálogo aparece para permitir introduzir um comentário, para que os outros possam ver porquê bloqueaste o ficheiro. O comentário é opcional só é usado correntemente com repositórios baseados em Svnserve. Se (e só se) necessitas de roubar o bloqueio de alguém, verifica a caixa Roubar bloqueio, e clica no OK.

Se seleccionas um ficheiro e então usas TortoiseSVN → Obter bloqueio..., a caixa de diálogo bloquear irá abrir com *todos os* ficheiros em *todas* as subpastas seleccionadas para bloquear. Se realmente queres bloquear uma hierarquia inteira, esta é a maneira de o fazer, mas podes-te tornar muito impopular junto dos teus colegas de trabalho se bloqueias todo o projecto. Usa com moderação...

4.21.3. Libertar um Bloqueio

Para ter a certeza que não te esqueces de libertar um bloqueio que não precisas mais, os ficheiros bloqueados são mostrados na caixa de diálogo submeter e seleccionados por defeito. Se continuares com a submissão, os bloqueios são removidos nos ficheiros seleccionados, mesmos se os ficheiros não foram modificados. Se não queres libertar o bloqueio em certos ficheiros, podes desseleccioná-las (se não foram modificadas). Se queres manter um bloqueio num ficheiro que modificaste, tens de activar a caixa de verificação Manter bloqueios, antes de submeteres as tuas alterações.

Para libertar manualmente um bloqueio, selecciona o(s) ficheiro(s) na tua cópia de trabalho, para os quais queres libertar os bloqueios e selecciona o comando TortoiseSVN → Libertar bloqueio Não há mais nada a introduzir pelo que o TortoiseSVN irá contactar o repositório e libertar os bloqueios. Podes também usar este comando numa pasta, para libertar recursivamente, todos os bloqueios.

4.21.4. Verificar o Estado dos Bloqueio

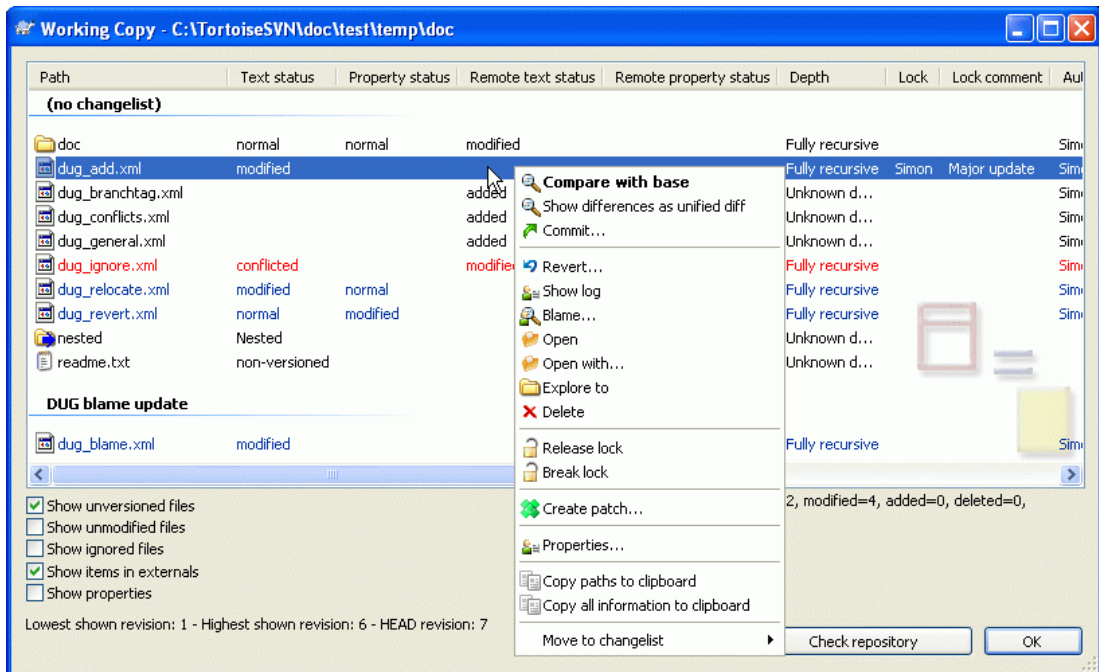


Figura 4.42. A Caixa de Diálogo Verificar Alterações

Para ver os bloqueios que tu e os outros têm, podes usar TortoiseSVN → Verificar alterações.... Testemunhos de bloqueios locais aparecerão imediatamente. Para verificar os bloqueios detidos por outros (e para ver se algum dos teus bloqueios estão quebrados ou roubados) necessitas de clicar em Verificar Repositório.

A partir daqui, do menu de contexto, podes também obter e libertar bloqueios tal como quebrar ou roubar bloqueios detidos por outros.



Evitar Quebrar e Roubar Bloqueios

Se quebrares ou roubares um bloqueio de outro sem lhe dizeres nada, podes potencialmente provocar perda de trabalho. Se estás a trabalhar com tipos de ficheiros não integráveis e roubas o bloqueio de outro, quando libertares o bloqueio, eles ficaram livres para submeter as suas alterações e reescrevendo as tuas. O Subversion não perde dados, mas tu perdeste a protecção de trabalho-em-equipa que o bloqueio te deu.

4.21.5. Tornar os Ficheiros Não-Bloqueados Só de Leitura

Como mencionado acima, a maneira mais eficaz de usar os bloqueios é configurar a propriedade `svn:needs-lock` em ficheiros. Consultar [Secção 4.17, “Configurações de Projecto”](#) para instruções em como configurar propriedades. Ficheiros com esta propriedade configurada serão sempre SVN exportadas e actualizadas com a bandeira só-de-leitura levantada, a não ser que a tua cópia de trabalho tenha um bloqueio.



Como lembrete o TortoiseSVN usa uma sobreposição especial para indicar isto.

Se aplicas uma política em que todo o ficheiro tem de ser bloqueado, então acharás mais fácil usar a funcionalidade auto-props do Subversion para fixar automaticamente a propriedade, de cada vez que adicionas novos ficheiros. Consulta [Secção 4.17.1.5, “Configuração automática de propriedades”](#) para mais informações.

4.21.6. Os Scripts do Gancho de Bloqueio

Quando crias um repositório novo com o Subversion 1.2 ou superior, são criados quatro templates de gancho na pasta `hooks` do repositório. Esses são invocados antes e depois de obter um bloqueio, e antes e depois de libertar um bloqueio.

Será uma boa ideia instalar no servidor um script de gancho `post-lock` e `post-unlock`, que envia um email indicando que ficheiro foi bloqueado. Com tal script instalado todos os teus utilizadores podem ser notificados se alguém bloqueia/desbloqueia um ficheiro. Podes encontrar um exemplo de um script de gancho `hooks/post-lock.tmpl` na pasta do teu repositório.

Podes também utilizar ganchos para proibir quebrar ou roubar bloqueios, ou talvez limitá-lo a um administrador nomeado. Ou talvez queres enviar um email ao dono, quando um dos seus bloqueios é quebrado ou roubado.

Consulta [Secção 3.3, “Scripts de gancho de servidor”](#) para saberes mais.

4.22. Criar e Aplicar Correções

Para projectos de código aberto (como este) toda a gente tem acesso de leitura ao repositório, e toda a gente pode contribuir para o projecto. Então como são controladas essas contribuições? Se toda a gente pudesse submeter alterações o projecto estaria em instabilidade permanente, e provavelmente permanentemente inutilizado. Nesta situação a alteração é gerida através da submissão de um ficheiro de *correção* para a equipa de desenvolvimento, que tem acesso para escrever. Eles podem então rever primeiro a correção e então submetê-la para o repositório ou devolvê-la ao autor.

Ficheiros de correção são simplesmente ficheiros de Comparação-Unificada que mostram as diferenças entre a tua cópia de trabalho e a revisão base.

4.22.1. Criar um Ficheiro de Correção

Primeiro tu precisas de construir e *testar* as tuas alterações. Então em vez de usares TortoiseSVN → Submeter... na pasta pai, seleccionas TortoiseSVN → Criar correção...

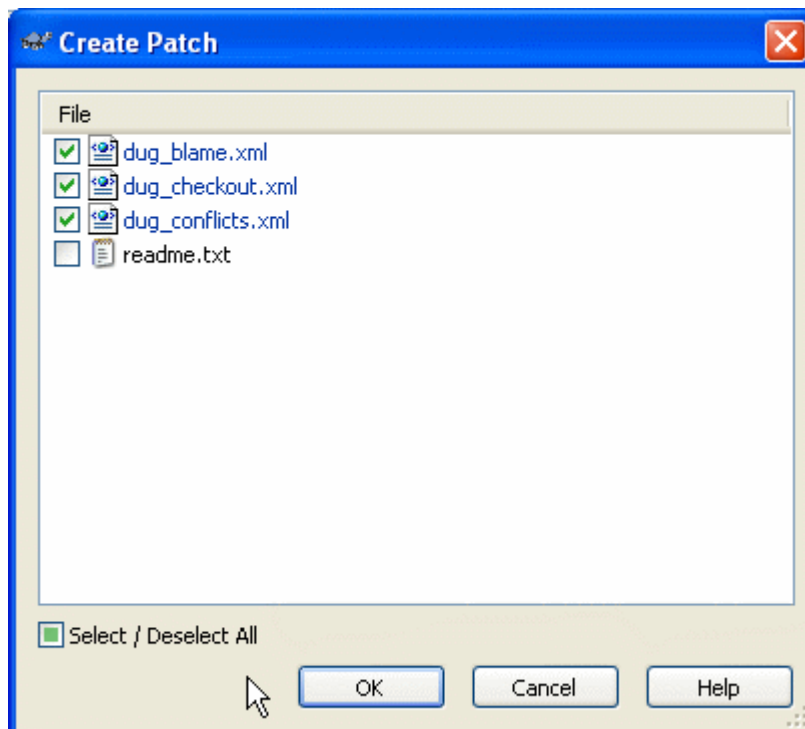


Figura 4.43. A caixa de diálogo de Criar Correção

Agora podes seleccionar os ficheiros que queres incluir na correcção, tal como tu farias numa submissão completa. Isto irá produzir um ficheiro simples contendo um sumário de todas as alterações que efectuaste nos ficheiros seleccionados, desde a última actualização do repositório.

As colunas nesta caixa de diálogo podem ser customizadas da mesma maneira que as colunas na caixa de diálogo **Verificar alterações**. Consulta [Secção 4.7.3, “Estado Remoto e Local”](#) para mais detalhes.

Tu podes produzir correcções separadas, contendo alterações em diferentes conjuntos de ficheiros. É claro que se criaste um ficheiro de correcção, e efectuaste mais alterações aos *mesmos* ficheiros, criando uma segunda correcção, o segundo ficheiro de correcção irá incluir *ambos* os conjuntos de alterações.

Grava apenas o ficheiro usando um nome da tua escolha. Ficheiros de correcções podem ter a extensão que desejes, mas por convenção eles dever usar a extensão `.patch` ou `.diff`. Agora estás pronto para submeter o teu ficheiro de correcção.

Tu podes também guardar a correcção na área de transferência em vez de o fazer para um ficheiro. Poderás fazer isto para que possas colá-lo num email para ser revisto por outros. Ou se tiveres duas cópias de trabalho numa máquina e quiseses transferir alterações de uma para outra, a correcção na área de transferência é a maneira conveniente de o fazer.

4.22.2. Aplicar um Ficheiro de Correcção

Os ficheiros de correcção são aplicados na tua cópia de trabalho. Isto deverá ser feito a partir de uma pasta do mesmo nível da que foi usada para criar a correcção. Se não tens a certeza de qual é, olha apenas para a primeira linha do ficheiro de correcção. Por exemplo, se o primeiro ficheiro a ser trabalhado era `doc/source/english/chapter1.xml` e a primeira linha no ficheiro de correcção `Index: english/chapter1.xml`, então precisas de aplicar a correcção na pasta `doc/source/`. No entanto, mesmo que estejas na cópia de trabalho correcta, se escolheres o nível de pastas errado, o TortoiseSVN irá dar conta e sugere o nível correcto.

De modo a aplicar um ficheiro de correcção à tua cópia de trabalho, necessitas de pelo menos, ter acesso de leitura ao repositório. A razão para isto é que, o programa de integração tem de referenciar as alterações até à revisão em que elas foram efectuadas pelo programador remoto.

From the context menu for that folder, click on TortoiseSVN → Apply Patch... This will bring up a file open dialog allowing you to select the patch file to apply. By default only `.patch` or `.diff` files are shown, but you can opt for “All files”. If you previously saved a patch to the clipboard, you can use **Open from clipboard...** in the file open dialog.

Alternativamente, se o ficheiro de correcção tem uma extensão `.patch` ou `.diff`, podes clicar directamente nele com o botão direito e seleccionar TortoiseSVN → Aplicar correcção.... Neste caso serás solicitado a introduzir a localização da cópia de trabalho.

Estes dois métodos apenas diferem no modo de fazer a mesma coisa. Com o primeiro método, tu seleccionas a CT e navegas até ao ficheiro de correcção. Com o segundo, seleccionas o ficheiro de correcção e navegas até à CT.

Uma vez que tenhas seleccionado o ficheiro de correcção e a localização da cópia de trabalho, o TortoiseMerge irá correr, integrando as alterações do ficheiro de correcção com a tua cópia de trabalho. Uma pequena janela lista os ficheiros que foram alterados. Efectua duplo clique em cada uma de cada vez, para reverter as alterações e guardares os ficheiros integrados.

A correcção do programador remoto foi agora aplicada na tua cópia de trabalho, então precisas agora de submeter de modo a permitir que todos possam aceder às alterações a partir do repositório.

4.23. Quem Alterou Que Linha

Por vezes tu precisas conhecer não só que linhas foram alteradas, mas também quem exactamente alterou linhas específicas num ficheiro. É quando o comando TortoiseSVN → Responsabilizar..., por vezes também referido como *anotar* vem a calhar.

Este comando lista, para cada linha num ficheiro, o autor e a revisão em que a linha foi alterada.

4.23.1. Responsabilidade para Ficheiros

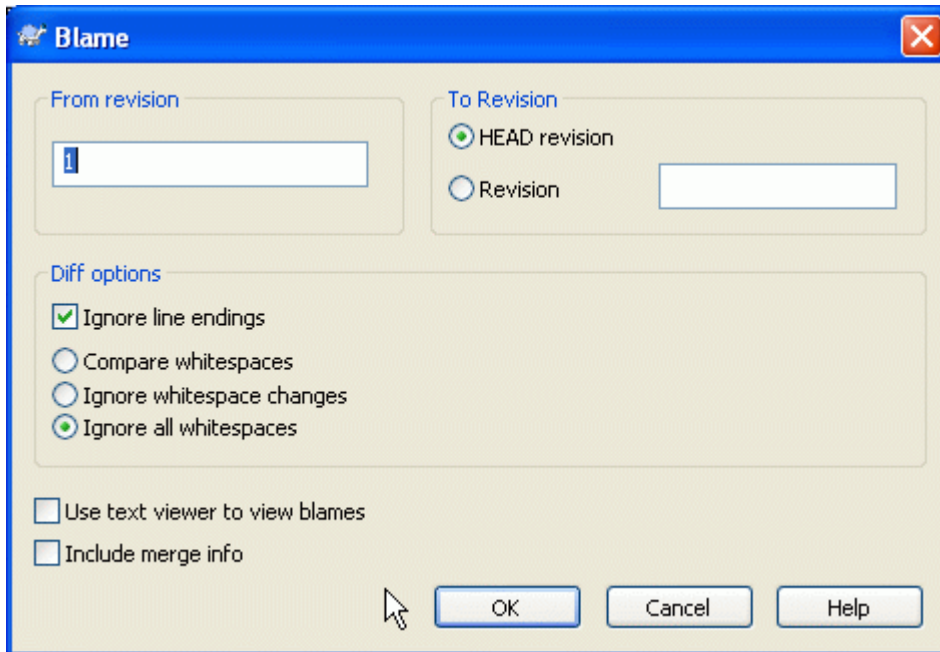


Figura 4.44. A Caixa de Diálogo de Anotar/Responsabilizar

Se não estás interessado nas alterações em revisões anteriores, podes fixar a revisão a partir da qual a responsabilização deve começar. Fixa para 1, se queres a responsabilização para *cada* revisão.

Por defeito o ficheiro responsabilizado é visualizado usando o *TortoiseBlame*, que salienta as diferentes revisões para torná-lo mais fácil de ler. Se desejas imprimir ou editar o ficheiro de responsabilidade, selecciona Usar leitor de Texto para ver responsabilidades

Tu podes especificar a forma como são lidados os fim-de-linha e os espaços em branco. Essas opções estão descritas em [Secção 4.10.2, “Opções de Fim-de-Linha e Espaços-Branco”](#). O comportamento por defeito é tratar todas as diferenças de espaços-em-branco e fim-de-linha como alterações reais, mas se queres ignorar uma alteração de indentação e encontrar o autor original, podes escolher uma opção apropriada aqui.

Uma vez que primas **OK**, o TortoiseSVN começa a readquirir os dados para criar o ficheiro de responsabilidade. Toma atenção: Isto pode levar vários minutos a concluir, dependendo de quanto o ficheiro mudou e, é claro, da ligação de rede ao repositório. Uma vez que o processo de responsabilidade tenha concluído, o resultado é escrito num ficheiro temporário onde poderás ver os resultados.

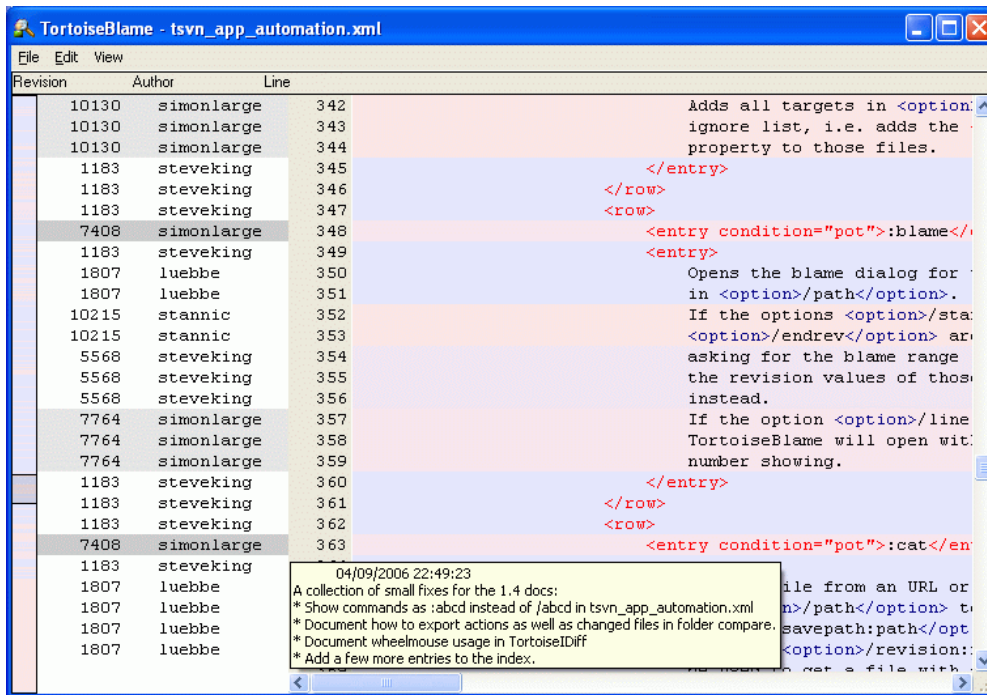


Figura 4.45. TortoiseBlame

O TortoiseBlame, que está incluído com o TortoiseSVN torna o ficheiro de responsabilidade mais fácil de ler. Quando passas com o rato sobre uma linha, na coluna de informação da responsabilidade, todas as linhas com a mesma revisão são mostradas com um fundo mais escuro. Linhas de outras revisões que foram alteradas pelo mesmo autor, são mostradas com um fundo mais claro. A coloração poderá não funcionar tão bem se o teu écran estiver configurado para o modo de 256 cores.

Se clicares com o botão direito numa linha, todas as linhas com a mesma revisão serão destacadas, e todas as linhas de outras revisões com o mesmo autor serão destacadas numa cor mais clara. Este destaque é persistente permitindo-te mover o rato sem perder os destaques. clica outra vez nessa revisão para remover o destaque.

Os comentários de revisão (mensagem de registo) são mostrados numa caixa de dica, quando passas com o rato sobre a coluna de informação de responsabilidade. Se queres copiar a mensagem de registo dessa revisão usa o menu de contexto, que aparece quando clicas com o botão direito na coluna de informação de responsabilidade.

Tu podes pesquisar no relatório de responsabilidade usando **Editar → Localizar...** Isto permite-te procurar por números de revisões, autores e o conteúdo das próprias mensagens. Mensagens de registo não são incluídas na pesquisa - tu deverás usar a caixa de diálogo de Registo para efectuar essa pesquisa.

Podes também saltar para uma número de linha específica usando, **Editar → Ir Para Linha...**

Quando o rato está sobre as colunas de informação de responsabilidade, um menu de contexto está disponível, que te ajudará a comparar revisões e examinar o histórico, usando o número de revisão da linha sobre o rato, como referência. **Menu de contexto → Responsabilizar pela revisão anterior** gera um relatório de responsabilidade para o mesmo ficheiro, mas usando a revisão anterior como o limite superior. Isto dá-te o relatório de responsabilidade para o estado do ficheiro, imediatamente antes da linha que estás a ver ter sido alterada. **Menu de contexto → Mostrar alteações** arranca o teu leitor de comparações, mostrando-te o que foi alterado na revisão referenciada. **Menu de contexto → Mostrar registo** mostra a caixa de diálogo de registo de revisões, a começar na revisão referenciada.

Se precisas um melhor indicador visual de onde estão as mais antigas e mais recentes alterações, selecciona **Ver → Colorir idade das linhas**. Isto irá usar um gradiente de cores para mostrar linhas mais

recentes a vermelho e linhas mais antigas em azul. A coloração por defeito é algo clara, mas podes alterá-lo usando as preferências do TortoiseBlame.

Se estás a usar o rastreamento de integração, onde as linhas foram alteradas como resultado de uma integração de outro caminho, o TortoiseBlame irá mostrar a revisão e autor da última alteração no ficheiro original em vez da revisão onde a integração foi efectuada. Essas linhas estão assinaladas ao mostrar o autor e revisão em itálico. Se queres que as linhas integradas não sejam mostradas desta maneira, não verifiques a caixa de diálogo Incluir informação de integração.

Se queres ver os caminhos envolvidos na integração, selecciona **Ver → Caminhos de integração**.

As preferências para o TortoiseBlame podem ser acedidas usando **TortoiseSVN → Preferências...** na aba TortoiseBlame. Consultar [Secção 4.30.9, “Preferências do TortoiseBlame”](#).

4.23.2. Diferenças de responsabilidade

Uma das limitações do relatório de responsabilidade é que só mostra como estava o ficheiro numa determinada revisão, e mostra a última pessoa que alterou cada linha. Por vezes tu queres conhecer que alteração foi feita, tal como quem a fez. O que tu precisas aqui é uma combinação de relatórios de comparação e responsabilidade.

A caixa de diálogo registo de revisão inclui várias opções que te permite fazer isto.

Responsabilizar Revisões

No painel de topo, selecciona 2 revisões e selecciona **Menu de contexto → Responsabilizar revisões**. Esta operação irá buscar os dados de responsabilidade para as 2 revisões, e usa o visualizador de comparações para comparar os dois ficheiros responsabilizados.

Responsabilizar alterações

Selecciona uma revisão no painel de topo e então pega num ficheiro no painel de fundo e selecciona **Menu de contexto → Responsabilizar alterações**. Esta operação irá buscar os dados de responsabilidade para a revisão seleccionada e para a revisão anterior, e usar o visualizador de comparação para comparar os dois ficheiros responsabilizados.

Comparar e Responsabilizar com a BASE de Trabalho

Mostra o registo para um só ficheiro, e no painel de topo, selecciona uma única revisão e selecciona **Menu de contexto → Comparar e Responsabilizar com a BASE de trabalho**. Esta operação irá buscar os dados de responsabilidade para a revisão seleccionada, e para o ficheiro na BASE de trabalho, e usa o visualizador de comparação para comparar os dois ficheiros responsabilizados.

4.24. O Navegador de Repositório

Por vezes necessitas de trabalhar directamente no repositório sem teres uma cópia de trabalho. É para isto que existe o *Navegador de Repositório*. Tal como o explorador e as sobreposições de ícones te permitem ver a tua cópia de trabalho, da mesma maneira o Navegador de Repositório permite-te ver a estrutura e o estado do repositório.

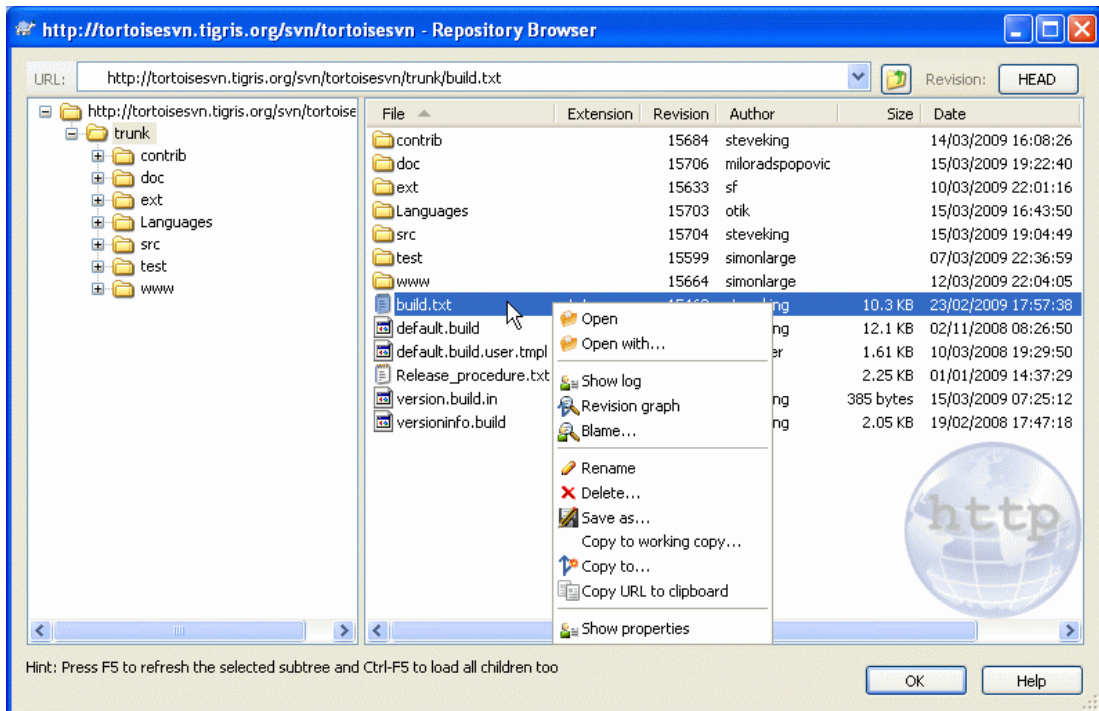


Figura 4.46. O Navegador de Repositório

Com o Navegador de Repositório podes executar comandos tais como, copiar, mover, renomear,... directamente no repositório.

O Navegador de Repositório tem um aspecto muito similar ao explorador do Windows, à excepção de que mostra os conteúdos do repositório numa revisão particular, em vez dos ficheiros no teu computador. No painel da direita podes ver a árvore de pastas e no painel da esquerda os conteúdos da pasta seleccionada. No topo da janela de navegação, do Navegador de Repositório, tu podes introduzir o URL e a revisão do repositório que queres navegar.

Tal como o explorador do Windows, tu podes clicar nos cabeçalhos de colunas do painel da direita, se queres seleccionar a disposição da ordenação. E também como no explorador existem menus de contexto disponíveis em ambos os painéis.

O menu de contexto para um ficheiro permite-te:

- Abrir o ficheiro seleccionado com o leitor por defeito para esse tipo de ficheiro ou, com um programa à tua escolha.
- Guarda uma cópia não versionada do ficheiro para o teu disco rígido.
- Mostrar o registo de revisão para esse ficheiro, ou mostra o gráfico de todas as revisões, para que possas ver de onde o ficheiro veio.
- Responsabiliza o ficheiro, para veres quem alterou que linha e quando.
- Remove ou renomea o ficheiro.
- Faz uma cópia do ficheiro, numa parte diferente do repositório, ou para uma cópia de trabalho originada do mesmo repositório.
- Ver/Editar as propriedades dos ficheiros.

O menu de contexto para uma pasta, permite-te:

- Mostrar o registo de revisão para esse ficheiro, ou mostra o gráfico de todas as revisões, para que possas ver de onde veio a pasta.
- Exportar a pasta, para uma cópia local não versionada no teu disco rígido.
- SVN Exportar uma pasta, para produzir uma cópia de trabalho local no teu disco rígido.
- Criar uma nova pasta no repositório.
- Adicionar ficheiros ou pastas, directamente no repositório.
- Remover ou renomear a pasta.
- Faz uma cópia da pasta, para uma diferente localização do repositório, ou para a cópia de trabalho enraizada no mesmo repositório.
- Ver/Editar as propriedades da pasta.
- Marcar a pasta para comparação. A pasta marcada é mostrada em negrito.
- Comparar a pasta com uma pasta previamente marcada, como uma comparação unificada, ou como uma lista de ficheiros alterados que podem então ser visualmente comparados, usando a ferramenta de comparação por defeito. Isto pode ser particularmente útil na comparação de duas etiquetas, ou o trunk e um ramo, para ver o que foi alterado.

Se seleccionares duas pastas no painel direito podes ver as suas diferenças como uma comparação unificada, ou como uma lista de ficheiros que podem ser visualmente comparados usando a ferramenta de comparação por defeito.

Se seleccionares múltiplas pastas no painel da direita, podes svn exportar todas de uma vez só para uma pasta pai comum.

Se seleccionares 2 etiquetas que foram copiadas a partir da mesma origem (tipicamente o /trunk/), podes usar **Menu de Contexto** → **Mostrar registo...** para ver a lista de revisões entre estes dois pontos de etiquetas.

Podes usar **F5** para refrescar a vista, como usual. Isto refrescará tudo o que é actualmente visualizado. Se queres efectuar uma pré-busca ou refrescar a informação de nós que não foram ainda abertos, usa **Ctrl-F5**. Depois disso a expansão de qualquer nodo será instantânea, sem atrasos de rede enquanto a informação é recebida.

Podes também usar o navegador de repositório para operações de arrastar-e-largar. Se arrastares uma pasta do explorador para o navegador-de-repositório, esta será importada para o repositório. De notar que, se arrastares múltiplos itens, estes serão importados em submissões separadas.

Se queres mover um item dentro do repositório, arrasta com o botão direito do rato para a nova localização. Se queres criar uma cópia, em vez de moveres o item, usa em alternativa **Ctrl**-arrastar com o botão direito. Ao copiar o cursor fica com o símbolo “mais” tal como no explorador.

Se queres copiar/mover um ficheiro ou pasta para outra localização, e também dar-lhe um novo nome ao mesmo tempo, podes então arrastar com o botão direito ou **Ctrl**-arrastar com o botão direito no item, em vez de usar arrastar com o botão esquerdo. Nesse caso, a caixa de diálogo renomear é mostrada, e podes então introduzir um novo nome para a pasta ou ficheiro.

Sempre que efectuares alterações no repositório usando um desses métodos, serás recebido com uma caixa de diálogo de mensagem de registo. Se arrastares algo por em engano, está é a tua oportunidade para cancelar a operação.

Por vezes quando tentas abrir um caminho és recebido com uma mensagem de erro, em vez dos detalhes do item. Isto pode acontecer de tu especificaste um URL inválido, se não tens permissões de acesso, ou se há algum problema no servidor. Se queres copiar esta mensagem, para a incluir num email, clica apenas

nela e usa o Menu de contexto → Copiar mensagem de erro para a área de transferência, ou usa simplesmente **Ctrl+C**.

4.25. Gráficos de Revisões

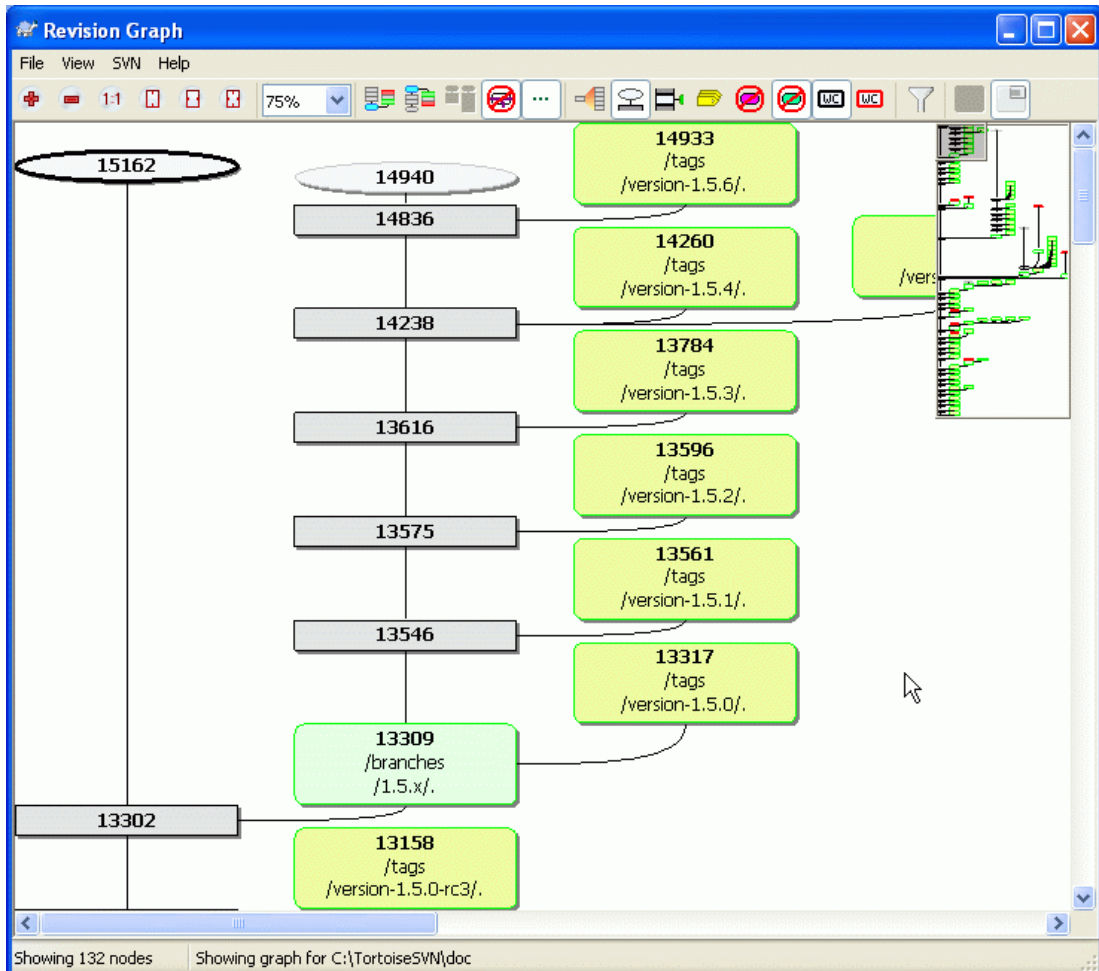


Figura 4.47. O Gráfico de Revisões

Necessitas, por vezes, de saber de onde os ramos e etiquetas foram retirados do trunk, e a forma ideal para visualizar este tipo de informação é um gráfico ou uma estrutura em árvore. É quando necessitas de usar TortoiseSVN → Gráfico de revisões...

Este comando analisa a história das revisões e tenta criar uma árvore que mostra os pontos em que as cópias foram feitas, e quando os ramos/etiquetas foram removidos.



Importante

De modo a gerar o gráfico, o TortoiseSVN precisa de carregar da raiz do repositório todas as mensagens de registo. Não será necessário dizer que isto irá levar vários minutos, mesmo com um repositório com alguns milhares de revisões, dependendo da rapidez do servidor, largura de banda da rede, etc. Se tentares isto com algo como o projecto *Apache*, que actualmente tem acima de 500.000 revisões, poderás ter de esperar por algum tempo.

As boas notícias é que estás a usar cache de registo, pelo que apenas terás de sofrer este atraso uma vez. Depois disso os dados de registo são armazenados localmente. O provisionamento de registo (cache) está activo nas preferências do TortoiseSVN.

4.25.1. Nós do Gráfico de Revisões

Cada nó do gráfico de revisões representa uma revisão no repositório onde algo foi alterado, na árvore que estás a observar. Diferentes tipos de nós podem ser distinguidos através da forma e cor. As formas são fixas, mas as cores podem ser alteradas usando TortoiseSVN → preferências

Itens adicionados ou copiados

Os itens que foram adicionados ou criados pela cópia de outro ficheiro/pasta, são mostrados como um rectângulo de cantos arredondados. A sua cor por defeito é o verde. Etiquetas e trunks são tratados como um caso especial, e usam uma forma diferente dependendo de TortoiseSVN → Preferências

Itens removidos

Itens removidos, e.g. um ramo que não é mais necessário, são mostrados como um octógono (rectângulo com os cantos cortados). A cor por defeito é o vermelho.

Itens renomeados

Itens renomeados são também mostrados como um octógono, mas a cor por defeito é o azul.

Revisão da extremidade do ramo

O gráfico está normalmente restringido a mostrar pontos de ramo, mas é também útil e comum, ter a possibilidade de ver a respectiva revisão HEAD para cada ramo. Se seleccionares **Mostrar revisões HEAD (mais recentes)**, cada nó de revisão HEAD será mostrado como uma elipse. De notar que a HEAD aqui refere-se à última revisão submetida nesse caminho e não a revisão HEAD do repositório

Revisão da cópia de trabalho

Se invocaste o gráfico de revisões a partir da cópia de trabalho podes optar por mostrar a revisão BASE no gráfico usando **Show WC revision**, que marca o nó BASE com uma moldura a negrito.

Cópia de trabalho modificada

Se invocaste o gráfico de revisões a partir de uma cópia de trabalho podes optar por mostrar um nó adicional, que representa a cópia de trabalho modificada usando **Mostrar modificações do WC**. Este será um nó elíptico com uma moldura a negrito, em vermelho por defeito.

Item normal

Todos os outros itens são mostrados como um rectângulo simples.

De notar que por defeito, o gráfico só mostra os pontos onde itens foram adicionados, copiados ou removidos. Mostrando cada revisão do projecto irá gerar um gráfico muito extenso para casos não triviais. Se queres realmente ver *todas* as revisões onde foram feitas alterações, existe uma opção para o fazer no menu **Ver** e na barra de ferramentas.

A vista por defeito (sem agrupamento) coloca os nós de modo que a sua posição vertical fique na ordem estrita de revisões, para que tenhas uma dica visual da ordem em que as coisas foram feitas. Quando estão na mesma coluna dois nós, a ordem é muito óbvia. Quando estão em colunas adjacentes dois nós, o deslocamento é muito menor porque não há necessidade de evitar que os nós se sobreponham, e como resultado a ordem é um pouco menos óbvia. Tais optimizações são necessárias de modo a manter a complexidade dos gráficos numa dimensão aceitável. Ter em atenção que esta ordenação usa o *canto* do nó no lado *mais antigo* como referência, i.e. o canto inferior do nó, quando o gráfico é mostrado com os nós mais antigos no fundo. O canto de referência é significativo porque todas as formas de nós não são da mesma altura.

4.25.2. Alterando a Vista

Porque o gráfico de revisões é normalmente bastante complexo, existem um número de funcionalidades que podem ser usadas para moldar a vista á tua maneira. Estas estão disponíveis no menu **Ver** e a partir da barra de ferramentas.

Agrupar ramos

O comportamento por defeito (sem agrupamento) é ter todas as linhas ordenadas estritamente por revisão. Como resultado, ramos com vida longa e submissões dispersas, ocupam uma coluna inteira para apenas poucas alterações tornando o gráfico muito amplo.

Este modo agrupa as alterações por ramo para que não exista uma ordenação global de revisões: Revisões consecutivas num ramo serão (normalmente) mostradas em linhas consecutivas. Sub-ramos no entanto, são arrumados de tal maneira que ramos posteriores serão mostrados na mesma coluna, acima dos ramos mais antigos, de modo a manter o gráfico esguio. Como resultado uma dada linha poderá conter alterações de diferentes revisões.

Mais antigo no topo

Normalmente o gráfico mostra a revisão mais antiga no fundo e a árvore cresce para cima. Usa esta opção para em alternativa crescer para baixo a partir do topo

Alinhar árvores no topo

Quando um gráfico está partido em várias árvores pequenas, as árvores podem aparecer na ordem natural de revisão ou alinhadas no fundo da janela, dependendo se estás a usar a opção **Agrupar ramos**. Usa esta opção para todas as árvores em alternativa crescerem a partir do topo.

Reduzir linhas cruzadas

Se a disposição do gráfico produziu muitas linhas cruzadas, usa esta opção para limpá-las. Isto poderá fazer com que as colunas de disposição apareçam em locais menos lógicos, por exemplo numa linha diagonal em vez de uma coluna, e o gráfico pode requerer mais área de desenho.

Nomes de caminhos diferenciais

Nomes de caminhos longos podem ocupar muito espaço e tornar muito largas as caixas dos nós. Usa esta opção para mostrar só a parte diferente de um caminho, substituindo a parte comum com pontos. E.g. se criares um ramo `/branches/1.2.x/doc/html` a partir de `/trunk/doc/html`, o ramo pode ser mostrado na forma compacta como `/branches/1.2.x/..` porque os dois últimos níveis `doc` e `html` não se alteraram.

Mostrar todas as revisões

Isto faz apenas aquilo que estás á espera, e mostra cada revisão em que algo (na árvore em que estás a construir o gráfico) foi alterado. Para históricos longos isto pode produzir um gráfico realmente gigante.

Mostrar revisões HEAD

Isto assegura que a última revisão de cada ramo é sempre mostrada no gráfico.

Fontes exactas de cópia

Quando um ramo/etiqueta é criado, o comportamento por defeito é mostrar o ramo como retirado do último nó onde foi efectuada uma alteração. Estritamente falando isto não é exacto, visto que os ramos são normalmente criados a partir da HEAD corrente em vez de uma revisão específica. É possível então mostrar a revisão mais correcta (mas menos útil) que foi usada para criar a cópia. De notar que esta revisão pode ser mais recente que a revisão HEAD do ramo fonte.

Etiquetas dobradas

Quando um projecto tem muitas etiquetas, mostrar cada etiqueta como um nó separado no gráfico ocupa muito espaço e ofusca a mais interessante estrutura do ramo de desenvolvimento. Ao mesmo tempo poderás necessitar de facilmente aceder ao conteúdo da etiqueta, para que possas efectuar comparações de revisões. Esta opção esconde os nós das etiquetas e em alternativa, mostra-os na etiqueta de dica do nó de onde foram copiadas. Um ícone de etiqueta no lado direito do nó fonte indica que foram criadas tags.

Esconder caminhos removidos

Esconde caminhos que já não estão presentes na revisão HEAD do repositório, e.g. ramos removidos.

Esconder ramos não modificados

Esconde ramos onde não foram submetidas alterações, no respectivo ficheiro ou subpasta. Isto não indica necessariamente que o ramo não foi usado, apenas que não foram feitas alterações *nesta* parte.

Mostra a revisão da CT

Marca no gráfico a revisão que corresponde à revisão actualizada do item, para o qual obtiveste o gráfico. Se acabaste de actualizar, esta será a revisão HEAD, mas se outros submeteram alterações

desde a tua última actualização, a tua CT poderá estar algumas revisões abaixo da HEAD. O nó é marcado com uma moldura a negrito.

Mostrar modificações da CT

Se a tua CT contém alterações locais, esta opção desenha-as como um nó elíptico separado e ligado ao nó a que a tua CT foi pela última vez actualizada. A cor de contorno por defeito é o vermelho. Para capturar as alterações mais recentes, poderás necessitar de refrescar o gráfico usando **F5**.

Filtro

Por vezes o gráfico de revisões contém mais revisões do que aquelas que queres ver. Esta opção abre a caixa de diálogo que permite restringir o intervalo de revisões mostradas e esconder caminhos particulares pelo nome.

Listas de árvore

Quando o gráfico contém várias árvores, é por vezes útil usar cores de fundo alternadas para ajudar à distinção entre elas.

Mostra visão geral

Mostra uma pequena figura do gráfico inteiro com a janela da vista corrente como um rectângulo que podes arrastar. Isto permite-te navegar pelo gráfico de forma mais fácil. De notar que para gráficos muito grandes a visão global pode-se tornar inutil, devido ao factor de ampliação (zoom) extremo, e por isso não ser mostrado nesses casos.

4.25.3. Utilizar o Gráfico

Usa a janela de vista global para tornar mais fácil a navegação em grandes gráficos. Este mostra o gráfico inteiro numa janela pequena, com a parcela correntemente mostrada salientada. Podes arrastar a área salientada de forma a alterar a região visualizada.

A data da revisão, o autor e os comentários são mostrados numa caixa de dica, quando o rato passa sobre a caixa de revisão.

Se seleccionares duas revisões (Usa **Ctrl**-clique com o botão direito), podes usar o menu de contexto para mostrar as diferenças entre estas revisões. Podes optar por mostrar as diferenças nos pontos de criação de ramos, mas usualmente queres mostrar as alterações nas pontas dos ramos, i.e. na revisão HEAD.

Podes ver as diferenças como um ficheiro de Comparação-Unificada que mostra todas as diferenças num único ficheiro com contexto mínimo. Se optares por **Menu de contexto** → **Comparar Revisões** será apresentado com uma lista de ficheiros alterados. Efectua duplo clique num nome de ficheiro de modo a obter ambas as revisões do ficheiro e compará-las recorrendo à ferramenta de comparação visual.

Se tu clicas com o botão direito numa revisão, podes usar **Menu de contexto** → **Mostrar registo** para ver o histórico.

Podes também integrar as alterações nas revisões seleccionadas, numa cópia de trabalho diferente. A caixa de diálogo selecção de pasta, permite-te escolher a cópia de trabalho para onde integrar, mas após esta não existe caixa de confirmação nem uma oportunidade para tentar uma integração de teste. Será uma boa ideia integrar numa cópia de trabalho não modificada, para que possas reverter as alterações se não resultar! Esta é uma funcionalidade útil se pretendes integrar as revisões seleccionadas, de um ramo para outro.



Aprender a Ler o Gráfico de Revisões

Utilizadores pela primeira vez, podem ser surpreendidos pelo facto de que, o gráfico de revisões mostra algo que não corresponde ao seu modelo mental. Se uma revisão altera múltiplas cópias ou ramos de um ficheiro ou pasta, por exemplo, então irão existir múltiplos nós para essa simples revisão. É uma boa prática começar com as opções mais à esquerda, na barra de ferramentas, e customizar o gráfico passo a passo até este chega perto do teu modelo mental.

Todas as opções de filtro tentam perder o mínimo de informação possível, isso poderá fazer com que alguns nós mudem de cor. Sempre que o resultado seja inesperado, desfaz a última operação de filtragem e, tenta compreender o que essa revisão ou ramo tem de especial. Na maior parte dos casos o resultado, inicialmente esperado da operação de filtragem, terá sido impreciso ou enganador.

4.25.4. Refrescar a Vista

Se quiseres consultar novamente o servidor por nova informação poderás simplesmente refrescar a vista usando o **F5**. Se estiveres a usar a cache de registo (activada por defeito) o repositório será consultado para verificar novas submissões e buscará somente as novas. Se a cache de registo está em modo fora-de-linha, esta operação irá também tentar retornar ao modo em-linha.

Se estás a usar a cache de registo, e desconfias que o conteúdo da mensagem ou o autor foram alterados, então deverás recorrer à caixa de diálogo de registo para refrescar as mensagens de que necessitas. Desde que o gráfico de revisões funciona a partir da raiz do repositório, teríamos de invalidar toda a cache de registo, e uma recarga poderia levar um tempo *muito* longo.

4.25.5. Podar as Árvores

Uma árvore grande pode-se tornar difícil de navegar e por vezes tu quererás esconder algumas partes, ou parti-la numa floresta de árvores mais pequenas. Se manteres o rato sobre o ponto onde um nó de ligação entra ou sai do nó, irás ver um ou mais botões de aparição que te permitiram fazer isso.



Clica no botão de subtração para colapsar a sub-árvore anexada.



Clica no botão de adição para expandir a árvore colapsada. Quando uma árvore está colapsada este botão mantém-se visível para sinalizar a sub-árvore escondida.



Clica no botão da cruz para dividir a sub-árvore anexada e mostrá-la no gráfico como uma árvore separada.



Clica no botão com o círculo para re-anexar a árvore dividida. Quando uma árvore foi dividida este botão permanece visível para indicar que existe uma sub-árvore separada.

Clica no fundo do gráfico para obteres o menu de contexto que te oferece opções para **Expandir todos** e **Juntar todos**. Se não existem nenhum ramo colapsado ou dividido, o menu de contexto não será mostrado.

4.26. Exportar uma Cópia de Trabalho do Subversion

Por vezes irás querer uma cópia da tua árvore de trabalho, mas sem as pastas `.svn`, e.g. de modo a criar um tarball zipado da tua fonte, ou para exportar para um servidor web. Em vez de efectuares uma cópia para depois apagar manualmente todas as pastas `.svn` podes usar o comando do Subversion `TortoiseSVN → Exportar...`. Exportar de um URL e exportar de uma cópia de trabalho são tratadas de forma ligeiramente diferente.

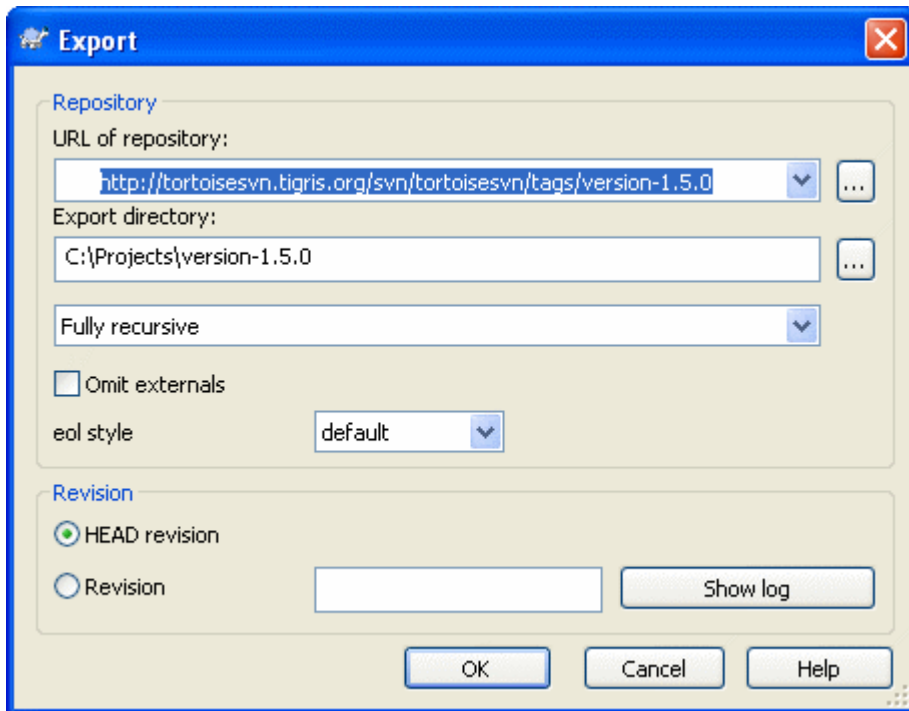


Figura 4.48. A Caixa de Diálogo Exportar-do-URL

Se executas este comando numa pasta não versionada, o TortoiseSVN irá assumir que essa pasta é o destino e abre uma caixa de diálogo para introduzires o URL e a revisão a exportar. Esta caixa de diálogo tem opções para, só exportar a pasta de topo, omitir referências externas e sobrepor o estilo de fim de linha para ficheiros que têm configurada a propriedade `svn:eol-style`.

É claro que podes também exportar directamente do repositório. Usa o Navegador de Repositório para navegares no repositório até à sub-árvore pretendida e usa **Menu de Contexto** → **Exportar**. Irás obter a caixa de diálogo **Exportar do URL** descrita acima.

Se executares este comando na tua cópia de trabalho serás questionado por uma localização para guardares a cópia de trabalho *limpa*, sem as pastas `.svn`. Por defeito só os ficheiros versionados serão exportados, mas podes usar a caixa de verificação **Exportar também ficheiros não versionados**, para incluir quaisquer outros ficheiros não versionados que possam existir na tua cópia de trabalho. Referências externas, usando `svn:externals` poderão, se necessário, ser omissas.

Outra forma de exportar a partir de uma cópia de trabalho é arrastar com o botão direito a cópia de trabalho para outro local e seleccionar **Menu de Contexto** → **SVN Exportar aqui** ou **Context Menu** → **SVN Exporta todos aqui**. A segunda opção inclui também os ficheiros não versionados.

Quando ao exportar de uma cópia de trabalho, se a pasta de destino já conter uma pasta com o mesmo nome daquela que está a exportar, será te dada a opção para escrever por cima do conteúdo já existente ou criares uma nova pasta com um nome gerado automaticamente, e.g. `Target (1)`.



Exportar ficheiros individuais

A caixa de diálogo exportar não te permite exportar ficheiros individuais, mesmo que o Subversion permita.

Para exportares ficheiros simples com o TortoiseSVN terás de usar o Navegador de repositório ([Secção 4.24, “O Navegador de Repositório”](#)). Simplesmente arrasta o(s) ficheiro(s) que queres exportar do repositório para o local, no explorador, onde os queres, ou usa o menu de contexto no repositório para os exportar.



Exportar uma árvore de alterações

Se queres exportar uma cópia da estrutura de árvore do teu projecto, mas contendo só os ficheiros que foram alterados numa revisão particular, ou entre duas quaisquer revisões, usa a funcionalidade comparar revisões, descrita em [Secção 4.10.3, “Comparando Pastas”](#).

4.26.1. Remover uma cópia de trabalho do controlo de versões

Por vezes tu tens uma cópia de trabalho que desejas converter de volta para uma pasta normal, sem as pastas `.svn`. O que tu realmente precisas é um comando exportar-no-local, que remove apenas as pastas de controlo, em vez de gerar uma nova árvore de pastas limpa.

A resposta é surpreendente simples - exportar a própria pasta! O TortoiseSVN detecta este caso especial e pergunta se tu queres tornar a cópia de trabalho não versionada. Se responderes *sim*, as pastas de controlo serão removidas, e terás uma simples árvore de pastas limpa.

4.27. Reposicionar uma cópia de trabalho

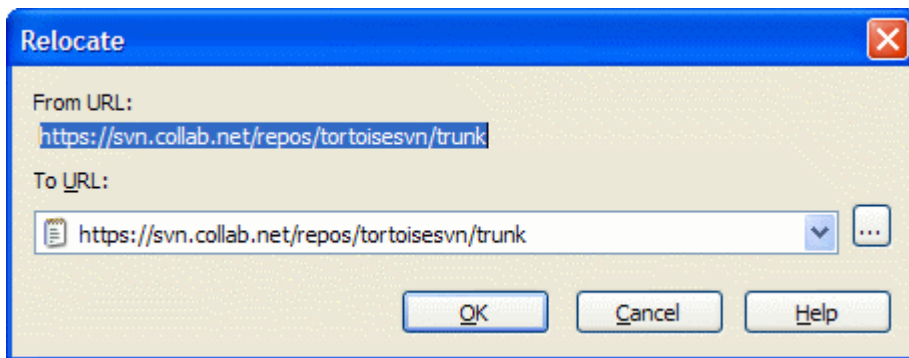


Figura 4.49. A Caixa de Diálogo Reposicionar

Se o teu repositório por alguma razão mudou a sua localização (IP/URL). Talvez estejas emperrado e não consigas submeter e não queres SVN exporta outra vez a tua cópia de trabalho a partir da nova localização, e mover todas as tuas alterações para a tua nova cópia de trabalho, o comando TortoiseSVN → Reposicionar é o que procuras. Basicamente não faz muito: pesquisa todas as entradas de ficheiros na pasta `.svn` e altera o URL das entradas para o novo valor.

Poderás ficar surpreendido por descobrires que o TortoiseSVN contacta o repositório como parte desta operação. Tudo o que faz é executar algumas simples verificações para ter a certeza que o novo URL refere-se realmente ao mesmo repositório da cópia de trabalho existente.



Atenção

Esta é uma operação muito raramente usada. O comando reposicionar é só usado se o URL da raiz do repositório foi alterado. Possíveis razões:

- O endereço IP do servidor foi alterado.
- O protocolo foi alterado (e.g. `http://` para `https://`).
- O caminho raiz de repositório na configuração do servidor foi alterado

Posto de outra maneira, necessitas de reposicionar quando a tua cópia de trabalho se refere à mesma localização no mesmo repositório, mas o próprio repositório foi deslocado.

Não se aplica se:

- Se queres mover para um repositório do Subversion diferente. Nesse caso deverás executar um SVN exportar limpo a partir da localização do novo repositório.
- Se queres trocar para um ramo diferente ou pasta dentro do mesmo repositório. Para o fazeres deverás usar TortoiseSVN → Trocar.... Consulta [Secção 4.19.2, “SVN Exportar ou Trocar...”](#) para mais informação.

Se reposicionares em qualquer dos casos acima citados, irás *corromper a tua cópia de trabalho*, e irás obter muitas mensagens de erro inexplicáveis enquanto actualizares, submeteres, etc. Uma vez que isso acontecer a única reparação a fazer é efectuar um SVN exportar de fresco.

4.28. Integração com Sistemas de controlo de problemas/ Gestores de Problemas

É muito comum no desenvolvimento de Software que as alterações possam ser relacionadas com um problema ou bug específico. Os utilizadores dos sistemas de identificação de bugs (gestores de problemas) gostariam de associar a alterações que fazem no Subversion, com um ID específico, no seu sistema de gestão de problemas. A maioria dos gestores de problemas providencia um script de gancho pre-commit que analisa a mensagem de registo de modo a encontrar o ID do bug, com o qual a submissão está associada. Isto é algo sujeito a erros, visto que confia que o utilizador escreva uma mensagem de registo correcta, para que o script de gancho pre-commit o possa analisar correctamente.

O TortoiseSVN pode ajudar o utilizador de duas maneiras:

1. Quando o utilizador introduz a mensagem de registo, pode ser automaticamente adicionada uma linha bem definida, que inclui o número de problema associado com a submissão. Isto irá reduzir o risco de que o utilizador introduza um número de problema de uma maneira tal que, as ferramentas de identificação de bugs não o consigam analisar correctamente.

Ou o TortoiseSVN poderá salientar a parte da mensagem inserida, que é reconhecida pelo gestor de problemas. Dessa maneira o utilizador saberá que a mensagem de erro pode ser correctamente analisada.

2. Quando o utilizador navega nas mensagens de registo, o TortoiseSVN cria uma hiperligação a partir de cada ID de bug na mensagem de erro, que aponta o navegador de internet para o problema mencionado.

4.28.1. Adicionar Números de Problemas nas Mensagens de Registo

Tu podes integrar a ferramenta de controlo de problemas da tua escolha com o TortoiseSVN. Para o fazer, terás de definir algumas propriedades que começam por `bugtraq:`. Estas deverão ser colocadas nas pastas: ([Secção 4.17, “Configurações de Projecto”](#))

Existem duas maneiras de integrar o TortoiseSVN com os gestores de problemas. Um é baseado em cadeias de caracteres simples, a outra é baseado em *expressões regulares*. As propriedades usadas em ambas as abordagens são:

`bugtraq:url`

Configura esta propriedade com o URL da tua ferramenta de controlo de problemas. Deverá ser um URI propriamente codificado e terá de conter `%BUGID%`. O `%BUGID%` é substituído pelo número de problema que introduzires. Isto permite ao TortoiseSVN mostrar uma hiperligação na caixa de diálogo de registo, para que quando estiveres a olhar para o registo de revisões, possas saltar directamente para a tua ferramenta de identificação de bugs. Tu não precisas de providenciar essa propriedade, mas então o TortoiseSVN apenas mostrará o número

do problema, e não a hiperligação para este. E.g. o projecto TortoiseSVN usa `http://issues.tortoisesvn.net/?do=details&id=%BUGID%`

Poderás também usar URLs relativos em vez de absolutos. Isto é útil quando o teu gestor de problemas está no mesmo domínio/servidor que o teu repositório de fontes. No caso do nome de domínio se alterar, tu não precisarás de ajustar a propriedade `bugtraq:url`. Existem então duas maneiras de especificar um URL relativo:

Se começa com a cadeia de caracteres `^/` é assumido que é relativo à raiz do repositório. Por exemplo, `^/..?do=details&id=%BUGID%` irá ser resolvido para `http://tortoisesvn.net/?do=details&id=%BUGID%` se o teu repositório está localizado em `http://tortoisesvn.net/svn/trunk/`.

Um URL que comece pela cadeia de caracteres `/` é assumido como sendo relativo ao nome do servidor. Por exemplo `/?do=details&id=%BUGID%` será resolvido para `http://tortoisesvn.net/?do=details&id=%BUGID%` se o teu repositório está localizado em qualquer ponto do domínio `http://tortoisesvn.net`.

`bugtraq:warnifnoissue`

Fixa este a `true` se queres que o TortoiseSVN te avise, que o campo de texto do número-de-problema está vazio. Valores válidos são `true/false`. *Se não definido é assumido o valor `false`.*

4.28.1.1. Número de Problema numa Caixa de Texto

Na abordagem mais simples o TortoiseSVN mostra ao utilizador um campo de entrada separado onde o ID do bug pode ser introduzido. Então uma linha separada é pre/pós anexada à mensagem de registo que o utilizador introduziu.

`bugtraq:message`

Esta propriedade activa o sistema de controlo de problemas no modo *Campo de Entrada*. Se esta propriedade é fixada então o TortoiseSVN ir-te-á pedir para introduzires o número de problema quando submeteres as tuas alterações. Será usado para adicionar uma linha no fim da mensagem de registo. Deverá conter `%BUGID%`, que será substituído pelo número de problema que introduzires ao submeter. Isto garante que teu registo de submissão contém uma referência ao número de problema, que estará sempre num formato consistente e poderá ser analisado pela tua ferramenta de identificação de problemas, de forma a associar o número de problema com uma submissão particular. Como exemplo poderás usar `Issue : %BUGID%`, mas tal dependerá da tua ferramenta.

`bugtraq:append`

Esta propriedade define se o bug-ID é anexado (`true`) ao fim da mensagem de registo ou inserido (`false`) no início da mensagem de registo. Valores válidos são `true/false`. *Se não definido, é assumido o valor `true` para que os projectos já existentes não sejam danificados.*

`bugtraq:label`

Este texto é mostrado pelo TortoiseSVN na caixa de diálogo submeter, para etiquetar a caixa de edição onde irás introduzir o número do problema. Se não fixado, será mostrado `Bug-ID / Issue-Nr:`. Tem em mente que a janela não será redimensionada para acomodar esta etiqueta, pelo que, mantém o tamanho da etiqueta abaixo dos 20-25 caracteres.

`bugtraq:number`

Se configurado a `true` só serão permitidos números no campo de texto do número-de-problema. Uma excepção é a vírgula para que possas separar vários números. Valores válidos são `true/false`. *Se não definido, é assumido `true`.*

4.28.1.2. Números de Problema Recorrendo a Expressões Regulares

Na abordagem com as *expressões regulares*, o TortoiseSVN não mostra um campo de entrada separado, mas marca a parte da mensagem de registo que o utilizador introduz, e que é reconhecida pelo controlador de problemas. Isto é efectuado enquanto o utilizador escreve a mensagem de registo. Também significa

que o ID do problema pode estar algures dentro da mensagem de registo! Este método é muito mais flexível e é o utilizado pelo próprio projecto do TortoiseSVN.

bugtraq:logregex

Esta propriedade activa o sistema de controlo de bugs no modo *Regex*. Ele contém uma única expressão regular ou duas expressões regulares separadas por uma nova linha.

Se são usadas duas expressões, então a primeira é usada como um pré-filtro para encontrar as expressões que contêm os IDs dos problemas. A segunda expressão extrai então os IDs dos bugs encontrados, a partir do resultado da primeira regex. Isto permite-te usar uma lista de IDs de problemas e as expressões, em linguagem natural, que desejes. e.g. poderás resolver vários problemas e incluir uma string com esta: “Esta alteração resolve os assuntos #23, #24 e #25”

Se queres apanhar os IDs dos problemas, como é usado na expressão acima no interior da mensagem de registo, podes usar as seguintes strings regex, que são as usadas pelo projecto TortoiseSVN:
`[Ii]ssues?:?(\\s*(,|and)?\\s*#\\d+)+ and (\\d+)`

A primeira expressão pega nos “assuntos #23, #24 and #25” a partir da mensagem de registo circundante. A segunda regex extrai simples números decimais a partir da saída da primeira regex, pelo que irá retornar “23”, “24” e “25” para serem usados como IDs de problemas.

Dividindo um pouco a primeira regex, esta deve começar com a palavra “issue”, e possivelmente em maiúsculas. Esta é opcionalmente seguida por um “s” (mais de um assunto) e também opcionalmente por dois pontos. Isto é seguido por um mais grupos, cada um tendo à frente zero ou mais espaços em branco, e um opcionalmente uma vírgula ou “and” e mais espaços opcionais. Finalmente existe um “#” obrigatório e um também obrigatório número decimal.

Se só for usada uma expressão, então os IDs dos problemas descobertos terão de ter correspondência nos grupos da string regex. Exemplo: `[Ii]ssue(?:s)? #?(\\d+)` Este método é requerido por alguns controladores de assuntos, e.g. trac, mas é mais difícil a construção da regex. Nós recomendamos que só uses este método se a documentação do teu controlador de assuntos te disser para o fazer.

Se não estás familiarizado com as expressões regulares, dá uma vista de olhos na introdução em http://en.wikipedia.org/wiki/Regular_expression, e na documentação em linha e tutorial em <http://www.regular-expressions.info/>.

Se ambas as propriedades bugtraq:message e bugtraq:logregex estão configuradas, toma precedência a logregex



Dica

Mesmo que não tenhas um controlador de assuntos, com um gancho de pré-submissão a analisar a tua mensagem de registo, poderás ainda usar isto para converter os assuntos mencionados nas tuas mensagens de registo em hiperligações!

E mesmo que não precisas das hiperligações, os números de problemas aparecerão como uma coluna separada na caixa de diálogo registo, tornando mais fácil encontrar as alterações que estão relacionadas com um assunto em particular.

Algumas propriedades `tsvn:` requerem um valor `true/false`. O TortoiseSVN também entende `yes` como sinónimo para `true` e `no` como sinónimo para `false`.



Configurar Propriedades em Pastas

Essas propriedades devem ser configuradas em pastas para o sistema funcionar. Quando submetes um ficheiro ou pasta, as propriedades são lidas dessa pasta. Se as propriedades não

são aí encontradas, o TortoiseSVN irá pesquisar através, e em direcção do topo, da árvore de pastas para as encontrar, até encontrar uma pasta não versionada, ou a raiz da árvore (eg. C:\). Se tens a certeza que cada utilizador só checou-a, a partir do e.g. trunk/ e não de alguma subpasta, é suficiente configurar as propriedades na pasta trunk/. Se não tens a certeza deverás configurar as propriedades em cada subpasta, recursivamente. Configurar uma propriedade na profundidade da hierarquia do projecto sobrepõem-se à configuração em níveis superiores (mais perto do trunk/).

Podes usar a caixa de verificação **Recursivo**, apenas para as propriedades tsvn:, para fixar a propriedade a todas as sub-pastas na hierarquia, sem a fixares também em todos os ficheiros.



Não existe informação do Controlador de Assuntos no Navegador de Repositório

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.

A integração com o controlador de assuntos não está restricta ao TortoiseSVN; pode também ser usada por qualquer cliente Subversion. Para mais informações, consulta o *Issue Tracker Integration Specification* [<http://tortoisesvn.googlecode.com/svn/trunk/doc/issuetrackers.txt>] completo no repositório fonte do TortoiseSVN. ([Secção 3, “O TortoiseSVN é grátis!”](#) explica-te como aceder ao repositório).

4.28.2. Obter Informações do Gestor de Problemas

A secção anterior ocupa-se com a adição da informação do problema nas mensagens de registo. Mas se tu precisares de obter informações do teu gestor de problemas? A caixa de diálogo submeter tem um interface COM que permite a integração com um programa externo, que consegue comunicar com o teu gestor. Tipicamente tu poderás querer consultar o gestor para obter a lista de problemas abertos atribuídos a ti, de modo a que possas seleccionar os problemas que estarão a ser tratados nesta submissão.

Tal interface, qualquer que seja, é claro altamente específico ao teu sistema de gestão de problemas pelo que não poderemos fornecer esta parte, e descrever como criar tal programa está fora do âmbito deste manual. A definição de interface e amostras de plugins em C# e C++/ATL podem ser obtidas na pasta contrib em [repositório do TortoiseSVN](#) [<http://tortoisesvn.tigris.org/svn/tortoisesvn/trunk/contrib/issue-tracker-plugins>]. ([Secção 3, “O TortoiseSVN é grátis!”](#) explica-te como aceder ao repositório). Um resumo da API é também fornecido em [Capítulo 6, Interface IBugtraqProvider](#) Outro exemplo (funcional) de plugin em C# é o *Gurtle* [<http://code.google.com/p/gurtle/>] que implementa o interface COM necessário para interagir com o gestor de assuntos do *Google Code* [<http://code.google.com/hosting/>].

Para efeitos de demonstração, vamos supor que o teu administrador de sistemas providenciou-te um plugin de gestor de problemas que tu instalaste, e entretanto configuraste algumas das tuas cópias de trabalho para usar o plugin, na caixa de diálogo de preferências do TortoiseSVN. Quando abres a caixa de diálogo Submeter a partir de uma cópia de trabalho para a qual o plugin foi atribuído, verás um novo botão no topo da caixa de diálogo.

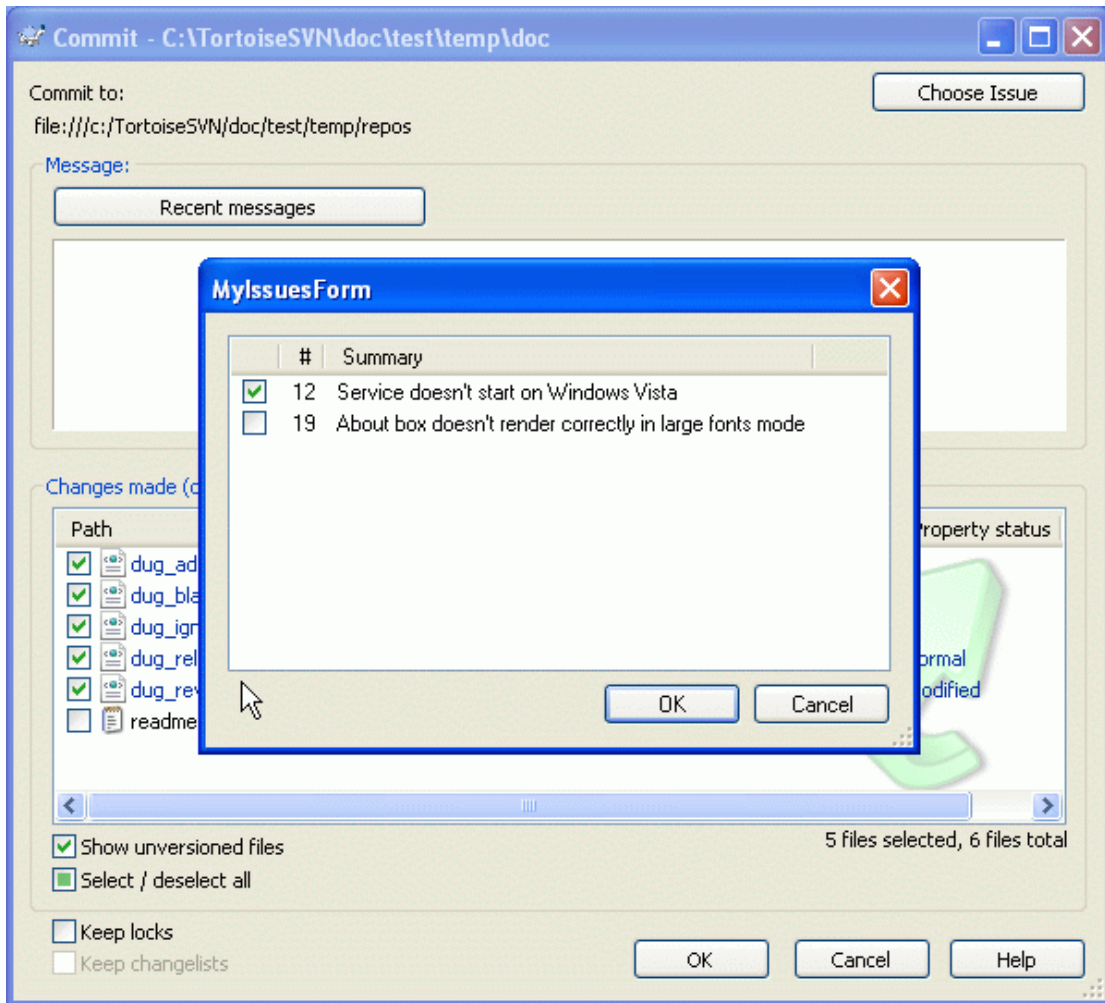


Figura 4.50. Caixa de diálogo de exemplo da consulta ao gestor de problemas

Neste exemplo tu podes seleccionar um ou mais problemas abertos. O plugin pode então gerar um texto, especialmente formatado, que adicionará à tua mensagem de registo.

4.29. Integração com visualizadores de repositório Web-based

Existem disponíveis vários visualizadores de repositório web-based para o uso com o Subversion, tais como o [ViewVC](http://www.viewvc.org/) [http://www.viewvc.org/] e o [WebSVN](http://websvn.tigris.org/) [http://websvn.tigris.org/]. O TortoiseSVN providência um meio de ligar-se com esses visualizadores.

Tu podes integrar um visualizador de repositório, da tua escolha, com o TortoiseSVN. Para o fazeres terás de configurar algumas propriedades que definem a ligação. Estas têm de ser configuradas nas pastas: (Secção 4.17, “Configurações de Projecto”)

webviewer:revision

Configura esta propriedade com o URL do teu visualizador de repositório, para veres todas as alterações numa revisão específica. O URL deverá ser devidamente codificado e terá de conter %REVISION%. O %REVISION% será substituído com o número de revisão em questão. Isto permite ao TortoiseSVN mostrar uma entrada do menu de contexto na caixa de diálogo registo Menu de Contexto → Ver revisão no visualizador web

webviewer:pathrevision

Set this property to the URL of your repo viewer to view changes to a specific file in a specific revision. It must be properly URI encoded and it has to contain %REVISION% and %PATH%. %PATH

% is replaced with the path relative to the repository root. This allows TortoiseSVN to display a context menu entry in the log dialog **Context Menu** → **View revision and path in webviewer**. For example, if you right-click in the log dialog bottom pane on a file entry `/trunk/src/file` then the `%PATH%` in the URL will be replaced with `/trunk/src/file`.

Podes também usar URLs relativos em vez de absolutos. Isto é útil no caso do teu visualizador web estar no mesmo domínio/servidor que o teu repositório fonte. No caso de o nome de domínio alguma vez for alterado, não terás de ajustar as propriedades `webviewer:revision` e `webviewer:pathrevision`. O formato é o mesmo para a propriedade `bugtraq:url`. Consultar [Secção 4.28, “Integração com Sistemas de controlo de problemas/Gestores de Problemas”](#).



Configurar Propriedades em Pastas

Essas propriedades devem ser configuradas em pastas para o sistema funcionar. Quando submetes um ficheiro ou pasta, as propriedades são lidas dessa pasta. Se as propriedades não são aí encontradas, o TortoiseSVN irá pesquisar através, e em direcção do topo, da árvore de pastas para as encontrar, até encontrar uma pasta não versionada, ou a raiz da árvore (eg. `C:\`). Se tens a certeza que cada utilizador só checou-a, a partir do e.g. `trunk/` e não de alguma subpasta, é suficiente configurar as propriedades na pasta `trunk/`. Se não tens a certeza deverás configurar as propriedades em cada subpasta, recursivamente. Configurar uma propriedade na profundidade da hierarquia do projecto sobrepõem-se à configuração em níveis superiores (mais perto do `trunk/`).

Podes usar a caixa de verificação **Recursivo**, *apenas* para as propriedades `tsvn:`, para fixar a propriedade a todas as sub-pastas na hierarquia, sem a fixares também em todos os ficheiros.



Sem links do Visualizador do Repo a partir do Navegador de Repositório

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. Preferências do TortoiseSVN

Para descobrires para que servem as diferentes preferências deixa, por um segundo, o ponteiro do rato sobre as caixas-edição/caixas-verificação... e uma etiqueta com uma útil dica, irá aparecer.

4.30.1. Preferências Gerais

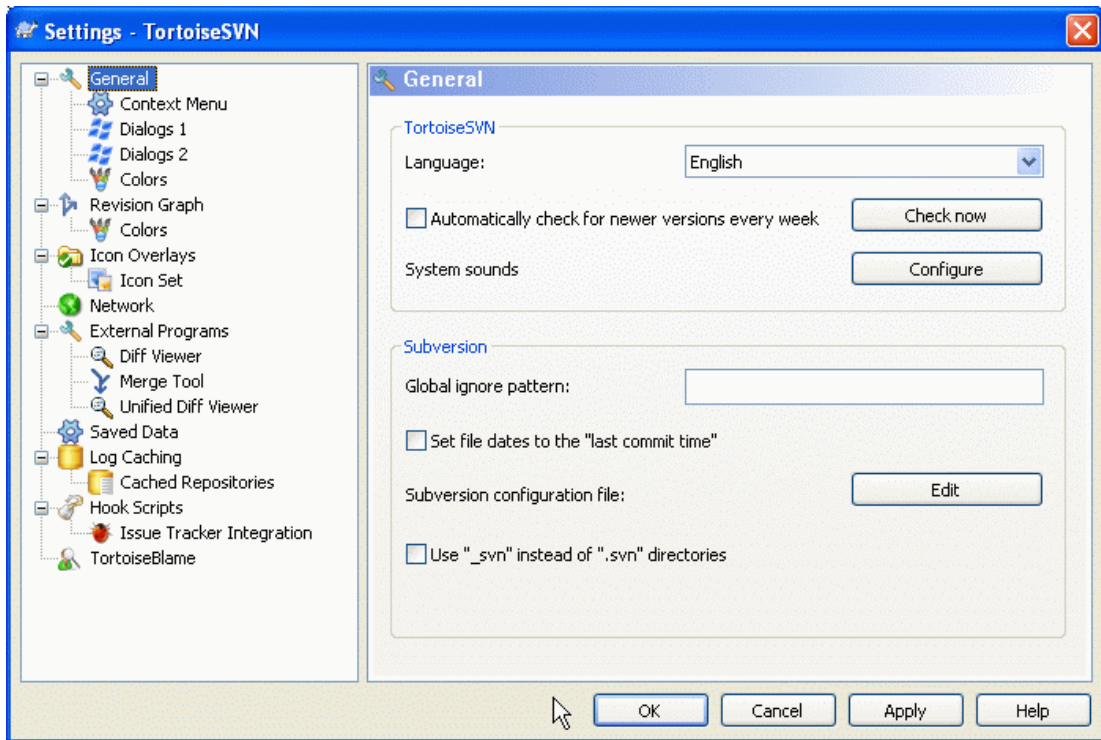


Figura 4.51. A Caixa de Diálogo Preferências, Página Geral

Esta caixa de diálogo permite-te especificar a tua língua de preferência, e preferências específicas do Subversion.

Língua

Selecciona a língua do teu interface de utilizador. O que é que esperavas?

Verifica automaticamente, cada semana, a existência de novas versões

Se verificada, o TortoiseSVN irá contactar, uma vez por semana, o seu sítio de descarga para ver se existe disponível uma versão mais nova do programa. Usa **Verificar agora** se queres uma resposta imediata. A nova versão, se existir, não será logo descarregada; receberás simplesmente uma caixa de diálogo de informação, que te dirá que, tens disponível uma nova versão.

Sons do sistema

O TortoiseSVN tem três sons personalizados que estão instalados por defeito.

- Erro
- Anúncio
- Aviso

Podes seleccionar diferentes sons (ou desligá-los completamente) usando o Painel de Controlo do Windows. **Configurar** é uma atalho para o Painel de Controlo.

Padrão global para ignorar arquivos

Padrões globais para ignorar são usados para evitar que ficheiros não versionados apareçam, por exemplo, na caixa de diálogo submeter. Os ficheiros que correspondam ao padrão são também ignorados ao importar. Ignora ficheiros ou pastas apenas ao digitares o seu nome ou extensão. Os padrões são separados por espaços e.g. `bin obj *.bak *.~?? *.jar *.[Tt]mp`. Esses padrões não devem incluir qualquer separador de caminho. De salientar que, também não existe

maneira de diferenciar pastas de ficheiros. Consultar [Secção 4.13.1, “Correspondência de Padrões em Listas de Ignorados”](#) para mais informação sobre a sintaxe de correspondência-de-padrões.

Nota que os padrões de ignorar que aqui especificas também afectaram outros clientes do Subversion, que correres no teu PC, incluindo o cliente de linha de comando.



Cuidado

Se usares o ficheiro de configuração do Subversion para configurares o padrão `global-ignores`, ele ir-se-á sobrepor à preferência que efectuaste aqui. O ficheiros de configuração do Subversion pode ser acedido usando o **Editar**, como aqui descrito.

Este padrão de ignorar irá afectar todos os teus projectos. Não é versionado pelo que não afectará outros utilizadores. Por contraste podes também usar a propriedade versionada `svn:ignore` para excluir ficheiros ou pastas do controlo de versões. Consulta [Secção 4.13, “Ignorando Ficheiros E Pastas”](#) para mais informações.

Alterar datas de ficheiros para a “hora da última submissão”

Esta opção diz ao TortoiseSVN para alterar a data dos ficheiros para a hora da última submissão, quando efectuares um checkout ou actualização. De outra maneira o TortoiseSVN irá usar a data corrente. Se fores um programador é geralmente melhor usar a data corrente, porque os sistemas de construção olham normalmente para os carimbos de tempo, para decidir que ficheiros precisam de ser compilados. Se usares a “hora da última submissão”, e reverteres para uma revisão anterior de ficheiro, o teu projecto poderá não compilar como esperado.

Ficheiro de configuração do Subversion

Usa **Editar** para editar directamente o ficheiro de configuração do Subversion. Algumas preferências não podem ser editadas directamente pelo TortoiseSVN e necessitam então de ser aqui configuradas. Para mais informações sobre o ficheiro `config` do Subversion consulta o [Runtime Configuration Area](#) [<http://svnbook.red-bean.com/en/1.5/svn.advanced.confarea.html>], é de interesse particular a secção no [Automatic Property Setting](#) [<http://svnbook.red-bean.com/en/1.5/svn.advanced.props.html#svn.advanced.props.auto>], e é configurada aqui. De notar que o Subversion pode ler a informação d configuração de vários locais, e necessitas de conhecer qual deles tem prioridade. Consulta o [Configuration and the Windows Registry](#) [<http://svnbook.red-bean.com/en/1.5/svn.advanced.confarea.html#svn.advanced.confarea.windows-registry>] para mais informações.

Usa pastas `_svn` em vez de `.svn`

O VS.NET quando usado com projectos web não consegue lidar com as pastas `.svn` que o Subversion usa para armazenar a sua informação interna. Isto não é um bug no Subversion. O bug é no VS.NET e nas extensões frontpage que usa. Consulta [Secção 4.30.11, “Pastas de Trabalho do Subversion”](#) para descobrires mais sobre este problema.

Se quiseres alterar o comportamento do Subversion e do TortoiseSVN, podes usar esta caixa-de-verificação para configurar a variavel de ambiente que controla isto.

Deves ter em conta que ao mudar esta opção, não irás converter automaticamente as cópias de trabalho existentes, para usar a nova pasta de administração. Terás de o fazer por ti usando um script (Lê as nossas FAQ) ou simplesmente cria uma cópia de trabalho de fresco.

4.30.1.1. Menu de Contexto preferências

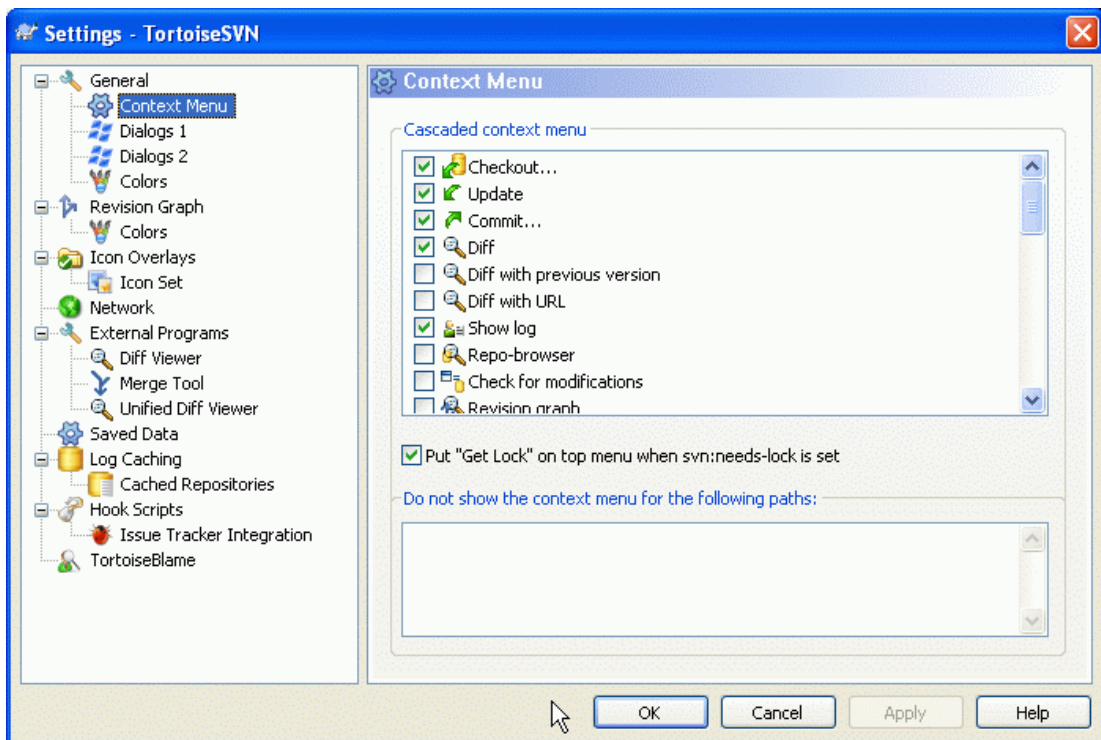


Figura 4.52. A Caixa de Diálogo Preferências, Página Menu de Contexto

Esta página permite-te especificar, quais das entradas do menu de contexto do TortoiseSVN, que irão aparecer no menu de contexto principal, e quais aparecerão no submenu do TortoiseSVN. Por defeito a maioria dos itens estão desseleccionados e aparecem no submenu.

Há um caso especial para o Obter Bloqueio. É claro que podes promovê-lo ao nível de topo, usando a lista acima mostrada, mas como a maioria dos ficheiros não precisam de bloqueios, isto só adiciona mais confusão. No entanto um ficheiro que tenha a propriedade `svn:needs-lock` necessitará sempre desta acção de cada vez que é editado, então nesse caso será muito útil tê-lo no nível de topo. Verificando a caixa aqui, significa que quando o ficheiro for seleccionado e tenha a propriedade `svn:needs-lock` configurada, o Obter Bloqueio irá aparecer sempre no nível de topo.

Se existem alguns caminhos no teu computador onde não queres que o menu de contexto do TortoiseSVN apareça de modo algum, podes listá-los na caixa de texto ao fundo.

4.30.1.2. Preferências das Caixas de Diálogo TortoiseSVN 1

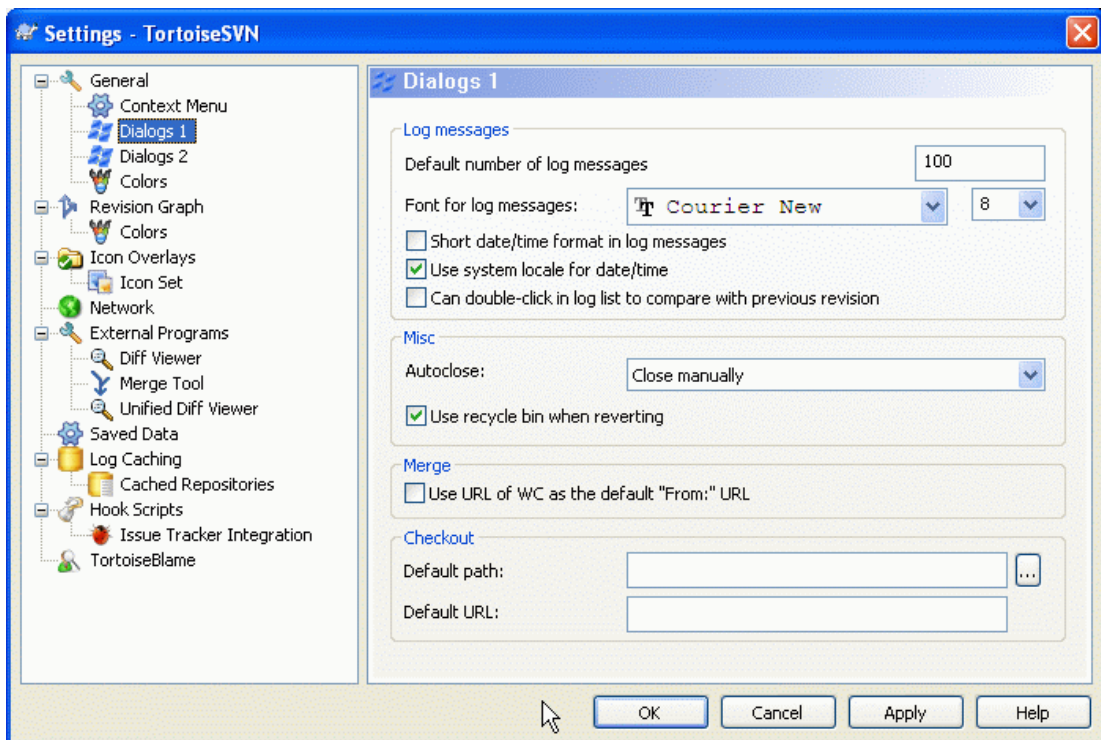


Figura 4.53. A Caixa de Diálogo Preferências, Página Diálogos 1

Esta caixa de diálogo permite-te configurar algumas das caixas de diálogo do TortoiseSVN, da maneira que mais gostas delas.

Número de mensagens de registo por omissão

Limita o número de mensagens de registo que o TortoiseSVN adquire quando tu seleccionas primeiro o TortoiseSVN → Mostrar Registo Útil para conexões lentas com o servidor. Tu podes sempre usar o Mostrar Todos ou Próximos 100 para obter mais mensagens.

Tipo de letra para mensagens de registo

Selecciona o tipo de letra e o tamanho usado para mostrar a própria mensagem de registo, no painel ao meio da caixa de diálogo de Registo de Revisão, e quando na composição das mensagens na caixa de diálogo Submeter.

Formato conciso de data/hora nas mensagens de registo

Se as longas mensagens padrão ocupam muito espaço no teu écran, usa o formato conciso.

Faça um duplo clique na lista de mensagens de registo para comparar com uma revisão anterior

Se te encontras frequentemente a comparar revisões no painel de topo da caixa de diálogo Registo, podes usar esta opção para permitir essa acção ao duplo clique. O que não está activo por defeito, porque adquirir as diferenças é normalmente um processo moroso e a maior parte das pessoas prefere evitar esperar após um duplo-clique accidental, que é a razão por que esta opção não está activa por defeito.

Caixas de Diálogo de Progresso

O TortoiseSVN pode fechar automaticamente todas as caixas de diálogo de progresso, quando uma acção é finalizada sem erros. Esta preferência permite-te seleccionar as condições para fechar as caixas de diálogo. A preferência por defeito (recomendada) é Fechar manualmente o que te permite rever todas as mensagens e rever o que aconteceu. No entanto tu podes decidir que queres ignorar alguns tipos de mensagens, e teres a opção da caixa de diálogo a fechar automaticamente, se não existirem alterações críticas.

Fechar automaticamente se não houver integrações, adições ou remoções significa que a caixa de diálogo progresso irá fechar-se no caso de actualizações simples, mas se alterações do repositório foram integradas com as tuas, ou se qualquer ficheiros foi adicionado ou removido, a caixa de diálogo permanecerá aberta. Irá também permanecer aberta se existir algum conflito ou erro durante a operação.

Fechar automaticamente se não houver integrações, adições ou remoções para operações locais significa que a caixa de progresso irá fechar como para Fechar automaticamente se não houver integrações, adições ou remoções, mas só para operações locais como adicionar ficheiros ou reverter alterações. Para operações remotas a caixa permanecerá aberta.

Fechar automaticamente se não houver conflitos relaxa ainda mais o critério e fechará a caixa de diálogo mesmo se existem integrações, adições ou remoções. No entanto se existir algum conflito ou erro, a caixa permanecerá aberta.

Fechar automaticamente se não houver erros fecha sempre a caixa de diálogo mesmo se existirem conflitos. A única condição que mantém a caixa de diálogo aberta é se existir uma condição de erro, que ocorre quando o Subversion não consegue finalizar a tarefa. Por exemplo se uma actualização falha, porque o servidor não está acessível, ou uma submissão falha, porque a cópia de trabalho está desactualizada.

Usar a reciclagem quando a reverter

Quando reverteres as modificações locais, as tuas alterações são descartadas. O TortoiseSVN dá-te uma rede de segurança extra, enviando o ficheiro modificado para o cesto de reciclagem, antes de trazer de volta a cópia pura. Se preferires saltar o cesto de reciclagem desselecciona esta opção.

Usar o URL da cópia de trabalho como o URL “De:” por omissão

Na caixa de diálogo integrar, o comportamento por defeito é o URL De: ser recordado entre integrações. No entanto, algumas pessoas gostam de executar integrações a partir de muitos pontos diferentes da sua hierarquia, e acham mais fácil começar com o URL da cópia de trabalho corrente. Isto pode ser editado para se referir a um caminho paralelo noutra ramo.

Caminho por defeito para colonizar

Tu podes especificar o caminho, por defeito, para as colonizações. Se manteres todas as tuas colonizações num único local, será útil ter uma unidade e pasta pré-preenchida, para que só tenhas de adicionar no fim o nome da nova pasta.

URL por omissão de colonização

Podes também especificar o URL por omissão para colonizações. Se, com frequência, colonizas subprojectos de algum grande projecto, poderá ser útil ter um URL pré-preenchido, para que só tenhas de adicionar o nome do subprojecto no fim.

4.30.1.3. Preferências das Caixas de Diálogo TortoiseSVN 2

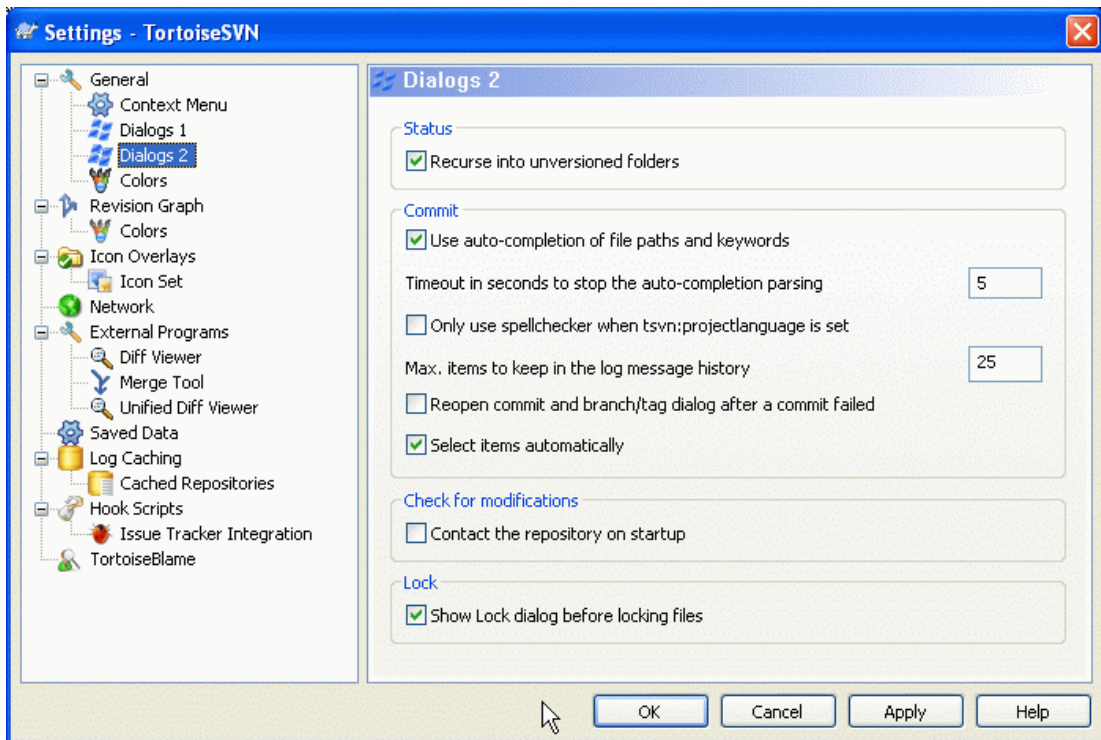


Figura 4.54. A Caixa de Diálogo Preferências, Página Diálogos 2

Usar recursividade para entrar em pastas sem controlo de versões

Se esta caixa for verificada (estado por defeito), então sempre que o estado de uma pasta não versionada é mostrado na caixa de diálogo Adicionar, Submeter ou Verificar alterações, cada ficheiro ou pasta filho é também mostrado. Se desmarcares esta caixa, apenas o pai não versionado é mostrado. Desmarcar, reduz a confusão nestas caixas de diálogo. Nesse caso se seleccionares uma pasta não versionada para Adicionar, esta é adicionada recursivamente.

Usar auto-completamento para caminhos de arquivo e palavras-chave

A caixa de diálogo submeter inclui a facilidade de analisar uma lista de nomes de ficheiros a submeter. Quando tu digitas as 3 primeiras letras de um item da lista, a caixa de auto-completamento aparecerá, e podes premir o Enter para completar o nome do ficheiro. Verifica a caixa para activar esta funcionalidade.

Prazo em segundos até para a análise para o auto-completamento

O analisador do auto-completamento pode ser algo lento se existe um grande número de ficheiros a analisar. Este prazo evita que a caixa de submissão seja mantida presa por demasiado tempo. Se estás a perder informação importante de auto-completamento, poderás então estender o valor deste prazo.

Usar o verificador ortográfico apenas quando a `tsvn:projectlanguage` estiver definida

Se não desejas utilizar o verificador ortográfico para todas as submissões, verifica esta caixa. O verificador ortográfico ainda estará activo nos projectos onde as suas propriedades o requerer.

Máximo de itens a preservar no histórico de mensagens de registo

Quando digitas uma mensagem de registo, na caixa de diálogo de registo, o TortoiseSVN armazena-a para possível reutilização futuro. Por defeito irá manter as últimas 25 mensagens de registo para cada repositório, mas tu podes personalizar aqui esse valor. Se tens muitos repositórios diferentes poderás querer reduzir este valor de modo a evitar encher o teu cartório de registo.

De notar que esta preferência se aplica só a mensagens digitadas neste computador. Não tem nada a ver com a cache de registo.

Reabrir a caixa de diálogo de submissão e ramificação/etiquetagem após uma submissão falhada

Quando uma submissão falha por alguma razão (a cópia de trabalho precisa de actualização, gancho de pre-commit rejeita a submissão, error de rede, etc.), tu podes seleccionar esta opção para manter a caixa de diálogo de submeter aberta para nova tentativa. No entanto, deverás estar ciente de que isto poderá levar a problemas. Se a falha significar que necessitas de actualizar a tua cópia de trabalho e essa actualização pode levar a conflitos, que tens de resolver primeiro.

Seleccionar itens automaticamente

O comportamento normal na caixa de diálogo submeter é para todos os itens (versionados) modificados serem seleccionados automaticamente para submissão. Se preferires começar com nada seleccionado, e escolher manualmente os itens a submeter, desmarca esta caixa.

Contactar o repositório ao iniciar

A caixa de diálogo Verificar alterações, verifica a cópia de trabalho por defeito, e só contacta o repositório quando tu clicas Verificar o repositório. Se queres verificar sempre o repositório, podes usar esta preferência para executar automaticamente essa acção.

Mostrar caixa de diálogo de Bloqueio antes de bloquear arquivos

Quando seleccionas um ou mais ficheiros e de seguida usas TortoiseSVN → Bloquear para bloquear esses ficheiros, nalguns projectos é costume de escrever uma mensagem de bloqueio a explicar a razão porque os bloqueastes. Se não usares mensagens de bloqueio poderás desmarcar esta caixa, de modo a saltares esta caixa de diálogo, e bloquear de imediato os ficheiros.

Se usares o comando de bloqueio numa pasta, serás sempre levado à presença de uma caixa de diálogo de bloqueio, já que também te dará a opção de seleccionar os ficheiros a bloquear.

Se o teu projecto está a usar a propriedade `tsvn:lockmsgminsize` verás a caixa de diálogo bloquear, independentemente desta preferência, porque o projecto *requer* mensagens de bloqueio.

4.30.1.4. Preferências de cor do TortoiseSVN

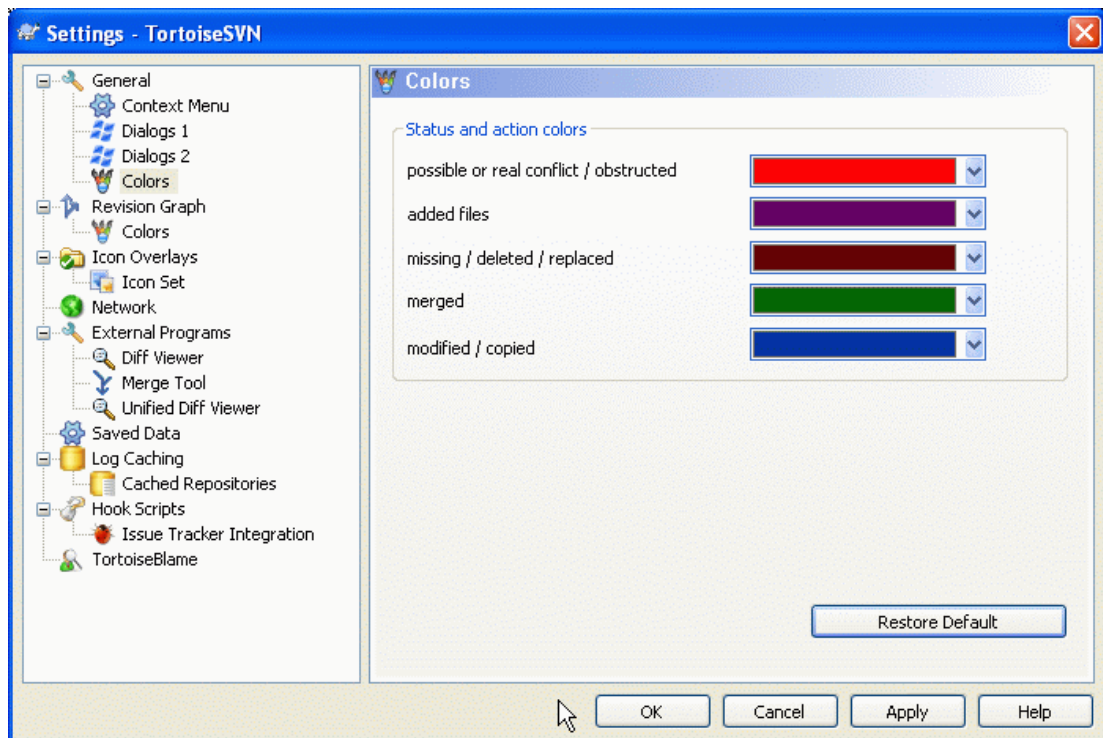


Figura 4.55. A Caixa de Diálogo Preferências, Página de Cores

Esta caixa de diálogo permite-te configurar as cores do texto usado nas caixas de diálogo do TortoiseSVN, do modo que tu gostas delas.

em conflito possível ou real / obstruído

Um conflito ocorreu durante a actualização, ou pode ocorrer durante a integração. A actualização está obstruída por um ficheiro/pasta existente, não versionado, do mesmo nome que o versionado.

Esta cor também é usada para mensagens de erro nas caixas de diálogo de progresso.

arquivos adicionados

Itens adicionados ao repositório.

em falta / removido / substituído

Itens removidos do repositório, desaparecidos ou removidos da cópia de trabalho e substituídos por outro ficheiro do mesmo nome.

integrado

Alterações, do repositório, sucessivamente integradas na CT sem criar qualquer conflito.

alterado / copiado

Adicionar com histórico, ou caminhos copiados no repositório. Também usado na caixa de diálogo de registo para entradas que incluem itens copiados.

Nó removido

Um item que foi removido do repositório.

Nó adicionado

Um item que foi adicionado ao repositório por uma operação de adição, cópia ou movimentação.

Nome de nó alterado

Um item que cujo nome foi alterado dentro do repositório.

Nó substituído

O item original foi removido e um novo item com o mesmo nome substituído.

4.30.2. Preferências do Gráfico de Revisões

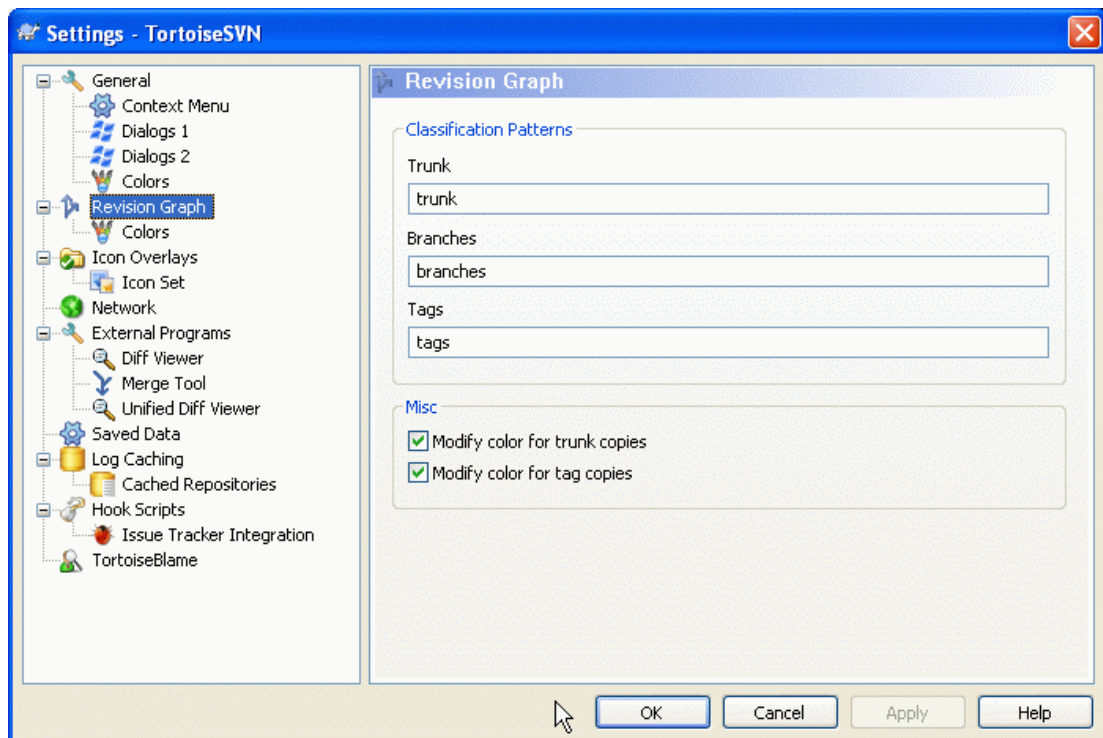


Figura 4.56. A Caixa de Diálogo Preferências, Página Gráfico de Revisões

Padrões de classificação

O gráfico de revisões tenta mostrar uma imagem mais nítida da estrutura do teu repositório, distinguindo entre o trunk, ramos e etiquetas. Como não existe tal classificação instalada no Subversion, esta informação é então extraída a partir de nomes dos caminhos. As preferências por defeito assumem que irás utilizar os nomes convencionais em Inglês, como sugerido na documentação do Subversion, mas é claro que o uso que lhe dás pode ser diverso.

Especifica os padrões usados para reconhecer esses caminhos, para tal usa as três caixas providenciadas. Os padrões serão correspondidos, sem distinção de maiúsculas e minúsculas, mas deverás especificá-los em minúsculas. Os caracteres livres * e ? funcionaram como de costume e podes usar ; para separar vários padrões. Não necessitas incluir qualquer carácter em branco adicional, já que tal irá ser incluído na especificação da correspondência.

Modificar Cores

São usadas cores no gráfico de revisões para indicar o tipo do nó, i.e. se o nó foi adicionado, removido ou renomeado. De modo a te ajudar a identificar as classificações dos nós, tu podes permitir que o gráfico de revisões misture cores de modo a te dar tanto a indicação do tipo do nó como da classificação. Se a caixa está desmarcada, a cor é usada só para indicar o tipo do nó. Utiliza a caixa de diálogo de selecção de cor para atribuir as cores específicas usadas.

4.30.2.1. Cores do Gráfico de Revisões

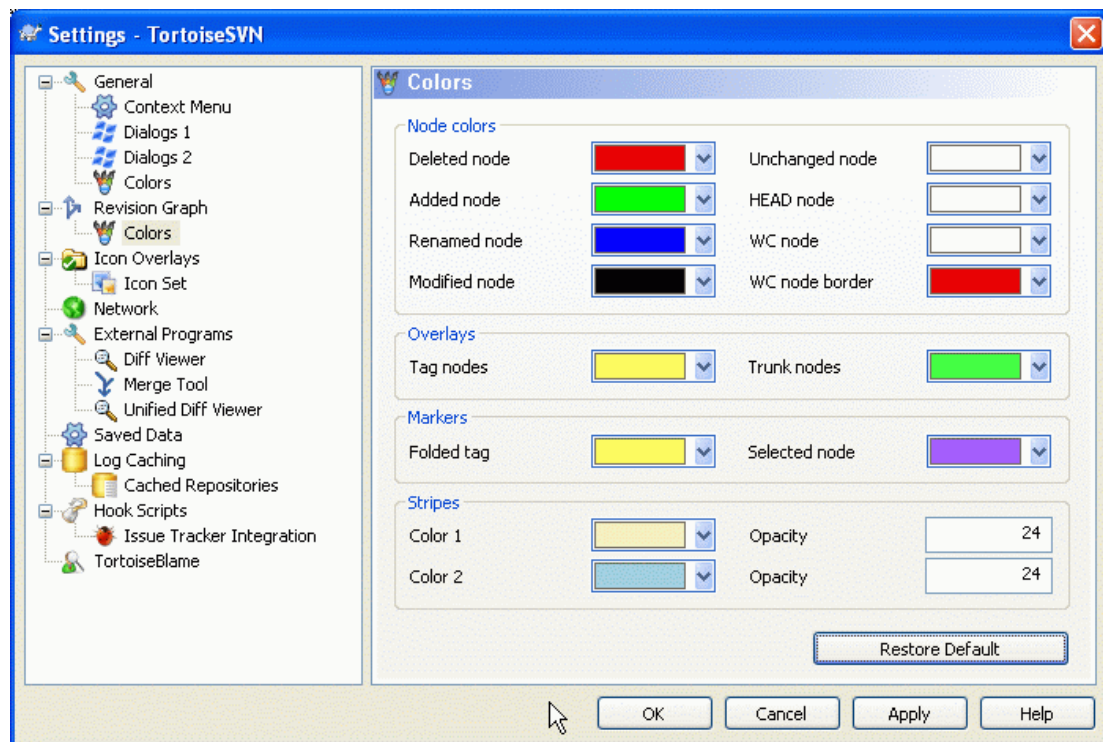


Figura 4.57. A Caixa de Diálogo preferências,Página Cores do Gráfico de Revisões

Esta página permite-te configurar as cores usadas. Tomar atenção que a cor aqui especificar é uma cor sólida. A maior parte dos nós é colorido usando uma mistura da cor do tipo do nó, a cor de fundo, e opcionalmente, a cor de classificação.

Nó Removido

Itens que foram removidos, e não copiados para outro lado, na mesma revisão.

Nó Adicionado

Itens recentemente adicionados ou copiados (adicionar com histórico).

Nome do Nó Alterado

Itens removidos de uma localização e adicionados noutra, na mesma revisão.

Nó Alterado

Alterações simples, sem qualquer adição ou remoção.

Nó não Alterado

Poderá ser utilizada para mostrar a revisão usada como origem da cópia, mesmo quando nenhuma alteração (ao item objecto do gráfico) tenha ocorrido nessa revisão.

Nó HEAD

Revisão corrente HEAD, no repositório.

Nó WC

Se optares por mostrar um nó extra para a tua cópia de trabalho modificada, anexado à tua revisão da última submissão no gráfico, usa esta cor.

Limite do Nó WC

Se optares por mostrar se a cópia de trabalho foi modificada, usa esta cor de moldura no nó WC, para o caso de serem encontradas alterações.

Nós Etiquetas

Nós classificados como etiquetas podem ser misturados com esta cor.

Nós do Tronco

Nós classificados com tronco poder ser misturados com esta cor.

Marcadores de Etiqueta Dobrada

Se usares etiquetas dobradas, para poupar espaço, as etiquetas são marcadas na cópia de origem como um bloco desta cor.

Marcadores de Nó Seleccionado

Quando clicas com o botão esquerdo num nó seleccionando-o, o marcador usado para indicar esta selecção é um bloco desta cor.

Tiras

Essas cores são usadas quando o gráfico é dividido em sub-árvores, e o fundo é colorido com tiras alternadas, para ajudar a identificar as árvores separadas.

4.30.3. Preferências de Sobreposição de Ícones

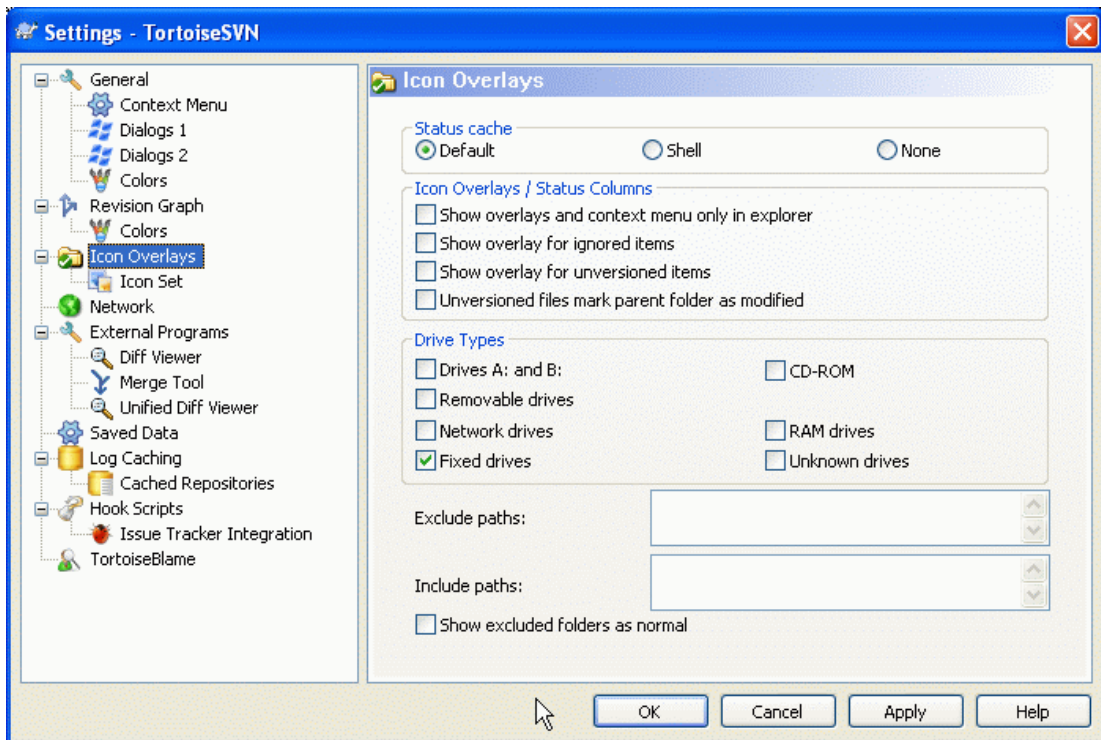


Figura 4.58. A Caixa de Diálogo Preferências, Página Sobreposição de Ícones

Esta página permite-te escolher os itens para os quais o TortoiseSVN irá mostrar sobreposições de ícones.

Por defeito as sobreposições de ícones e menus de contexto irão aparecer em todas as caixas de abrir/guardar, tal como no Explorador do Windows. Se queres que apareçam *só* no Explorador do Windows verifica a caixa **Mostrar sobreposições e menu de atalho apenas no explorador**.

Aos itens ignorados e não versionados não é normalmente dada uma sobreposição. Se queres mostrar sobreposições nesses casos, basta verificares as caixas.

Podes também optar por marcar as pastas como modificadas, se estas contêm itens não versionados. Isto pode ser útil para te lembrar que criaste novos ficheiros que não foram ainda adicionados ao controlo de versões. Esta opção só está disponível quando usas a opção de cache (ver abaixo) de estado *por defeito*.

Sendo que demora algum tempo a adquirir o estado da cópia de trabalho, o TortoiseSVN usa a cache para armazenar o estado, par que o explorador não fique muito tempo pendurado a mostrar as sobreposições. Podes escolher o tipo da cache que o TortoiseSVN deverá usar, de acordo com o teu sistema e tamanho de cópia de trabalho:

Por omissão

Armazena na cache toda a informação num processo separado (TSVNCache.exe). Esse processo monitoriza todas as unidades à procura de alterações, e adquire de novo o estado, se algum ficheiro dentro da cópia de trabalho foi modificado. O processo corre com a prioridade mais baixa para que os outros programas não fiquem pendurados por sua causa. Isso também significa que a informação do estado não é dada em *tempo real*, mas pode levar alguns segundos até as sobreposições serem actualizadas.

Vantagem: As sobreposições mostram o estado recursivamente, i.e. se um ficheiro, bem no interior da tua cópia de trabalho é alterado, todas as pastas até à raiz da cópia de trabalho irão também mostrar a sobreposição modificado. E visto que o processo pode enviar notificações para a shell, as sobreposições na árvore da vista à esquerda, também serão normalmente alteradas.

Desvantagem: O processo corre continuamente mesmo que não estejas a trabalhar nos teus projectos. E também usa cerca de 10-50 MB de RAM, dependendo do número e tamanho das tuas cópias de trabalho.

Shell

A cache é executada directamente do interior da dll da extensão da shell, mas apenas para a pasta correntemente visível. De cada vez que navegas para outra pasta, a informação de estado é novamente adquirida.

Vantagem: necessita apenas de muito pouca memória (à volta de 1MB de RAM) e pode mostrar o estado em *tempo real*.

Desvantagem: Visto que só uma pasta é posta em cache, as sobreposições não mostram recursivamente o estado. Para cópias de trabalho grandes, pode demorar mais tempo a mostrar uma pasta no explorador do que a opção por omissão. A coluna mime-type também não está disponível.

Nenhum

Com esta preferência o TortoiseSVN não adquire de modo algum o estado no Explorador. Devido a isso, os ficheiros não obtêm uma sobreposição e as pastas só obtêm a sobreposição 'normal', se estiverem versionadas. Nenhuma outra sobreposição é mostrada, e também não está disponíveis colunas extra.

Vantagem: Não usa absolutamente nenhuma memória adicional e não desacelera o Explorador ao navegar.

Desvantagem: Não é mostrada no Explorador a informação de estado de ficheiros e pastas. Para veres se as tuas cópias de trabalho foram modificadas terás de usar a caixa de diálogo “Verificar alterações”.

O próximo grupo permite-te seleccionar quais as classes de armazenamento que devem mostrar sobreposições. Por defeito só unidades fixas estão seleccionadas. Podes mesmos desactivar todas as sobreposições de ícones, mas qual seria a piada?

Unidades de rede podem ser muito lentas, pelo que por defeito, os ícones não são mostrados para cópias de trabalho localizadas em partilhas de rede.

Unidades Flash USB parecem ser um caso especial em que o tipo de unidade é identificado pelo próprio dispositivo. Algumas aparecem como unidades fixas e outra como amovíveis.

O Excluir caminhos é usado para comunicar ao TortoiseSVN que esses caminhos para os quais *não* deve ser mostrado sobreposições de ícones e colunas de estado. Isto é útil se tens algumas cópias de trabalho muito grandes, contendo só bibliotecas que não irás alterar e por isso não necessitam de sobreposições. por exemplo:

`f:\development\SVN\Subversion` irá desactivar as sobreposições *só* nessa pasta específica. Podes no entanto ver ainda as sobreposições, em todas as pastas e ficheiros, dentro dessa pasta.

`f:\development\SVN\Subversion*` irá desactivar as sobreposições em *todos* os ficheiros e pastas cujos caminhos comecem por `f:\development\SVN\Subversion`. Isso significa que não verás sobreposições para qualquer ficheiro ou pasta debaixo desse caminho.

O mesmo se aplica para o Incluir caminhos. À excepção de que para esses caminhos as sobreposições são mostradas mesmo se estiverem desactivadas para esse tipo de unidade, ou por um caminho de exclusão, especificado acima.

Os utilizadores por vezes perguntam como essas preferências interagem, e a resposta definitiva é:

```
if (path is in include list)
```



```
show overlays
if (path is allowed drive type) AND (path is not in exclude list)
show overlays
```

A lista de incluídos mostrará *sempre* as sobreposições. De outro modo as sobreposições são sempre mostradas para todas as unidades marcadas *a não ser que* o caminho esteja excluído.

O TSVNCache.exe também usa esses caminhos para restringir a sua monitorização. Se tu queres olhar só para algumas pastas em particular, desactiva todos os tipos de unidades e inclui apenas as pastas que tu queres que sejam especificamente monitoradas.



Excluir unidades SUBST

É conveniente usar uma unidade SUBST para aceder às tuas cópias de trabalho, e.g. usando o comando

```
subst T: C:\TortoiseSVN\trunk\doc
```

No entanto isto pode provocar a não actualização das sobreposições, já que o TSVNCache só receberá uma notificação quando o ficheiro é alterado, e essa é normalmente o caminho original. Isto significa que as tuas sobreposições no caminho `subst` poderá nunca ser actualizado.

Uma maneira fácil de dar a volta a isto é excluir o caminho original de mostrar sobreposições, para que as sobreposições apareçam, por sua vez, no caminho `subst`.

Por vezes tu excluirás áreas que contêm cópias de trabalho, que poupam o TSVNCache de monitorar e pesquisar por alterações, mas queres à mesma a indicação visual que tais pastas são versionadas. A caixa de verificação **mostrar pastas excluídas como normais** te irá permitir fazer isso. Com esta opção as pastas versionadas em qualquer área excluída (tipo de unidade não verificada, ou especificamente excluída) será mostrada como normal e actualizada, com uma marca de verificação verde. Isto lembrar-te-á que estás a olhar para uma cópia de trabalho, mesmo que as sobreposições da pasta não estejam correctas. Os ficheiros não obterão nenhuma sobreposição. Ter em atenção que os menus de contexto ainda funcionarão mesmo que as sobreposições não sejam mostradas.

Como excepção especial a isto, as unidades A: e B: nunca são consideradas para a opção **Mostrar pastas excluídas como 'normais'**. Isto é porque o Windows força a busca na unidade, o que pode resultar num atraso de vários segundos quando o Explorador arranca, mesmo se o teu PC não tem uma unidade de disquete.

4.30.3.1. Seleção de Conjunto de Ícones

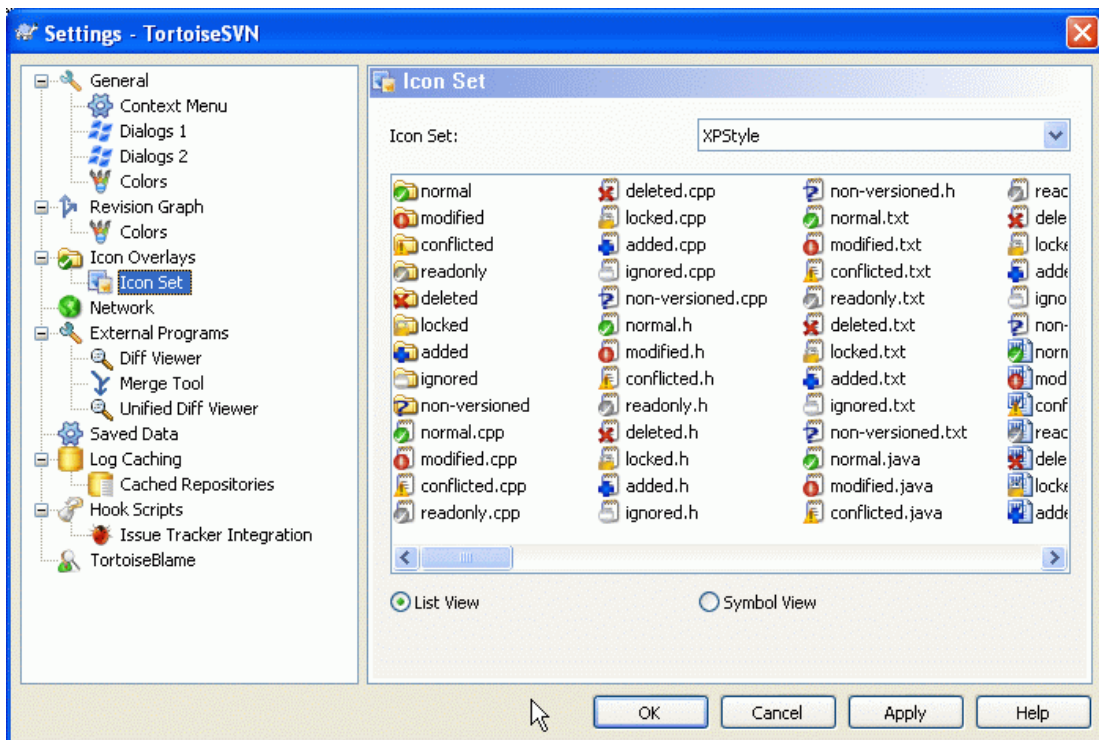


Figura 4.59. A Caixa de Diálogo preferências, Página Conjunto de Ícones

Podes alterar o conjunto de sobreposições para o que mais gostares. De notar que se alterares o conjunto de sobreposições, poderás ter de reiniciar o teu computador para as alterações surtam efeito.

4.30.4. Preferências de Rede

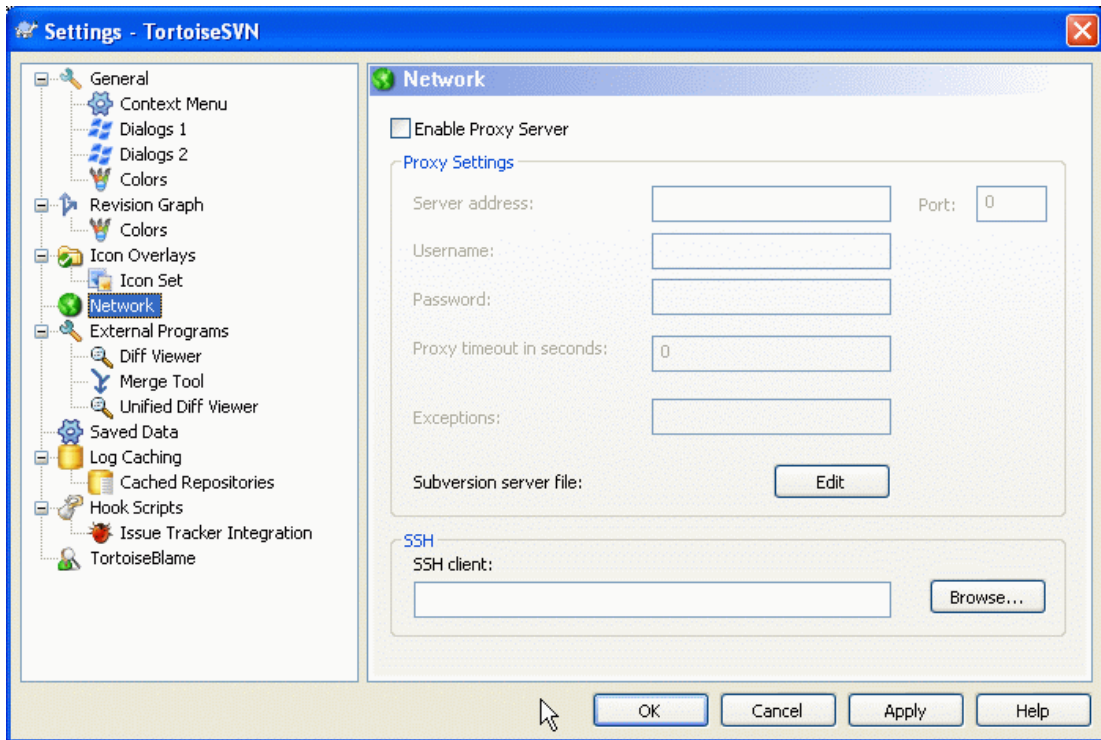


Figura 4.60. A Caixa de Diálogo Preferências, Página de Rede

Aqui podes configurar o teu servidor proxy, se necessitas de passar através da firewall da tua empresa.

Se necessitas de configurar as preferências do proxy por repositório, precisas de configurar isso no ficheiro `servers` do Subversion. Usa o **Editar** para ir lá directamente. Consulta o [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] para mais detalhes em como utilizar este ficheiro.

Podes também especificar qual o programa que o TortoiseSVN deverá usar para estabelecer uma ligação segura com um repositório `svn+ssh`. Nós recomendamos que uses o `TortoisePlink.exe`. Este é uma versão do popular programa `Plink` e é incluído com o TortoiseSVN, mas é compilado como uma aplicação sem janelas, para que não tenhas uma caixa de diálogo de DOS a aparecer-te sempre que te autentiques.

Deverás especificar o caminho completo para o executável. Que no caso do `TortoisePlink.exe` será, por defeito, a pasta `bin` do TortoiseSVN. Usa o botão **Navegar** para te ajudar a localizá-lo. Tem em conta que, se o caminho contiver espaços deverás pô-los entre aspas, e.g.

```
"C:\Program Files\TortoiseSVN\bin\TortoisePlink.exe"
```

Um efeito secundário de não ter uma janela é que não há sitio para onde irem as mensagens de erro, pelo que se a autenticação falha, irás receber apenas uma mensagem dizendo algo como “Não foi possível escrever para o destino por defeito”. Por essa razão, nós recomendamos que primeiro configure usando o `Plink` padrão. Quando tudo já está a funcionar, podes então usar o `TortoisePlink` com exactamente os mesmos parâmetros.

O `TortoisePlink` não tem nenhuma documentação sua, porque é apenas uma variante menor do `Plink`. Descobre mais sobre os parâmetros de linha de comandos a partir de [PuTTY website](http://www.chiark.greenend.org.uk/~sgtatham/putty/) [http://www.chiark.greenend.org.uk/~sgtatham/putty/]

Para evitares estares sempre a ser interrompido para inserires a palavra-passe, poderás considerar o uso de uma ferramenta de cache da palavra-passe como o `Pageant`. Esta está também disponível para descarga no sítio do `PuTTY`.

Finalmente, configurar o SSH no servidor e clientes é um processo não trivial, que está para além do âmbito deste documento. No entanto poderás encontrar um manual nas FAQs so TortoiseSVN, listadas em [Subversion/TortoiseSVN SSH How-To](http://tortoisesvn.net/ssh_howto) [http://tortoisesvn.net/ssh_howto].

4.30.5. Preferências de Programas Externos

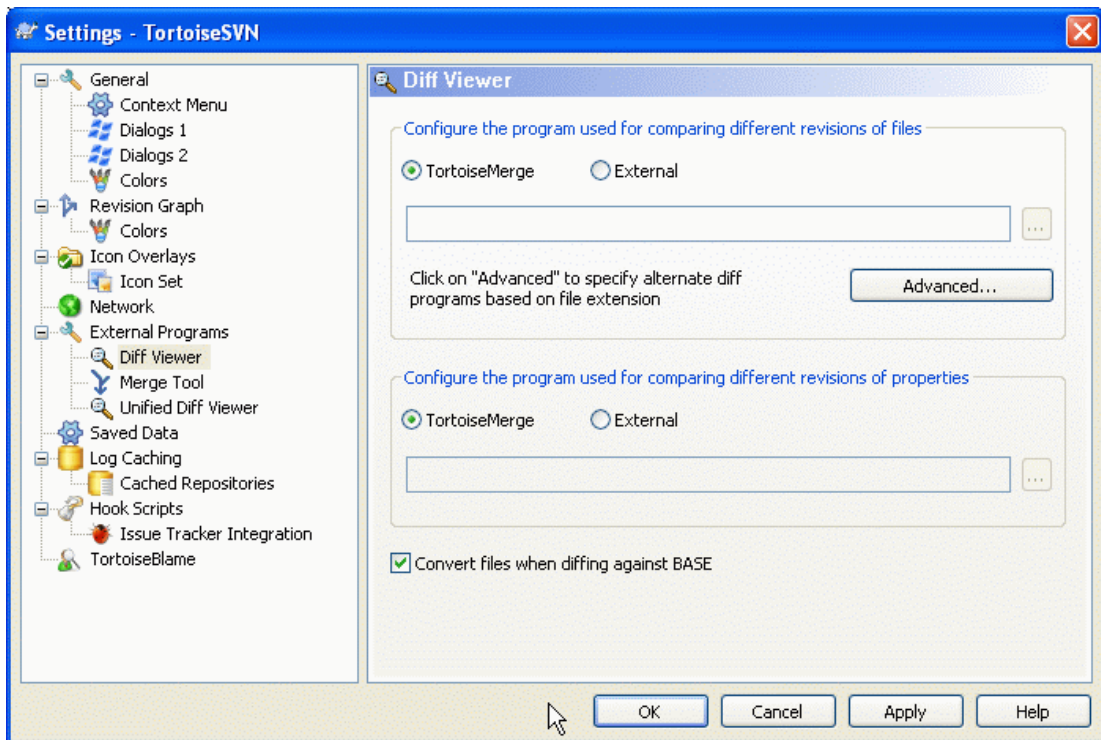


Figura 4.61. A Caixa de Diálogo Preferências, Página Visualizador de Comparação

Aqui, poderás definir os teus próprios programas de Comparação/Integração que o TortoiseSVN deverá usar. A preferência por defeito é a utilização do TortoiseMerge, que está instalado juntamente com o TortoiseSVN.

Consulta o [Secção 4.10.5, “Ferramentas de Comparação/Integração”](#), para uma lista de alguns programas de Comparação/Integração que as pessoas podem usar com o TortoiseSVN.

4.30.5.1. Visualizador de Comparações

Um programa de comparação externo pode ser usado para comparar diferentes revisões de ficheiros. O programa externo irá necessitar de obter os nomes dos ficheiros, a partir da linha de comandos juntamente com outras opções. O TortoiseSVN usa parâmetros de substituição prefixados com %. Quando encontra um desses, irá substituí-lo pelo valor apropriado. A order dos parâmetros irá depender do programa de comparação que usares.

%base

O ficheiro original sem as tuas alterações

%bname

O título da janela para o ficheiro base

%mine

O teu ficheiro, com as tuas alterações.

%yname

O título da janela do teu ficheiro

Os títulos das janelas não são nomes de ficheiros puros. O TortoiseSVN trata-os como nomes a mostrar e cria os nomes de acordo com tal. Pelo que e.g. se fores efectuar uma comparação de um ficheiro na revisão

123 com um ficheiro da tua cópia de trabalho, os nomes serão Nome do ficheiro : revisão 123 e Nome do ficheiro : cópia de trabalho

Por exemplo, com o ExamDiff Pro:

```
C:\Path-To\ExamDiff.exe %base %mine --left_display_name:%bname
--right_display_name:%ynome
```

or with KDiff3:

```
C:\Path-To\kdiff3.exe %base %mine --L1 %bname --L2 %ynome
```

or with WinMerge:

```
C:\Path-To\WinMerge.exe -e -ub -dl %bname -dr %ynome %base %mine
```

or with Araxis:

```
C:\Path-To\compare.exe /max /wait /title1:%bname /title2:%ynome
%base %mine
```

Se usares a propriedade `svn:keywords` para expandir as palavras-chave, e em particular a *revisão* do ficheiro, então poderá haver diferenças entre ficheiros, que são puramente devido a valor corrente da palavra-chave. Também se usares a `svn:eol-style = native` o ficheiro BASE irá ter fins-de-linha LF puros, enquanto o teu ficheiro irá ter fins-de-linha CR-LF. O TortoiseSVN irá, normalmente e automaticamente, esconder essas diferenças ao analisar primeiro o ficheiro BASE, de modo a expandir as palavras-chave e fins de linha antes de efectuar a operação de comparação. No entanto isto poder levar bastante tempo com grandes ficheiros. Se Converter arquivos ao comparar com a BASE está desmarcado então o TortoiseSVN irá saltar o pré-processamento dos ficheiros.

Podes também especificar uma ferramenta diferente de comparação para usares nas propriedades do Subversion. Visto que essas tendem a ser pequenas peças de texto simples, poderás querer usar um visualizador mais simples e compacto.

Se configuraste uma ferramenta alternativa de comparação, poderás aceder ao TortoiseMerge e a ferramenta de terceiros a partir dos menus de contexto. Menu de contexto → Comparar usa a ferramenta de comparação primária, e **Shift**+ Menu de contexto → Comparar usa a ferramenta de comparação secundária.

4.30.5.2. Ferramenta de Integração

Um programa de integração externo usado para solucionar ficheiros em conflito. A substituição de parâmetros é usada da mesma maneira que com o Programa de Comparação.

`%base`

O ficheiro original, sem as tuas alterações ou as dos outros

`%bname`

O título da janela para o ficheiro base

`%mine`

O teu ficheiro, com as tuas alterações

`%ynome`

O título da janela do teu ficheiro

`%theirs`

o ficheiro, como está no repositório

%tname

O título da janela para o ficheiro no repositório

%merged

o ficheiro em conflito, o resultado da operação de integração

%mname

O título da janela para o ficheiro integrado

Po exemplo para o Perforce Merge:

```
C:\Path-To\P4Merge.exe %base %theirs %mine %merged
```

ou com o KDiff3:

```
C:\Path-To\kdiff3.exe %base %mine %theirs -o %merged
--L1 %bname --L2 %yname --L3 %tname
```

ou com o Araxis:

```
C:\Path-To\compare.exe /max /wait /3 /title1:%tname /title2:%bname
/title3:%yname %theirs %base %mine %merged /a2
```

ou com o WinMerge (2.8 ou superior):

```
C:\Path-To\WinMerge.exe %merged
```

4.30.5.3. Preferências Avançadas de Comparar/Integrar

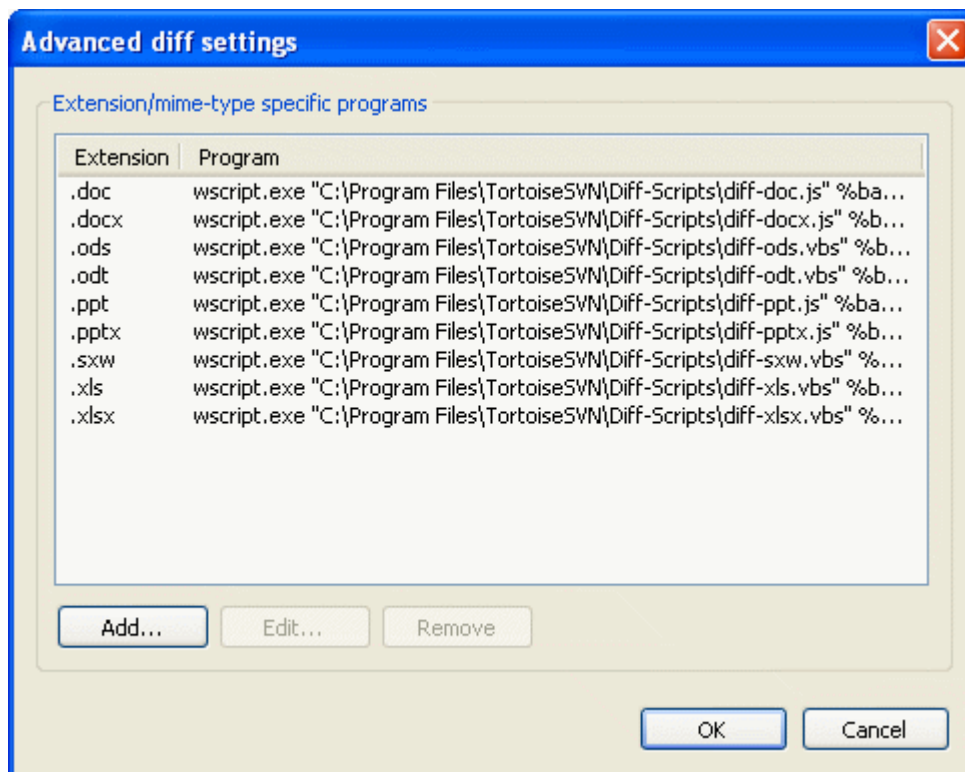


Figura 4.62. A Caixa de Diálogo Preferências, Caixa de Diálogo Comparar/Integrar Avançados

Nas preferências avançadas, poderás definir um programa diferente para comparar e integrar por cada extensão de ficheiro. Por exemplo poderás associar o Photoshop como o programa de “Comparar” para ficheiros `.jpg` :-) Podes também associar a propriedade `svn:mime-type` com um programa de comparação ou integração.

Para associar usando uma extensão de ficheiro, necessitarás de associar a extensão. Usa o `.bmp` para descrever ficheiros de bitmap do Windows. Para associar usando a propriedade `svn:mime-type`, especifica o tipo mime incluindo a barrar, por exemplo `text/xml`.

4.30.5.4. Visualizador de Comparações Unificadas

Um programa de visualização para ficheiros de comparação-unificada (ficheiros remendo). Não são requeridos parâmetros. A opção por defeito verifica por uma associação para ficheiros `.diff`, e então para ficheiros `.txt`. Se não tens um visualizador para ficheiros `.diff` irás, com muita probabilidade, obter o Notepad.

O programa NotePad original do Windows não se comporta bem com ficheiros que não têm o fim-de-linha padrão CR-LF. Visto que a maioria dos ficheiros de comparação-unificada têm fins-de-linha com LF puro, eles não são visualizados correctamente no NotePad. No entanto podes descarregar uma substituição gratuita para o NotePad em *Notepad2* [<http://www.flos-freeware.ch/notepad2.html>], que não apenas mostra os fins-de-linha correctamente, mas também colora as linhas adicionadas e removidas.

4.30.6. Preferências de Dados Guardados

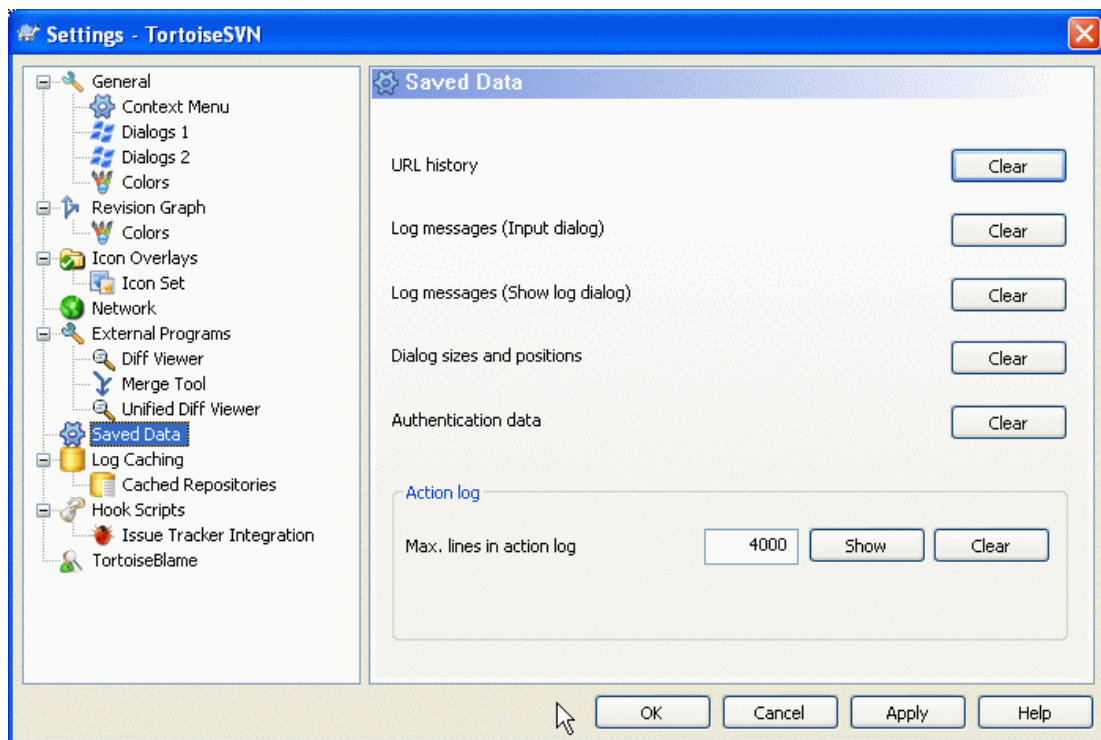


Figura 4.63. A Caixa de Diálogo, Página de Dados Guardados

Para tua comodidade o TortoiseSVN guarda muitas das preferências que usas, e lembra-se de onde andaste ultimamente. Se quiseres limpar essa cache de dado poderás o fazer aqui.

Histórico de URL

Sempre que povoas uma cópia de trabalho, integras alterações ou usas o navegador de repositório, o TortoiseSVN mantém um registo dos URLs recentemente usados e oferece-lhes uma combo box. Por vezes a lista é povoada com URLs desactualizados pelo que será útil periodicamente limpá-la.

Se queres remover um único item de uma das combo boxes, poderás fazê-lo no local. Clica apenas na seta para abrir a lista da combo box, move o rato por cima do item que desejas remover e digita **Shift+Del**.

Mensagens de registo (Caixa de diálogo de entrada)

O TortoiseSVN armazena as mensagens de registo recentes, que tu introduziste. Elas são armazenadas por repositório pelo que se acederes a muitos repositórios, essa lista poderá crescer bastante.

Mensagens de registo (Mostrar caixa de diálogo de registo)

O TortoiseSVN faz cache das mensagens de registo trazidas pela caixa de diálogo Mostrar Registo, para te poupar tempo quando a seguir vês o registo. Se alguém editar uma mensagem de registo e tens essa mensagem em cache, não irás ver a alteração até limpares a mesma. O uso da cache de mensagens está activado na página **Caches de Registos**.

Dimensões e posições de caixas de diálogo

Muitas caixas de diálogo lembram-se do tamanho e posição com que tu as usaste por último.

Dados de autenticação

Quando te autenticas num servidor Subversion o nome de utilizador e a palavra-chave são guardadas localmente na cache, para que não tenhas de estar sempre a introduzi-las. Poderás querer limpar esta informação, por razões de segurança, ou porque queres aceder ao repositório com um nome de utilizador diferente... o João sabe que estás a usar este PC?

Se queres limpar os dados de autenticação para um servidor em particular, consulta [Secção 4.1.5, "Autenticação"](#) para mais instruções em como encontrar os dados na cache.

Registo de acções

O TortoiseSVN mantém um registo de tudo o que foi escrito nas caixas de diálogo de progresso. Isto pode ser útil para quando, por exemplo, quiseses verificar o que aconteceu num comando recente de actualização.

O ficheiro de registo é limitado no seu tamanho e quando cresce demasiado, é descartado o conteúdo mais antigo. Por defeito são mantidas 4000 linhas, mas podes personalizar este número.

Daqui podes ver o conteúdo do ficheiro de registo, e também limpá-lo.

4.30.7. Cache de Registo

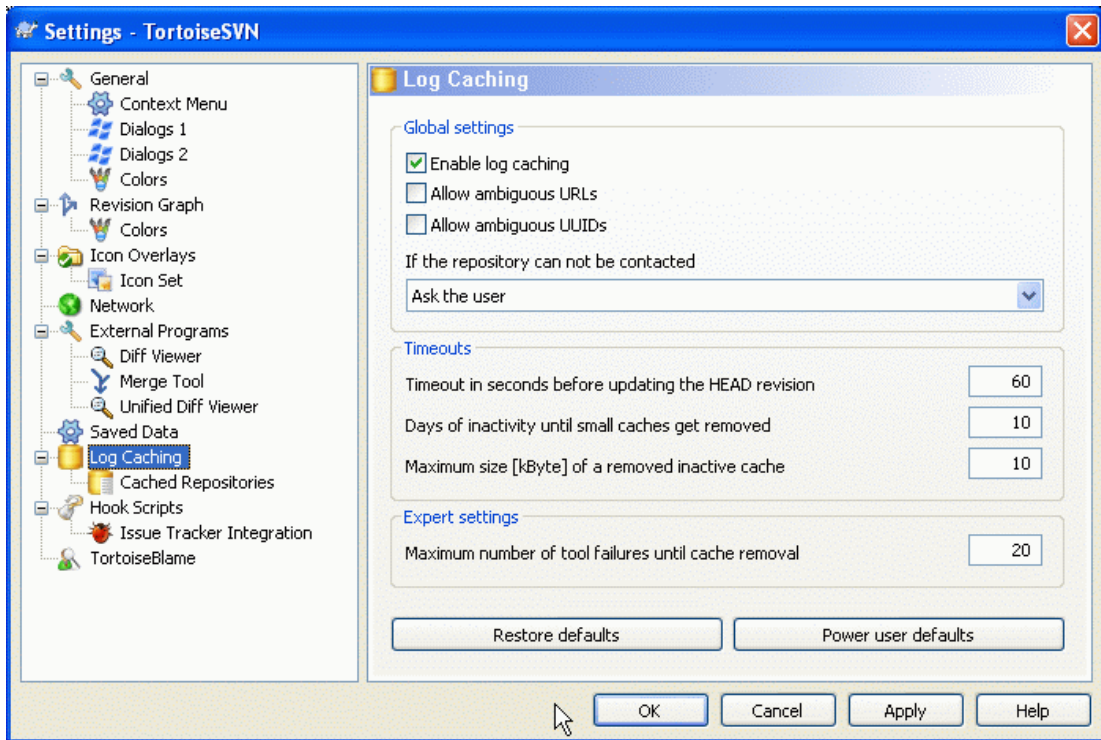


Figura 4.64. A Caixa de Diálogo Preferências, Página Cache de Registo

Esta caixa de diálogo permite-te configurar a funcionalidade de cache de registo do TortoiseSVN, que retém uma cópia local das mensagens de registo e caminhos alterados, para longas descargas a partir do servidor. O uso da cache de registos pode acelerar dramaticamente a caixa de diálogo de registo e o gráfico de revisões. Outra funcionalidade útil é que as mensagens de registo podem ainda ser acedidas em modo fora-de-linha.

Activar cache de registos

Activa a cache de registo sempre que forem requisitados dados de registo. Se verificada, os dados serão, quando disponíveis, adquiridos da cache e qualquer mensagem adquirida do servidor que não já lá esteja, será adicionada.

Se a cache estiver desactivada, os dados serão sempre adquiridos directamente do servidor, e não serão armazenados localmente.

Permitir URLs ambíguos

Ocasionalmente poderás ter de te ligar a um servidor que usa o mesmo URL para todos os repositórios. Versões antigas do `svnbridge` poderão ter este comportamento. Se necessitas aceder a tais repositórios terás de verificar esta opção. Se não, deixa-a como não verificada, para melhoria da performance.

Permitir UUIDs ambíguos

Alguns serviços de hospedagem dão a todos os seus repositórios o mesmo UUID. Poderás tu próprio o teres feito ao copiar uma pasta de repositório para criar um novo. Por todo tipo de razões, isto é uma má ideia - um UUID deve ser *único*. No entanto, a cache de registo poderá funcionar nesta situação se tu verificares esta caixa. Se não precisas dela, deixa-a não verificada de modo a melhorar a performance.

Se o repositório não puder ser contactado

Se estás a trabalhar fora-de-linha ou se o servidor do repositório estiver em baixo, a cache de registo poderá ser usada na mesma para fornecer as mensagens de registo já mantidas em cache. É claro que a cache poderá não estar actualizada, pelo que existem opções que te permitem seleccionar se esta funcionalidade deverá ser usada.

Quando os dados de registo estiverem a ser retirados da cache sem contactar o servidor, a caixa de diálogo que usará essas mensagens, irá mostrar o estado fora-de-linha na sua barra de título.

Prazo excedido antes de actualizar a revisão HEAD

Quando invocas a caixa de diálogo de registo, normalmente quererás contactar o servidor para verificar alguma nova mensagem de registo. Se o valor de prazo excedido configurado aqui é diferente de zero, então o servidor só será contactado quando o prazo foi excedido desde o último contacto. Isto pode reduzir as viagens de ida e volta ao servidor, se abrires a caixa de diálogo de registo com frequência e o servidor estiver em baixo, mas os dados aqui mostrados poderão não estar completamente actualizados. Se quiseres usar esta funcionalidade, sugerimos o uso do valor de 300 (5 minutos) como um compromisso.

Dias de inactividade até as caches pequenas serem removidas

Se navegares em muitos repositórios irás acumular muitas caches de registos. Se não as usas activamente, a cache não irá crescer muito pelo que o TortoiseSVN limpa-as após algum tempo configurado por defeito. Usa este item para controlar a limpeza da cache.

Tamanho máximo [kByte] de uma cache inactiva removida

Caches grandes são mais dispendiosas de adquirir pelo que o TortoiseSVN só limpa as pequenas. Afina o limite com este valor.

Número máximo de falhas de ferramentas até à remoção da cache

Ocasionalmente algo corre mal com a cache e é provocado um estoiro. Se isto acontecer a cache é normalmente automaticamente removida de modo a prevenir novas ocorrências do problema. Se usares as, menos estáveis, construções nocturnas poderás optar por manter a cache de qualquer modo.

4.30.7.1. Repositórios em Cache

Nesta página, tu podes ver a lista de repositórios que estão localmente em cache e o espaço usado na cache. Se seleccionares um dos repositórios poderás então usar os botões por debaixo da lista.

Clica no **Actualizar** para refrescar completamente a cache e preencher qualquer lacuna. Para um grande repositório isto poderá consumir muito tempo, mas será útil se estiveres prestes a entrar em modo fora-de-linha e queres a melhor cache disponível.

Clica no botão **Exportar** para exportar a cache inteira como um conjunto de ficheiros CSV. Isto poderá ser útil se quiseres processar os dados de registo usando um programa externo, embora seja maioritariamente útil aos programadores.

Clica no **Remover** para remover todos os dados em cache, para os repositórios seleccionados. Isto não desactiva a cache para o repositório, pelo que a próxima vez que requisitares dados de registo, a nova cache será criada.

4.30.7.2. Estatísticas da Cache de Registo

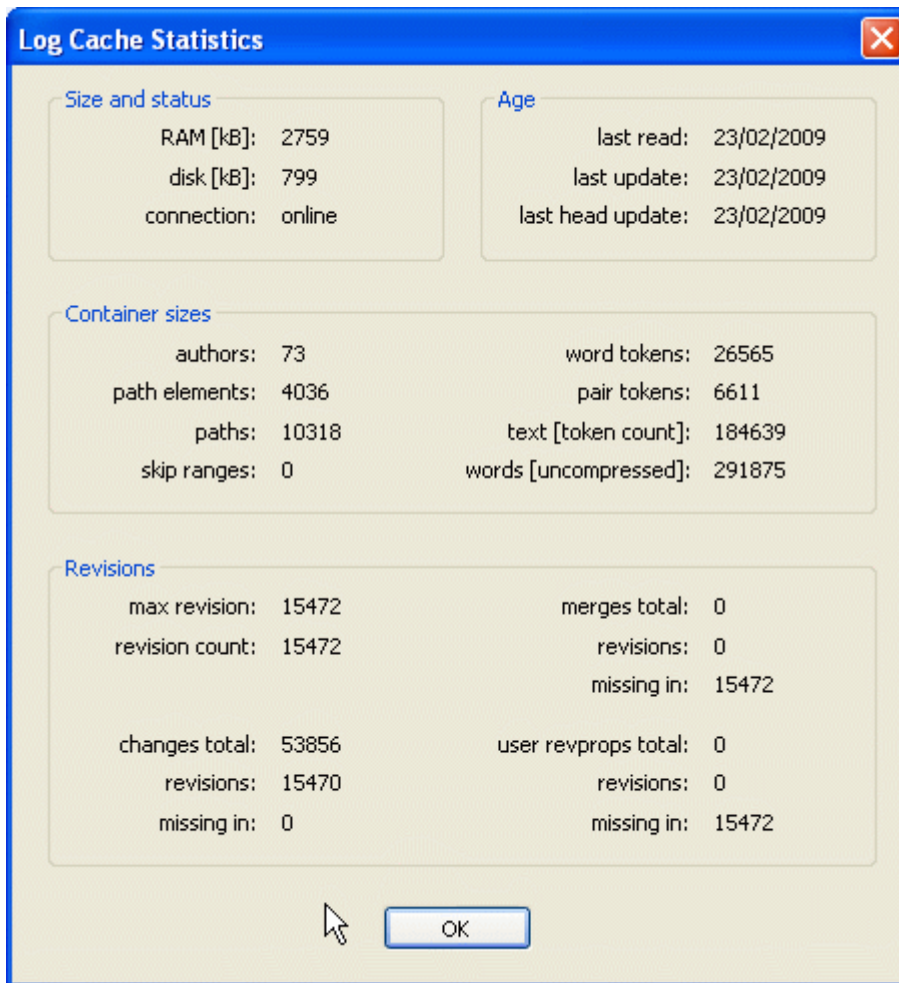


Figura 4.65. A Caixa de Diálogo Preferências, Estatísticas da Cache de Registo

Clica no botão **Pormenores** para veres as estatísticas detalhadas para uma cache em particular. Muitos dos campos aqui mostrados são de especial interesse apenas para os programadores do TortoiseSVN, pelo que não serão descritos em detalhe.

RAM

A quantidade de memória requerida para o serviço desta cache.

Disco

A quantidade de espaço em disco usado pela cache. Os dados são comprimidos pelo que a utilização de disco é geralmente razoavelmente modesta.

ligação

Mostra se o repositório estava disponível, da última vez que a cache foi usada.

última actualização

A última vez que o conteúdo da cache foi alterado.

última actualização da head

A última vez que requisitámos a revisão HEAD do servidor.

Autores

O número de diferentes autores com mensagens gravadas na cache.

Caminhos

O número de caminhos listados, como verias ao usar `svn log -v`.

Intervalos omitidos

O número de intervalos de revisões que não adquiridas porque não foram simplesmente requisitadas. Esta é uma medida do número de lacunas na cache.

Revisão máxima

O número de revisão mais alto armazenado na cache.

Número de revisões

O número de revisões armazenadas na cache. Esta é outra medida da completude da cache.

4.30.8. Scripts de Gancho do Lado do Cliente

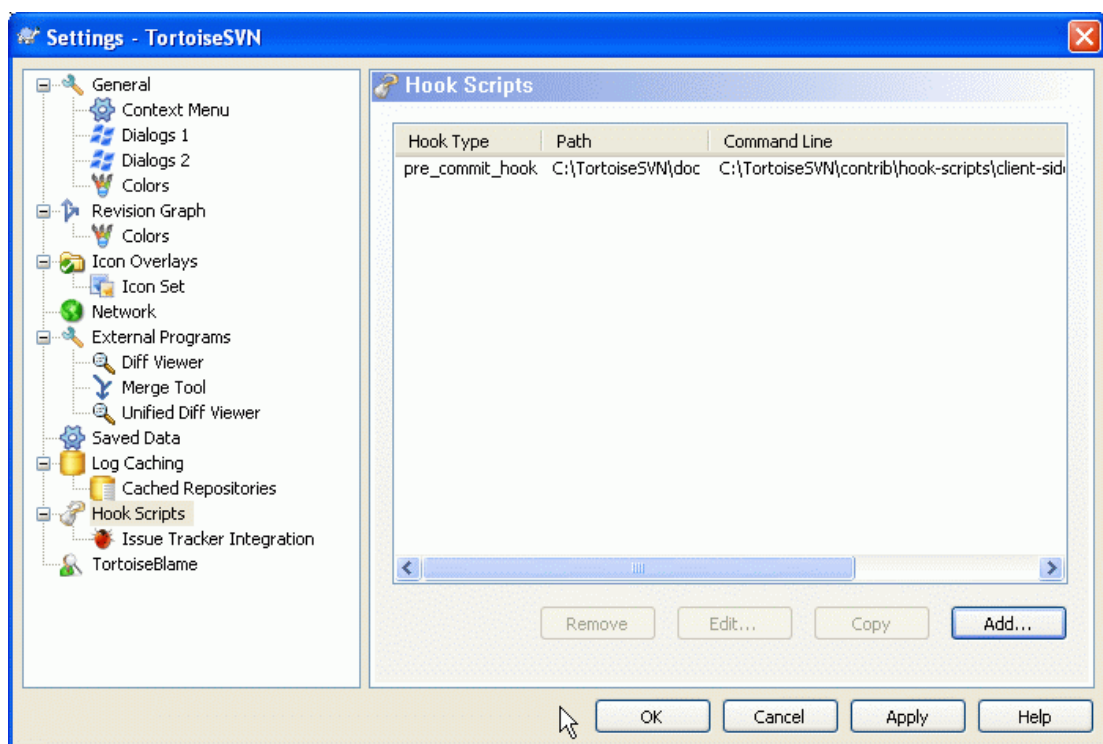


Figura 4.66. A Caixa de Diálogo Preferências, Página Scripts de Gancho

Esta caixa de diálogo permite-te configurar scripts de gancho, que serão executados automaticamente, quando certas acções do Subversion são executadas. Ao contrário dos scripts de gancho explicados em [Secção 3.3, “Scripts de gancho de servidor”](#), estes são executados localmente no cliente.

Uma aplicação para tais ganchos poderia ser, chamar um programa como `SubWCRev.exe` para actualizar os números de versão após uma submissão, e talvez disparar uma reconstrução.

Por várias razões de segurança e implementação, os scripts de gancho são definidos localmente na máquina, em vez de propriedades do projecto. Tu defines o que acontece independentemente do que alguém submete para o repositório. é claro que tu podes sempre escolher chamar um script que por sua vez está sob controlo de versões.

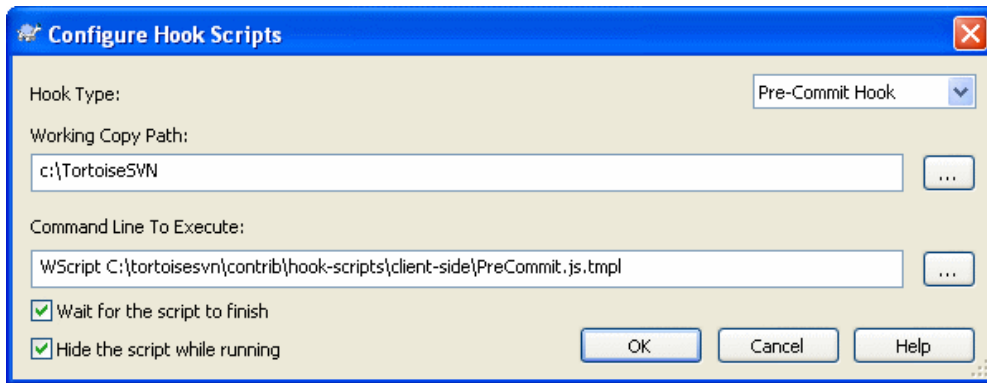


Figura 4.67. A Caixa de Diálogo Preferências, Configurar Scripts de Gancho

Para adicionar um novo script de gancho, clica simplesmente em Adicionar e preenche os detalhes.

Existem correntemente, seis tipos de scripts de gancho disponíveis

Início-submissão

Chamado antes de aparecer a caixa de diálogo submeter. Poderás querer usa-lo, se o gancho modificar um ficheiro versionado e afectar uma lista de ficheiros que necessitam de ser submetidos e/ou de uma mensagem de submissão. No entanto deverás ter em conta que, devido ao gancho ser chamado num estágio inicial, a lista completa de objectos seleccionados para submissão não está ainda disponível.

Pré-Submissão

Chamado depois de o utilizador clicar no OK na caixa de diálogo submeter, e antes de realmente começar a submissão. Este gancho tem a lista do que será exactamente submetido.

Pós-submissão

Chamado após a submissão finalizar (com sucesso ou não).

Início-actualização

Chamado antes da caixa de diálogo actualizar-para-revisão aparecer.

Pré-actualização

Chamado antes de efectivamente começar a actualização do Subversion.

Pós-actualização

Chamado após a actualização terminar (com sucesso ou não).

Um gancho é definido para um caminho em particular na cópia de trabalho. Tu só necessitas de especificar o caminho de nível superior; se executares uma operação numa subpasta, o TortoiseSVN irá procurar automaticamente para cima, por um caminho que corresponda.

A seguir, deverás especificar a linha de comando a executar, a começar pelo caminho para o script de gancho ou executável. Isto poderá ser um ficheiro batch, executável, ou outro qualquer ficheiro que tenha uma associação de ficheiro válida no Windows, e.g. um script de perl.

A linha de comando inclui vários parâmetros que serão preenchidos pelo TortoiseSVN. Os parâmetros passados dependem do gancho que for invocado. Cada gancho tem os seus próprios parâmetros, que são passados na seguinte ordem:

Início-submissão

PATHMESSAGEFILECWD

Pré-Submissão

PATHDEPTHMESSAGEFILECWD

Pós-submissão

PATHDEPTHMESSAGEFILEREVISIONERRORCWD

Início-actualização

PATHCWD

Pré-actualização

PATHDEPTHREVISIONCWD

Pós-actualização

PATHDEPTHREVISIONERRORCWD

O significado de cada um desses parâmetros é descrito aqui:

PATH

O caminho para um ficheiro temporário, que contém todos os caminhos para o qual a operação foi iniciada. Cada caminho está numa linha separada, no ficheiro temporário.

DEPTH

A profundidade com que a submissão/actualização é executada.

Valores possíveis são:

- 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

Caminho para um ficheiro que contém a mensagem de registo para a submissão. O ficheiro contém texto codificado em UTF-8. Após execução com sucesso do gancho Início-de-submissão, a mensagem de registo é novamente lida, dando ao gancho a oportunidade de modificá-la.

REVISION

A revisão do repositório para a qual a actualização deverá ser feita, ou a revisão após a conclusão de uma submissão.

ERROR

Caminho para um ficheiro contendo a mensagem de erro. Se não houve nenhum erro, o ficheiro estará vazio.

CWD

A pasta de trabalho corrente com a qual o script está a correr. Esta é configurada para a pasta raiz comum a todos os caminhos afectados.

Ter em conta que apesar de termos dado nomes por conveniência a esses parâmetros, tu não terás de te referir a esses nomes nas configurações dos ganchos. Todos os parâmetros listados para um gancho em particular são sempre passados, quer os queiras ou não ;-)

Se queres que ao funcionamento do Subversion espere até a execução do gancho finalizar, verifica Espere que o script termine.

Normalmente tu quererás esconder as caixas feias de DOS, enquanto o script corre, então a Ocultar o script durante a sua execução está verificada por defeito.

Exemplos de scripts de gancho de cliente podem ser encontrados na pasta contrib no repositório *TortoiseSVN* [<http://tortoisesvn.googlecode.com/svn/trunk/contrib/hook-scripts>]. (Secção 3, “O TortoiseSVN é grátis!” explica como aceder ao repositório).

4.30.8.1. Integração do Controlador de Problemas

O TortoiseSVN pode usar um plugin COM para consultar os controladores de problemas quando estiver na caixa de diálogo submiter. O uso de tais plugins está descrito em Secção 4.28.2, “Obter Informações do Gestor de Problemas”. Se o teu administrador de sistemas forneceu-te um plugin, que já instalaste e registaste, aqui é o local para especificares como ele se integra com a tua cópia de trabalho.

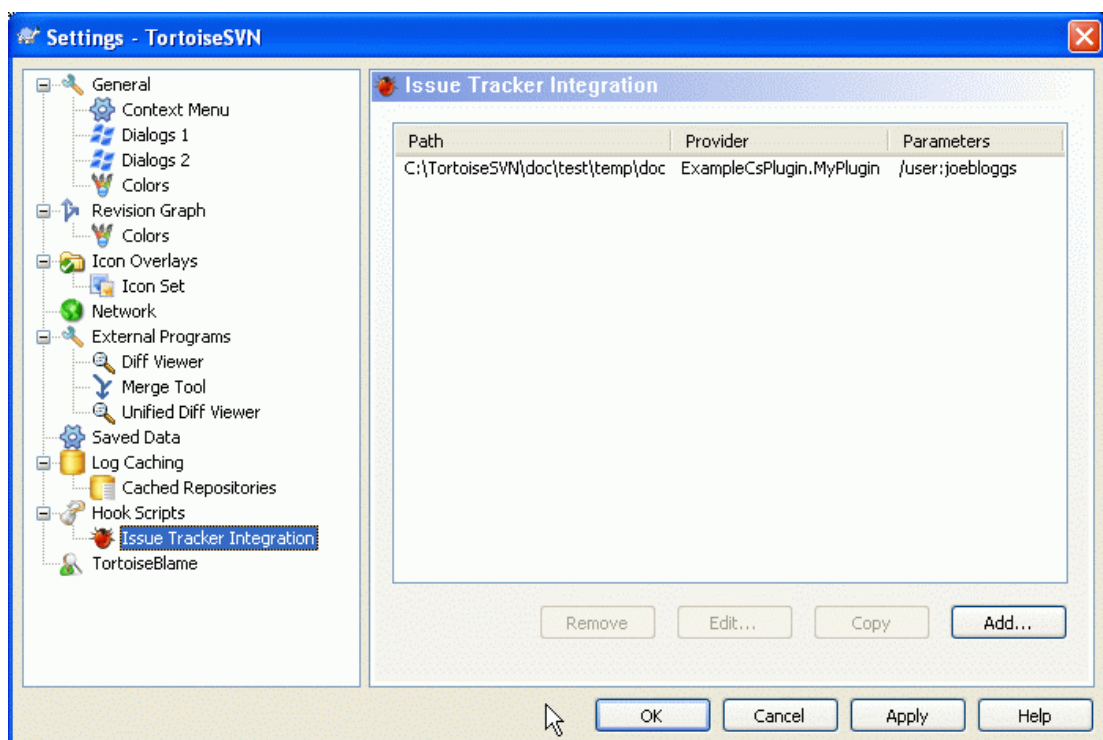


Figura 4.68. A Caixa de Diálogo Preferências, Página Integração com Controlador de Problemas

Clica no Adicionar... para utilizar o plugin com uma cópia de trabalho em particular. Aqui podes especificar o caminho da cópia de trabalho, escolher qual o plugin a usar a partir de um lista drop-down com todos os controladores de problemas registados, e qualquer parâmetro a passar. Os parâmetros serão específicos ao plugin, mas poderão incluir o teu nome de utilizador no controlador de problemas, para que o plugin possa consultar os problemas que te estão atribuídos.

Se queres que todos os utilizadores usem o mesmo plugin COM no teu projecto, podes especificar no plugin as propriedades bugtraq:provideruuid e bugtraq:providerparams.

bugtraq:provideruuid

Esta propriedade especifica o UUID COM do IBugtraqProvider, por exemplo {91974081-2DC7-4FB1-B3BE-0DE1C8D6CE4E}. (Neste exemplo é o UUID do *fornecedor Gurtle bugtraq* [<http://code.google.com/p/gurtle/>], que é o fornecedor para o controlador de problemas *Google Code* [<http://code.google.com/hosting/>]).

bugtraq:providerparams

Esta propriedade especifica os parâmetros passados para o IBugtraqProvider.

Consulta por favor a documentação do teu plugin IBugtraqProvider, de modo a descobrires o que especificar nessas duas propriedades.

4.30.9. Preferências do TortoiseBlame

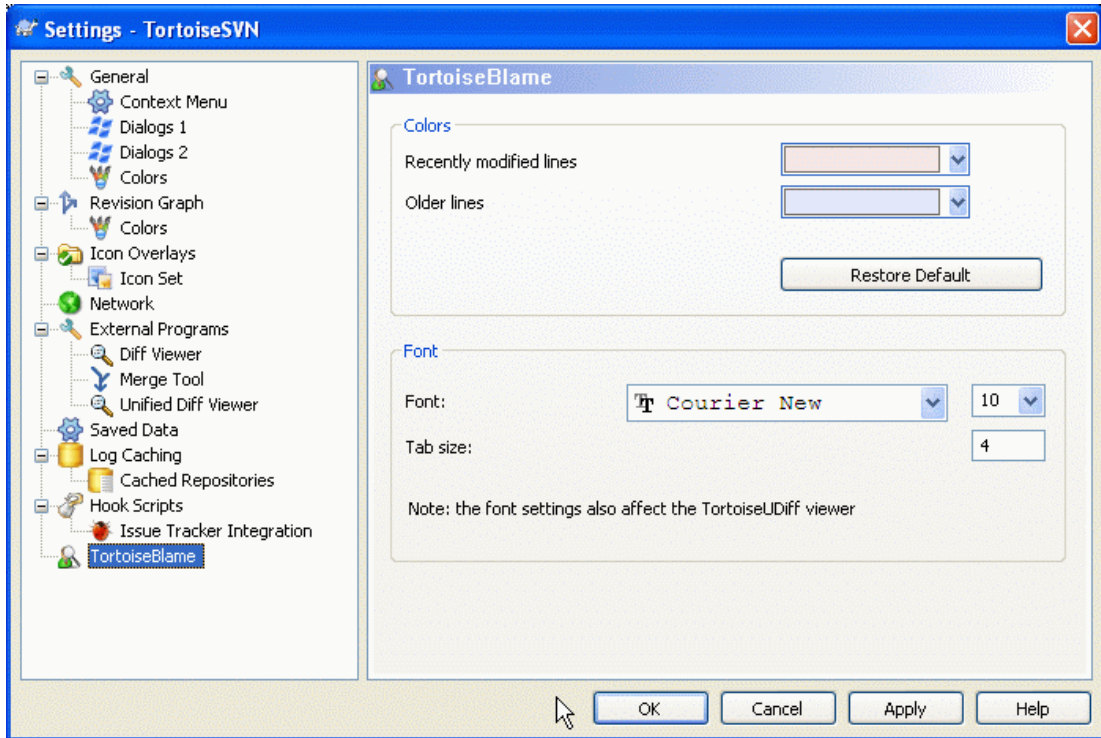


Figura 4.69. A Caixa de Diálogo Preferências, Página do TortoiseBlame

As preferências usadas pelo TortoiseBlame são controladas a partir do menu de contexto principal, e não directamente com o próprio TortoiseBlame.

Cores

O TortoiseBlame pode usar cores de fundo para indicar a idade das linhas num ficheiro. Tu configuras as pontas ao especificar as cores para a revisão mais nova e para a mais velha, e o TortoiseBlame usa uma interpolação linear entre essas duas cores de acordo com a revisão do repositório, indicada para cada linha.

Tipo de letra

Tu podes seleccionar o tipo de letra usada para mostrar o texto e também o tamanho do ponto a usar. Isto aplica-se também ao conteúdo do ficheiro, ao autor e informação de revisão mostrada no painel esquerdo.

Tabuladores

Define quantos espaços deverão ser usados na expansão, quando um caracter de tabulação for encontrado no conteúdo do ficheiro.

4.30.10. Configurações do Registo

Algumas, não muito usadas, configurações de utilizador apenas estão disponíveis através da edição directa do registo. No entanto deverá ser salientado que apenas deverás editar o registo se souberes exactamente o que estás a fazer.

Configuração

Poderás especificar diferentes localizações para o ficheiro de configuração do Subversion, usando a localização do registo `HKCU\Software\TortoiseSVN\ConfigDir`. Isto irá afectar todas as operações do TortoiseSVN.

ícone do tabuleiro de cache

To add a cache tray icon for the TSVNCache program, create a `DWORD` key with a value of 1 at `HKCU\Software\TortoiseSVN\CacheTrayIcon`. This is really only useful for developers as it allows you to terminate the program gracefully.

Depurar

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`.

ícones de Menu de Contexto

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.

Bloco de sobreposição de estado

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`.

URL de verificação de actualização

`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.

Nomes de ficheiros sem extensão na lista de auto-completar

The auto-completion list shown in the commit message editor displays the names of files listed for commit. To also include these names with extensions removed, create a `DWORD` key with a value of 1 at `HKCU\Software\TortoiseSVN\AutocompleteRemovesExtensions`.

colunas do Explorador em todo o lado

The extra columns the TortoiseSVN adds to the details view in Windows Explorer are normally only active in a working copy. If you want those to be accessible everywhere, not just in working copies, create a `DWORD` key with a value of 1 at `HKCU\Software\TortoiseSVN\ColumnsEveryWhere`.

Separador de registo de Integração

When you merge revisions from another branch, and merge tracking information is available, the log messages from the revisions you merge will be collected to make up a commit log message. A pre-defined string is used to separate the individual log messages of the merged revisions. If you prefer, you can create a `SZ` key at `HKCU\Software\TortoiseSVN\MergeLogSeparator` containing a separator string of your choice.

Responsabiliza sempre as alterações com o TortoiseMerge

TortoiseSVN allows you to assign external diff viewer. Most such viewers, however, are not suited for change blaming ([Secção 4.23.2, “Diferenças de responsabilidade”](#)), so you might wish to fall back to TortoiseMerge in this case. To do so, create a `DWORD` key with a value of 1 at `HKCU\Software\TortoiseSVN\DiffBlamesWithTortoiseMerge`.

Salientar a revisão corrente para pastas, ca caixa de diálogo de registo

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.

Assegura a falha do checkout se existe um item com o mesmo nome

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. Pastas de Trabalho do Subversion

O VS.NET 2003 quando usado com projectos web não consegue lidar com as pastas `.svn` que o Subversion usa para armazenar a sua informação interna. Isto não é um bug do Subversion. O bug é do VS.NET 2003 e das extensões frontpage que ele usa.

Ter em atenção que o bug está reparado no VS2005 e posteriores versões.

A partir da versão 1.3.0 do Subversion e TortoiseSVN, tu podes configurar a variável de ambiente `SVN_ASP_DOT_NET_HACK`. Se essa variável estiver configurada, o Subversion irá usar as pastas `_svn` em vez das `.svn`. Tu terás de reiniciar a tua shell para que essa variável de ambiente tome efeito. Normalmente isso significa reiniciar o PC. Para tornar isto mais fácil, tu podes fazer isto agora a partir da página de preferências gerais, usando uma simples caixa de verificação - consulta [Secção 4.30.1, "Preferências Gerais"](#).

Para mais informações, e outras formas de evitar este problema em primeiro lugar, verifica o artigo sobre este assunto nas nossas [FAQs](http://tortoisesvn.net/aspdotnethack) [http://tortoisesvn.net/aspdotnethack].

4.31. Passo Final

Doa!

Mesmo que o TortoiseSVN e o TortoiseMerge sejam gratuitos, tu podes apoiar os programadores enviando correcções e tomando uma papel activo no desenvolvimento. Podes também ajudar a alegrar-nos durante as intermináveis horas que passamos em frente dos nossos computadores.

Enquanto trabalhamos no TortoiseSVN adoramos ouvir música. E visto que passamos tantas horas no projecto, precisamos de *muita* música. Por isso criámos umas listas de desejos, com os nossos CDs e DVDs de música favoritos : <http://tortoisesvn.tigris.org/donate.html> Por favor olha também para a lista de pessoas que contribuíram para o projecto ao enviar correcções ou traduções.

Capítulo 5. O Programa SubWCRev

O SubWCRev é um programa de consola do Windows que pode ser usado para ler o estado da cópia de trabalho do Subversion, e opcionalmente executar substituição de palavras-chave num ficheiro template. Isto é frequentemente usado como parte do processo de construção, como meio de incorporar informação da cópia de trabalho no objecto que estás a construir. Poderá ser usado tipicamente para incluir o número de revisão numa caixa “Sobre”.

5.1. A Linha de Comando SubWCRev

O SubWCRev, por defeito, lê o estado Subversion de todos os ficheiros numa cópia de trabalho, incluindo os externos. Ele grava o número da mais alta revisão submetida que foi encontrada e a marca temporal dessa revisão, também grava se existem alterações locais na cópia de trabalho ou misturadas, revisões actualizadas.

O SubWCRev.exe é chamado a partir da linha de comando ou de um script e, é configurado através de parâmetros na linha de comandos.

```
SubWCRev WorkingCopyPath [SrcVersionFile DstVersionFile] [-nmdfe]
```

O `WorkingCopyPath` representa o caminho para a cópia de trabalho a ser verificada. Tu só podes usar o SubWCRev em cópias de trabalho e não directamente no repositório. O caminho pode ser absoluto ou relativo à pasta da cópia de trabalho corrente.

Se queres que o SubWCRev execute substituição de palavras-chave, para que campos como a revisão do repositório e URL sejam gravados num ficheiro de texto, irás necessitar de fornecer um ficheiro template `SrcVersionFile` e um ficheiro de destino `DstVersionFile` que contém a versão substituída do template.

Existem um número de opções que afectam a maneira como o SubWCRev funciona. Se usares mais do que um, eles devem ser especificados como um único grupo, e.g. `teral>-nm`

5.2. Substituição de Palavra-Chave

Se forem fornecidos os ficheiros de origem e destino, o SubWCRev copia da fonte para o destino, efectuando a substituição de palavras-chave da seguinte forma:

Palavra-chave	Descrição
<code>\$WCREV\$</code>	Substituída pela mais alta revisão de submissão da cópia de trabalho.
<code>\$WCDATE\$</code>	Substituída pela data/tempo de submissão da mais alta revisão submetida. Por defeito é usado o formato internacional: <code>yyyy-mm-dd hh:mm:ss</code> . Em alternativa poderás especificar um formato personalizado que poderá ser usado com <code>strftime()</code> , por exemplo: <code>\$WCDATE=%a %b %d %I:%M:%S %p\$</code> . Para uma lista de caracteres de formatação disponíveis, consulta o referência em linha [http://www.cppreference.com/stddate/strftime.html].
<code>\$WCNOW\$</code>	Substituída pela data/tempo corrente do sistema. Isto poderá ser usado para indicar a altura de construção. A formatação do tempo pode ser usada como descrita para o <code>\$WCDATE\$</code> .
<code>\$WCRANGE\$</code>	Substituída pelo intervalo de revisões actualizada na cópia de trabalho. Se a cópia de trabalho está num estado inconsistente, isto poderá ser apenas uma revisão. Se a cópia de trabalho contem revisões misturadas, seja devido a estar desactualizada, ou devido a uma actualização para revisão deliberada, então o intervalo de revisões será mostrado no formato <code>100:200</code>

Palavra-chave	Descrição
\$WCMIXED\$	\$WCMIXED?TText:FText\$ é substituída por TText se existem revisões de actualização misturadas, ou FText se não for o caso.
\$WCMODS\$	\$WCMODS?TText:FText\$ é substituída por TText se existem alterações locais, ou FText se não for o caso.
\$WCURL\$	Substituída pelo URL do repositório correspondente ao caminho da cópia de trabalho passada ao SubWCRev.
\$WCINSVN\$	\$WCINSVN?TText:FText\$ é substituída pelo TText se a entrada é versionada, ou FText se não o é.
\$WCNEEDSLOCK\$	\$WCNEEDSLOCK?TText:FText\$ é substituída por TText se a entrada tiver a propriedade svn:needs-lock configurada, ou por FText se não for o caso.
\$WCISLOCKED\$	\$WCISLOCKED?TText:FText\$ é substituída por TText se a entrada estiver bloqueada, ou por FText se não o estiver.
\$WCLOCKDATE\$	Substituída pela data de bloqueio. A formatação do tempo poderá ser usada de acordo com o descrito para o \$WCDATE\$.
\$WCLOCKOWNER\$	Substituída pelo nome do autor do bloqueio.
\$WCLOCKCOMMENT\$	Substituída pelo comentário do bloqueio.

Tabela 5.1. Lista de opções de linha de comando disponíveis



Dica

Algumas dessas palavras-chave só se aplicam a um único ficheiro, em vez de a toda a cópia de trabalho, pelo que só fará sentido usá-los quando o SubWCRev for chamado a examinar um único ficheiro. Isto aplica-se ao \$WCINSVN\$, \$WCNEEDSLOCK\$, \$WCISLOCKED\$, \$WCLOCKDATE\$, \$WCLOCKOWNER\$ e \$WCLOCKCOMMENT\$.

5.3. Exemplo de Palavra-Chave

O exemplo abaixo mostra como palavras-chave num ficheiro template são substituídas no ficheiro de saída.

```
// Ficheiro de teste para o SubWCRev: testfile.tpl

char *Revision = "$WCREV$";
char *Modified = "$WCMODS?Modificado:Não modificado$";
char *Date = "$WCDATE$";
char *Range = "$WCRANGE$";
char *Mixed = "$WCMIXED?Revisão misturada CT:Não misturada$";
char *URL = "$WCURL$";

#if $WCMODS?1:0$
#error A fonte está modificada
#endif

// Fim de ficheiro

Depois de correr o SubWCRev.exe path\to\workingcopy testfile.tpl
testfile.txt, o ficheiro de saída testfile.txt irá ficar assim:

// Ficheiro de teste para o SubWCRev: testfile.txt
```

```

char *Revision = "3701";
char *Modified = "Modificado";
char *Date     = "2005/06/15 11:15:12";
char *Range   = "3699:3701";
char *Mixed    = "Revisão misturada CT";
char *URL     = "http://project.domain.org/svn/trunk/src";

#if 1
#error A fonte está modificada
#endif

//Fim de ficheiro
    
```



Dica

Um ficheiro como este será incluído na construção, pelo que tu esperas que ele esteja sob controlo de versões. Tem a certeza de versionares o ficheiro template e não o gerado, de outra maneira de cada vez que regenerares o ficheiro de versão, terás de submeter a alteração o que por sua vez significa que o ficheiro de versão necessita então de ser actualizado.

5.4. Interface COM

Se necessitas de aceder à informação de revisão do Subversion a partir de outros programas, podes usar a interface COM do SubWCRev. O objecto a criar é o `SubWCRev.object` e os métodos suportados são os seguintes:

Método	Descrição
.GetWCInfo	Este método percorre a cópia de trabalho, reunindo a informação de revisão. Naturalmente, deverás chamar este método antes de poderes aceder à informação usando os restantes métodos. O primeiro parâmetro é o caminho. O segundo parâmetro deverá ser true se queres incluir as revisões das pastas. Equivalente à opção de linha de comando -f. O terceiro parâmetro deverá ser true se queres incluir svn:externals. Equivalente à opção de linha de comando -e.
.Revision	A mais alta revisão de submissão na cópia de trabalho. Equivalente ao \$WCREV\$
.Date	A data/tempo de submissão da mais alta revisão de submissão. Equivalente ao \$WCDATE\$
.Author	O autor da mais alta revisão de submissão, isto é, a última pessoa que submeteu alterações na cópia de trabalho.
.MinRev	A revisão de actualização mínima, como mostrada em \$WCRANGE\$
.MaxRev	A revisão de actualização máxima, como mostrado em \$WCRANGE\$
.HasModifications	True se existem modificações locais
.Url	Substituído com o URL do repositório, do caminho de cópia de trabalho, usado em GetWCInfo. Equivalente a \$WCURL\$
.IsSvnItem	True se o item é versionado.
.NeedsLocking	True se o item tem a propriedade svn:needs-lock configurada.
.IsLocked	True se o item está bloqueado.
.LockCreationDate	String que representa a data em que o bloqueio foi criado, ou uma string vazia se o item não está bloqueado.

Método	Descrição
.LockOwner	String que representa o autor do bloqueio, ou uma string vazia se o item não está bloqueado.
.LockComment	A mensagem introduzida quando o bloqueio foi criado.

Tabela 5.2. Métodos COM/automação suportados

O seguinte exemplo mostra como a interface pode ser usada.

```
// testCOM.js - ficheiro javascript
//script de testepara o 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 +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
"
Min Revision = " + revObject1.MinRev +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
"
Max Revision = " + revObject1.MaxRev +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
"
Date = " + revObject1.Date +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
"
URL = " + revObject1.Url + "
Author = " +
<span>></span>
<span>></span>
```

```
<span>></span>
<span>></span>
revObject1.Author + "
HasMods = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject1.HasModifications + "
IsSvnItem = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject1.IsSvnItem + "
NeedsLocking = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject1.NeedsLocking + "
IsLocked = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject1.IsLocked + "
LockCreationDate = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject1.LockCreationDate + "
LockOwner = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject1.LockOwner + "
LockComment = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject1.LockComment;
wcInfoString2 = "Revision = " + revObject2.Revision +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
"
Min Revision = " + revObject2.MinRev +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
"
Max Revision = " + revObject2.MaxRev +
```

```
<span>></span>
<span>></span>
<span>></span>
<span>></span>
"
Date = " + revObject2.Date +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
"
URL = " + revObject2.Url + "
Author = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject2.Author + "
HasMods = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject2.HasModifications + "
IsSvnItem = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject2.IsSvnItem + "
NeedsLocking = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject2.NeedsLocking + "
IsLocked = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject2.IsLocked + "
LockCreationDate = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject2.LockCreationDate + "
LockOwner = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject2.LockOwner + "
LockComment = " +
<span>></span>
<span>></span>
<span>></span>
```



```
<span>></span>
revObject2.LockComment;
wcInfoString3 = "Revision = " + revObject3.Revision +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
"
Min Revision = " + revObject3.MinRev +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
"
Max Revision = " + revObject3.MaxRev +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
"
Date = " + revObject3.Date +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
"
URL = " + revObject3.Url + "
Author = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject3.Author + "
HasMods = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject3.HasModifications + "
IsSvnItem = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject3.IsSvnItem + "
NeedsLocking = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject3.NeedsLocking + "
IsLocked = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject3.IsLocked + "
LockCreationDate = " +
```

```
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject3.LockCreationDate + "
LockOwner = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject3.LockOwner + "
LockComment = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject3.LockComment;
wcInfoString4 = "Revision = " + revObject4.Revision +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
"
Min Revision = " + revObject4.MinRev +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
"
Max Revision = " + revObject4.MaxRev +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
"
Date = " + revObject4.Date +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
"
URL = " + revObject4.Url + "
Author = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject4.Author + "
HasMods = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject4.HasModifications + "
IsSvnItem = " +
<span>></span>
<span>></span>
<span>></span>
```

```
<span>></span>
revObject4.IsSvnItem + "
NeedsLocking = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject4.NeedsLocking + "
IsLocked = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject4.IsLocked + "
LockCreationDate = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject4.LockCreationDate + "
LockOwner = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject4.LockOwner + "
LockComment = " +
<span>></span>
<span>></span>
<span>></span>
<span>></span>
revObject4.LockComment;

WScript.Echo(wcInfoString1);
WScript.Echo(wcInfoString2);
WScript.Echo(wcInfoString3);
WScript.Echo(wcInfoString4);
```

Capítulo 6. Interface IBugtraqProvider

Para obter uma integração mais íntima com os seguidores de problemas, do que usar simplesmente as propriedades `bugtraq:`, o TortoiseSVN pode fazer uso de plugins COM. Com tais plugins é possível adquirir informação directamente do seguidor de problemas, interagir com o utilizador e providenciar informação de volta ao TortoiseSVN sobre problemas abertos, verificar mensagens de registo introduzidas pelo utilizador, ou mesmo correr acções, após uma submissão com sucesso, para e.g. fechar um problema.

Nós não podemos fornecer informação e tutoriais em como implementar um objecto COM na tua linguagem de programação preferida, mas temos exemplos de plugins em C++/ATL e C#, no nosso repositório na pasta `contrib/issue-tracker-plugins`. Nessa pasta poderás também encontrar requeridos ficheiros de include que necessitas para construir o teu plugin. . (Secção 3, “O TortoiseSVN é grátis!” explica como aceder ao repositório).

6.1. O interface do IBugtraqProvider

O TortoiseSVN 1.5 pode usar plugins que implementam a interface IBugtraqProvider. A interface providencia alguns métodos que os plugins podem usar para interagir com o gestor de problemas.

```
HRESULT ValidateParameters (
    // Janela Pai para cada UI que necessita de ser
    // mostrada durante a validação.
    [in] HWND hParentWnd,

    // O parametro de string que necessita de ser validada.
    [in] BSTR parameters,

    // A string é valida?
    [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 (
    // Janela pai para qualquer (erro) de UI que necessite de ser mostrado.
    [in] HWND hParentWnd,

    // O parametro string, para o caso de necessitares de falar com o teu
    // web service (e.g.) para descobrires qual é o texto correcto.
    [in] BSTR parameters,

    // Que texto queres mostrar?
    // Use the current thread locale.
    [out, retval] BSTR *linkText
);
```

O plugin pode fornecer uma string que será usada na caixa de diálogo Submeter, para o botão que invoca o plugin. E.g., "Escolhe o problema" ou "Selecciona o bilhete". Confirma se a string não é muito longa,

de outro modo poderá não caber no botão. Se o método retornar um erro (e.g., E_NOTIMPL), é usado um texto por defeito para o botão.

```
HRESULT GetCommitMessage (
    // Janela pai para o UI do teu fornecedor.
    [in] HWND hParentWnd,

    // Parâmetros para o teu fornecedor.
    [in] BSTR parameters,
    [in] BSTR commonRoot,
    [in] SAFEARRAY(BSTR) pathList,

    // O texto já presente na mensagem de registo.
    // O teu fornecedor deverá incluir este texto na nova mensagem,
    // quando apropriado.
    [in] BSTR originalMessage,

    // O novo texto para a mensagem de registo.
    // Isto substituirá a mensagem original.
    [out, retval] BSTR *newMessage
);
```

This is the main method of the plugin. This method is called from the TortoiseSVN commit dialog when the user clicks on the plugin button. The `parameters` string is the string the user has to enter in the settings dialog when he configures the plugin. Usually a plugin would use this to find the URL of the issue tracker and/or login information or more. The `commonRoot` string contains the parent path of all items selected to bring up the commit dialog. Note that this is *not* the root path of all items which the user has selected in the commit dialog. The `pathList` parameter contains an array of paths (as strings) which the user has selected for the commit. The `originalMessage` parameter contains the text entered in the log message box in the commit dialog. If the user has not yet entered any text, this string will be empty. The `newMessage` return string is copied into the log message edit box in the commit dialog, replacing whatever is already there. If a plugin does not modify the `originalMessage` string, it must return the same string again here, otherwise any text the user has entered will be lost.

6.2. A interface IBugtraqProvider2

No TortoiseSVN 1.6 foi adicionada uma nova interface que providência mais funcionalidades aos plugins. Esta interface `IBugtraqProvider2` herda da `IBugtraqProvider`.

```
HRESULT GetCommitMessage2 (
    // Janela pai para o UI do teu fornecedor.
    [in] HWND hParentWnd,

    // Parametros do teu fornecedor.
    [in] BSTR parameters,
    // O URL comum da submissão
    [in] BSTR commonURL,
    [in] BSTR commonRoot,
    [in] SAFEARRAY(BSTR) pathList,

    // O Texto já presente na tua mensagem de submissão.
    // O teu fornecedor deverá incluir este texto na nova mensagem,
    // quando apropriado.
    [in] BSTR originalMessage,

    //Tu podes atribuir a uma submissão propriedades de revisão personalizadas
    // através da configuração dos dois próximos parâmetros.
```

```

// nota: Ambos os safearrays deverão ser do mesmo comprimento.
//      Para cada nome de propriedade deverá existir um valor de propriedade!

// O conteúdo do campo bugID (se mostrado)
[in] BSTR bugID,

// Conteúdo modificado do campo bugID
[out] BSTR * bugIDOut,

// A lista de nomes das propriedades de revisão.
[out] SAFEARRAY(BSTR) * revPropNames,

// A lista de valores de propriedades de revisão.
[out] SAFEARRAY(BSTR) * revPropValues,

// O novo texto para a mensagem de submissão.
// Isto substitui a mensagem original
[out, retval] BSTR * newMessage
);

```

This method is called from the TortoiseSVN commit dialog when the user clicks on the plugin button. This method is called instead of `GetCommitMessage()`. Please refer to the documentation for `GetCommitMessage` for the parameters that are also used there. The parameter `commonURL` is the parent URL of all items selected to bring up the commit dialog. This is basically the URL of the `commonRoot` path. The parameter `bugID` contains the content of the bug-ID field (if it is shown, configured with the property `bugtraq:message`). The return parameter `bugIDOut` is used to fill the bug-ID field when the method returns. The `revPropNames` and `revPropValues` return parameters can contain name/value pairs for revision properties that the commit should set. A plugin must make sure that both arrays have the same size on return! Each property name in `revPropNames` must also have a corresponding value in `revPropValues`. If no revision properties are to be set, the plugin must return empty arrays.

```

HRESULT CheckCommit (
    [in] HWND hParentWnd,
    [in] BSTR parameters,
    [in] BSTR commonURL,
    [in] BSTR commonRoot,
    [in] SAFEARRAY(BSTR) pathList,
    [in] BSTR commitMessage,
    [out, retval] BSTR * errorMessage
);

```

This method is called right before the commit dialog is closed and the commit begins. A plugin can use this method to validate the selected files/folders for the commit and/or the commit message entered by the user. The parameters are the same as for `GetCommitMessage2()`, with the difference that `commonURL` is now the common URL of all *checked* items, and `commonRoot` the root path of all checked items. The return parameter `errorMessage` must either contain an error message which TortoiseSVN shows to the user or be empty for the commit to start. If an error message is returned, TortoiseSVN shows the error string in a dialog and keeps the commit dialog open so the user can correct whatever is wrong. A plugin should therefore return an error string which informs the user *what* is wrong and how to correct it.

```

HRESULT OnCommitFinished (
    // Janela pai para qualquer (erro) UI que precisa de ser mostrado.
    [in] HWND hParentWnd,

    // A raiz comum para todos os caminhos que foram submetidos.

```

```
[in] BSTR commonRoot,  
  
// Todos os caminhos que foram submetidos.  
[in] SAFEARRAY(BSTR) pathList,  
  
// O texto já presente na mensagem de submissão.  
[in] BSTR logMessage,  
  
// A revisão da submissão.  
[in] ULONG revision,  
  
// Um erro a mostrar ao utilizador se esta função  
// retorna algo para além de S_OK  
[out, retval] BSTR * error  
);
```

Este método é chamado após uma submissão com sucesso. Um plugin pode usar este método para e.g.; fechar o problema seleccionado ou adicionar informação para o problema sobre a submissão. Os parâmetros são os mesmos do GetCommitMessage2.

```
HRESULT HasOptions(  
    // Para quando o fornecedor providenciar opções  
    [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(  
    // Janela pai para a caixa de diálogo opções  
    [in] HWND hParentWnd,  
  
    // Parâmetros para o teu fornecedor.  
    [in] BSTR parameters,  
  
    // A string parâmetros  
    [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.

Apêndice A. Questões Mais Frequentes (FAQ)

Porque o TortoiseSVN está a ser desenvolvido a todo o momento, é por vezes difícil manter a documentação completamente actualizada. Nós mantemos uma *FAQ em linha* [<http://tortoisesvn.tigris.org/faq.html>] que contém a selecção das questões que nos são perguntadas na maioria das listas de correio do TortoiseSVN, <dev@tortoisesvn.tigris.org> e <users@tortoisesvn.tigris.org>.

Nós também mantemos um projecto *Issue Tracker* [<http://issues.tortoisesvn.net>] que te pode dizer sobre algumas das coisas que nós temos nas nossas listas de afazeres, e bugs que já foram resolvidos. Se pensas que encontraste um bug, ou queres pedir uma nova funcionalidade, verifica primeiro aqui para veres se alguém já antes teve a mesma ideia.

If you have a question which is not answered anywhere else, the best place to ask it is on one of the mailing lists. <users@tortoisesvn.tigris.org> is the one to use if you have questions about using TortoiseSVN. If you want to help out with the development of TortoiseSVN, then you should take part in discussions on <dev@tortoisesvn.tigris.org>.

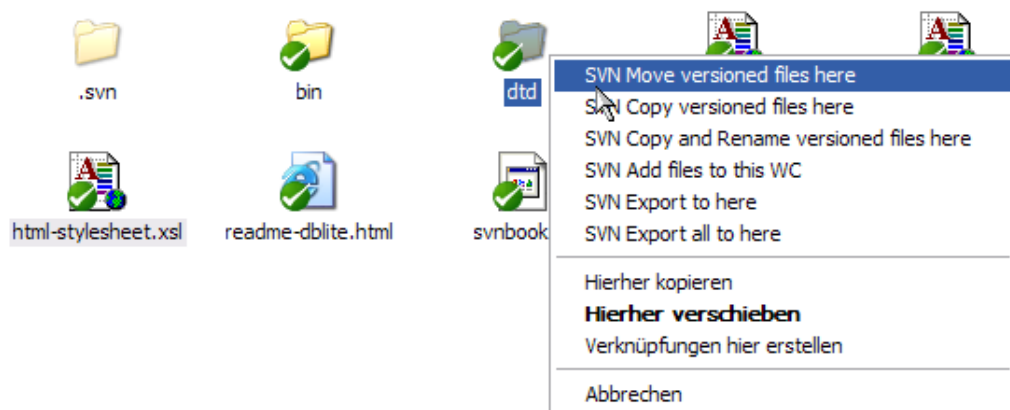
Apêndice B. Como Farei Para...

Este apêndice contém soluções para problemas/questões que poderás ter tido ao usar o TortoiseSVN.

B.1. Mover/copiar muitos ficheiros de uma vez só

Mover/Copiar ficheiros simples poderá ser feito usando o TortoiseSVN → Renomear.... Mas se pretendes mover/copiar muitos ficheiros, desta maneira será muito lento e trabalhoso.

A forma recomendada será através do arrastar com o botão direito dos ficheiros para uma nova localização. Simplesmente clica com o botão direito nos ficheiros que queres mover/copiar, sem largar o botão do rato. Então arrasta os ficheiros para uma nova localização e então larga o botão do rato. O menu de contexto irá aparecer, onde poderás escolher Menu de Contexto → SVN Copiar para aqui arquivos com controlo de versões ou Menu de Contexto → SVN Mover para aqui item(s) com controlo de versões.



B.2. Forçar utilizadores a introduzir uma mensagem de registo

Existem duas maneiras de impedir os utilizadores de submeter com uma mensagem de registo vazia. Uma é específica ao TortoiseSVN, a outra funciona para todos os clientes do Subversion mas requerendo acesso directo ao servidor.

B.2.1. Script-gancho no servidor

Se tens acesso directo ao servidor do repositório, poderás instalar um script de gancho pre-commit, que rejeita todas as submissões com uma mensagem de registo vazia ou muito curta.

Na pasta do repositório no servidor, existe uma subpasta `hooks` que contém alguns exemplos de scripts de gancho que podes usar. O ficheiro `pre-commit.tmpl` contém uma amostra de script que irá rejeitar submissões se não for fornecida mensagem de registo, ou a mesma for muito curta. O ficheiro também contém comentários em como instalar/usar este script. Segue apenas as instruções nesse ficheiro.

Este método é o método recomendado se os teus utilizadores, para além do TortoiseSVN, também usam outros clientes de Subversion. A desvantagem é que a submissão é rejeitada pelo servidor, e por isso os utilizadores irão obter uma mensagem de erro. O cliente não pode saber, antes da submissão, que ela será rejeitada. Se queres que o TortoiseSVN tenha o botão OK desactivado, até que a mensagem de registo seja suficientemente longa, então usa por favor o método descrito abaixo.

B.2.2. Propriedades do projecto

O TortoiseSVN usa propriedades para controlar algumas das suas funcionalidades. Uma dessas propriedades é a propriedade `tsvn:logminsize`.

Se tu configuras essa propriedade numa pasta, então o TortoiseSVN irá desactivar o botão OK em todas as caixas de diálogo submeter, até o utilizador introduza uma mensagem de registo com pelo menos um comprimento igual ao especificado na propriedade.

Para informação detalhada sobre essas propriedades de projecto, por favor consulte [Secção 4.17, “Configurações de Projecto”](#)

B.3. Actualizar ficheiros seleccionados a partir do repositório

Normalmente tu actualizas a tua cópia de trabalho usando TortoiseSVN → Actualizar. Mas se tu só queres seleccionar alguns ficheiros novos que um colega adicionou, sem simultaneamente integrar quaisquer alterações em outros ficheiros, necessitas então de uma abordagem diferente.

Usa TortoiseSVN → Verificar alterações, e clica em Verificar repositório para veres o que foi alterado no repositório. Selecciona os ficheiros que queres actualizar localmente e usa o menu de contexto para actualizar apenas esses ficheiros.

B.4. Reverter (Anular) revisões no repositório

B.4.1. Usar a caixa de diálogo registo de revisão

A maneira mais fácil para reverter as alterações de uma única revisão, ou de um intervalo de revisões, é usar a caixa de diálogo registo de revisões. Este é também o método a usar, se tu queres descartar alterações recentes e criar uma revisão anterior à nova HEAD.

1. Selecciona o ficheiro ou pasta na qual tu necessitas de reverter as alterações. Se queres reverter todas as alterações, esta deverá ser a pasta de nível de topo.
2. Selecciona TortoiseSVN → Mostrar registo para mostrar uma lista de revisões. Poderás necessitar de usar os botões Mostrar Todos ou Próximos 100 para mostrar as revisões em que estás interessado.
3. Selecciona a revisão que queres reverter. Se queres anular um intervalo de revisões, selecciona o primeiro e mantém premida a tecla **Shift** enquanto seleccionas a última. Ter em conta que, para revisões múltiplas, o intervalo deverá ser contínuo e sem falhas. Clicar com o botão direito nas revisões seleccionadas e então selecciona Menu de Contexto → Reverter alterações a partir desta revisão.
4. Ou se tu queres tornar uma revisão anterior a nova revisão HEAD, clica com o botão direito na revisão seleccionada e selecciona o Menu de Contexto → Reverter para esta revisão. Isto irá descartar *todas* as alterações após a revisão seleccionada.

Tu reverteste as alterações na tua cópia de trabalho. Verifica os resultados, e então submete as alterações.

B.4.2. Usa a caixa de diálogo integrar

Para anular um grande intervalo de revisões poderás usar a caixa de diálogo Integrar. O método anterior usa a integração por detrás do pano; este método usa-a explicitamente.

1. Na tua cópia de trabalho selecciona TortoiseSVN → Integrar.
2. No campo De: introduz o URL completo da pasta do ramo ou etiqueta, contendo as alterações que tu queres reverter na tua cópia de trabalho. Este deverá surgir como o URL por defeito.

3. No campo **Da Revisão** introduz o número de revisão em que correntemente estás. e tens a certeza que mais ninguém está efectuar alterações, podes usar a revisão HEAD.
4. Tem a certeza que está verificada a caixa Use "From:" URL
5. No campo **Para Revisão** introduz o número de revisão para o qual queres reverter, ou seja a *anterior* à primeira revisão a ser revertida
6. Clica no OK para completar a integração

Tu reverteste as alterações na tua cópia de trabalho. Verifica os resultados, e então submetes as alterações.

B.4.3. Usa o `svndumpfilter`

Visto que o TortoiseSVN nunca perde dados, as tuas revisões “revertidas” ainda existem como revisões intermédias no repositório. Só a revisão HEAD foi alterada para um estado anterior. Se tu queres fazer desaparecer completamente revisões do teu repositório, apagando todos os vestígios de qualquer existência, terás de usar medidas mais extremas. A não ser que exista uma razão muito boa para o fazer, *não é recomendado*. Uma possível razão seria a que alguém submetesse um documento confidencial num repositório público.

A única maneira de remover dados do repositório é usar a ferramenta de linha de comando do Subversion `svnadmin`. Tu poderás encontrar a descrição de como isto se processa em [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. Comparar duas revisões de um ficheiro ou pasta

Se queres comparar duas revisões na história de um item, por exemplo as revisões 100 e 200 do mesmo ficheiro, usa apenas TortoiseSVN → **Mostrar registo** para listar a história de revisões desse ficheiro. Selecciona duas revisões que queres comparar e então usa **Menu de Contexto** → **Comparar Revisões**.

Se queres comparar o mesmo item em duas árvores diferentes, por exemplo o trunk e um ramo, podes usar o navegador de repositório para abrir ambas as árvores e seleccionar o ficheiro em ambos os locais, e então usar **Menu de Contexto** → **Comparar Revisões**.

Se tu queres comparar duas árvores para ver o que foi alterado, por exemplo o trunk e uma entrega etiquetada, podes usar TortoiseSVN → **Navegador de Repositório** Selecciona os dois nós a comparar e usa **Menu de Contexto** → **Comparar revisões HEAD**. Isto irá mostrar uma lista de ficheiros alterados, e tu podes então seleccionar ficheiros individuais para ver as alterações em detalhe. Tu podes também exportar a estrutura de árvore contendo todos os ficheiros alterados, ou simplesmente uma lista de ficheiros alterados. Consulta [Secção 4.10.3, “Comparando Pastas”](#) para mais informações. Em alternativa usa **Menu de Contexto** → **Comparação Unificada de Revisões HEAD** para ver um sumário de todas as diferenças com um contexto minimal.

B.6. Incluir um subprojecto comum

Por vezes tu quererás incluir outro projecto na tua cópia de trabalho, talvez uma biblioteca de código. Tu não quererás criar um duplicado deste código no teu repositório, porque então, tu irás perder ligação com o código original (e mantido). Ou talvez tenhas vários projectos que partilham código aqui. Existem pelo menos 3 maneiras de lidar com isto.

B.6.1. Usa o `svn:externals`

Configurar a propriedade `svn:externals` para uma pasta no teu projecto. Esta propriedade consiste em uma ou mais linhas; cada linha tem o nome de uma subpasta que quererás usar como pasta de checkout

para código comum, e o URL do repositório do qual queres efectuar aqui checkout. Para mais detalhes consultar [Secção 4.18, "Itens Externos"](#).

Submeter a nova pasta. Agora quando actualizares, o Subversion trará uma cópia desse projecto do seu repositório para a tua cópia de trabalho. As subpastas, se necessário, serão automaticamente criadas. De cada vez que actualizares a tua principal cópia de trabalho, irás também receber a última versão de todos os projectos externos.

Se o projecto externo está no mesmo repositório, quaisquer alterações que faças aí, serão incluídas na lista de submissões, quando submeteres o teu projecto principal.

Se o projecto externo está num repositório diferente, qualquer alteração que fizeres no mesmo, será notificada quando submeteres o teu projecto principal, no entanto, terás de submeter essas alterações externas separadamente.

Dos três métodos descritos, este é o único que não requer instalação no lado do cliente. Assim que os externos sejam especificados nas propriedades da pasta, todos os clientes irão adquirir pastas povoadas, assim que actualizarem.

B.6.2. Usar uma cópia de trabalho aninhada.

Cria uma nova pasta no teu projecto para conter o código comum, mas não o adiciones ao Subversion.

Selecciona TortoiseSVN → Checkout para a nova pasta e efectua o checkout da cópia do código comum lá para dentro. Agora tu tens uma cópia de trabalho separada, aninhada na tua cópia de trabalho principal.

As duas cópias de trabalho são independentes. Quando tu submetes alterações na cópia pai as alterações na CT aninhada são ignoradas. por sua vez quando tu actualizas o pai a CT aninhada não será actualizada.

B.6.3. Usa uma localização relativa

Se usas o mesmo núcleo de código comum em vários projectos, e não queres manter múltiplas cópias de trabalho em cada projecto que o usa, poderás apenas efectuar um checkout numa localização separada que estará relacionada com todos os outros projectos que a usam. Por exemplo:

```
C:\Projects\Proj1
C:\Projects\Proj2
C:\Projects\Proj3
C:\Projects\Common
```

e referência o código comum recorrendo a um caminho relativo, e.g. `..\..\Common\DSPcore`.

Se os teus projectos estão espalhados por localizações não relacionadas, poderás usar uma variante que consiste em, colocar o código comum numa localização e usar uma letra de unidade de rede para mapear essa localização para algo que possas fixar nos teus projectos, e.g. Efectuar checkout do código comum para `D:\Documents\Framework` ou `C:\Documents and Settings\{login}\My Documents\framework` e então usar

```
SUBST X: "D:\Documents\framework"
```

para criar o mapeamento de unidade usado no teu código. Então o teu código poderá usar localizações absolutas.

```
#include "X:\superio\superio.h"
```

Este método só irá funcionar num ambiente apenas-PCs, e precisarás de documentar os mapeamentos de unidade de rede, para que a tua equipa saiba onde estão esses ficheiros misteriosos. Este método é para uso estrito em ambientes de desenvolvimento fechado e não é recomendado para uso geral.

B.7. Criar um atalho para um repositório

Se precisas com frequência de abrir o navegador de repositório numa localização específica, podes criar um atalho de ambiente de trabalho, usando o interface de automação para o TortoiseProc. Cria apenas um novo atalho e aponta-o para:

```
TortoiseProc.exe /command:repobrowser /path:"url/to/repository"
```

É claro que necessitas de inserir o URL do repositório real.

B.8. Ignorar ficheiros que já estão versionados

Se tu acidentalmente adicionaste alguns ficheiros que deveriam ter sido ignorados, como é que os tiras do controlo de versões sem os perderes? Talvez tenhas o teu próprio ficheiro de configuração do IDE, que não faz parte do projecto, mas que levou bastante tempo a configurar ao teu modo.

Se ainda não submeteste a adição, então tudo o que tens de fazer é usa o TortoiseSVN → Reverter... para anular a adição. Deverás então adicionar o(s) ficheiro(s) à lista de ignorados, para que eles não sejam adicionados outra vez por engano.

Se os ficheiros já estão no repositório, terás de ter um pouco mais de trabalho

1. Sustem a tecla **Shift** de modo a acederes ao menu de contexto estendido e usa TortoiseSVN → Remove (manter local) para marcar o ficheiro/pasta para remoção do repositório, sem perderes a cópia local.
2. TortoiseSVN → Submeter a pasta pai.
3. Adiciona o ficheiro/pasta à lista de ignorados, para que não tenhas o mesmo problema novamente.

B.9. Remover uma cópia de trabalho do controlo de versões

Se tu tens uma cópia de trabalho que queres converter de volta a uma pasta simples, sem as pastas `.svn`, podes simplesmente exportá-la para ela própria. Consulta [Secção 4.26.1, “Remover uma cópia de trabalho do controlo de versões”](#) para descobrir como o fazer.

B.10. Remover uma cópia de trabalho

Se tu tens uma cópia de trabalho que já não necessitas, como te livrares dela de modo correcto? Fácil - apenas apaga-a no Explorador do Windows! Cópias de trabalho são entidades locais privadas auto contidas.

Apêndice C. Dicas Úteis para Administradores

Este anexo contém soluções para problemas/questions que tu possas ter quando és responsável por instalar o TortoiseSVN em múltiplos computadores clientes.

C.1. Instalar o TortoiseSVN via políticas de grupo

O instalador do TortoiseSVN é um ficheiro MSI, o que significa que não deverás ter problemas a adicionar esse ficheiro MSI às políticas de grupo do teu controlador de domínio.

Uma boa abordagem de como fazer isso poderá ser encontrada no artigo 314934 da base de conhecimento da Microsoft: <http://support.microsoft.com/?kbid=314934>.

As versões 1.3.0 e superior do TortoiseSVN têm de ser instaladas sob *Configuration de Computador* e não sob *Configuração de Utilizador*. Isto é porque essas versões necessitam das novas DLLs CRT e MFC, que só podem ser instaladas *por computador* e não *por utilizador*. Se precisas mesmo de instalar o TortoiseSVN numa base por utilizador, então deves primeiro instalar o pacote MFC e CRT versão 8 da Microsoft em cada computador, se queres instalar o TortoiseSVN como por utilizador.

C.2. Redireccionar a verificação de actualização

O TortoiseSVN, com a periodicidade de alguns dias, verifica se existe uma nova versão disponível. Se existir disponível uma nova versão, uma caixa de diálogo aparece a informar o utilizador sobre esse facto.

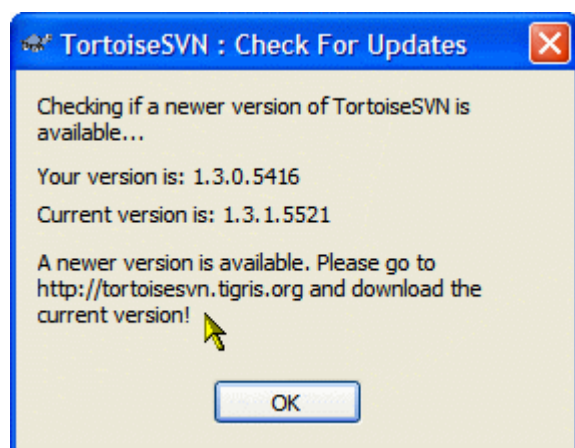


Figura C.1. A Caixa de Diálogo Actualização

Se és responsável por muitos utilizadores num domínio poderás querer que os teus utilizadores só usem versões que tu aprovaste, e não tenham instalada a última versão. Provavelmente não queres que apareça a caixa de diálogo de actualização, para que os teus utilizadores não vão imediatamente actualizar.

As versões 1.4.0 e superior do TortoiseSVN permitem-te redireccionar a verificação de actualização para o teu servidor de intranet. Tu podes configurar a chave do registo HKCU\Software\TortoiseSVN\UpdateCheckURL (valor string) para um URL a apontar para um ficheiro de texto na tua intranet. Esse ficheiro de texto tem de ter o seguinte formato:

1.4.1.6000

Está disponível uma nova versão do TortoiseSVN para descarregares!!
<http://192.168.2.1/downloads/TortoiseSVN-1.4.1.6000-svn-1.4.0.msi>

A primeira linha desse ficheiro é a string da versão. Deverás ter a certeza que esta condiz com a string da versão exacta do pacote de instalação do TortoiseSVN. A segunda linha é um texto personalizado mostrado na caixa de diálogo de actualização. Podes escrever aqui o que entenderes. Tem apenas em atenção que o espaço é limitado, mensagens muito longas serão truncadas! A terceira linha é o URL para o pacote da nova instalação. Este URL é aberto quando o utilizador clica na mensagem personalizada na caixa de diálogo de actualização. Tu também podes apenas apontar o utilizador para uma página web em vez de directamente para um ficheiro MSI. O URL é aberto com o browser por defeito, pelo que se especificares uma página web, essa página é aberta e mostrada ao utilizador. Se especificares o pacote MSI o browser irá interrogar o utilizador a guardar o ficheiro MSI localmente.

C.3. Configurar a variável de ambiente SVN_ASP_DOT_NET_HACK

As of version 1.4.0 and later, the TortoiseSVN installer doesn't provide the user with the option to set the SVN_ASP_DOT_NET_HACK environment variable anymore, since that caused many problems and confusions with users which always install *everything* no matter if they know what it is for.

Mas essa opção está só escondida para o utilizador. Podes ainda forçar o instalador do TortoiseSVN a configurar essa variável de ambiente fixando a propriedade ASPDOTNETHACK a TRUE. Por exemplo, tu podes arrancar o instalador da seguinte forma:

```
msiexec /i TortoiseSVN-1.4.0.msi ASPDOTNETHACK=TRUE
```

C.4. Desactivar entradas do menu de contexto

Como na versão 1.5.0 e superior, o TortoiseSVN permite-te desactivar (na realidade, esconder) entradas do menu de contexto. Visto que esta funcionalidade, que não deverá ser usada de ânimo leve mas só se existir uma razão forte, não tem GUI para isto e terá de ser feita directamente no registo. Isto pode ser usado para desactivar certos comandos para que os utilizadores não os usem. Mas por favor, tem em conta que só as entradas do menu de contexto no *explorador* são escondidos, e os comandos continuam disponíveis através de outros meios, e.g. a linha de comandos ou mesmo noutras caixas de diálogo no próprio TortoiseSVN!

As chaves de registo que guardam a informação de que entradas de menu de contexto são mostradas, são HKEY_CURRENT_USER\Software\TortoiseSVN\ContextMenuEntriesMaskLow e HKEY_CURRENT_USER\Software\TortoiseSVN\ContextMenuEntriesMaskHigh.

Cada uma dessas entradas de registo é um valor DWORD, com cada bit a corresponder a uma entrada específica do menu. Um bit activo, significa que a entrada de menu correspondente está desactivada.

Valor	Entrada de menu
0x0000000000000001	Checkout
0x0000000000000002	Actualizar
0x0000000000000004	Submeter
0x0000000000000008	Adicionar
0x0000000000000010	Reverter
0x0000000000000020	Limpar
0x0000000000000040	Resolver
0x0000000000000080	Trocar
0x0000000000000100	Importar

Valor	Entrada de menu
0x0000000000000200	Exportar
0x0000000000000400	Criar repositório aqui
0x0000000000000800	Ramo/Etiqueta
0x0000000000001000	Integrar
0x0000000000002000	Remover
0x0000000000004000	Alterar nome
0x0000000000008000	Actualizar para revisão
0x0000000000010000	Comparar
0x0000000000020000	Mostrar Registo
0x0000000000040000	Editar Conflitos
0x0000000000080000	Reposicionar
0x0000000000100000	Verificar alterações
0x0000000000200000	Ignorar
0x0000000000400000	Navegador de Repositório
0x0000000000800000	Responsabilizar
0x0000000001000000	Criar Correção
0x0000000002000000	Aplicar Correção
0x0000000004000000	Gráfico de revisões
0x0000000008000000	Bloquear
0x0000000010000000	Libertar Bloqueio
0x0000000020000000	Propriedades
0x0000000040000000	Comparar com URL
0x0000000080000000	Remover itens não versionados
0x2000000000000000	Preferências
0x4000000000000000	Ajuda
0x8000000000000000	Sobre

Tabela C.1. Entradas de menu e seus valores

Exemplo: para desactivar as entradas de menu “Reposicionar”, o “Remover itens não versionados” e o “Preferências”, adiciona os valores atribuídos a essas entradas do seguinte modo:

```

0x0000000000008000
+ 0x0000000008000000
+ 0x2000000000000000
= 0x2000000008008000

```

o valor DWORDo mais baixo (0x80080000) deverá ser então armazenado na HKEY_CURRENT_USER\Software\TortoiseSVN\ContextMenuEntriesMaskLow, o valor DWORD mais alto (0x20000000) em HKEY_CURRENT_USER\Software\TortoiseSVN\ContextMenuEntriesMaskHigh.

Para re-activar de novo as entradas do menu, simplesmente apaga as duas chaves de registo.

Apêndice D. Automatizar o TortoiseSVN

Visto que todos os comandos do TortoiseSVN são controlados através de parâmetros de linha de comandos, tu podes automatizá-los com scripts de batch ou arrancar comandos específicos e caixas de diálogo a partir de outros comandos (e.g. o teu editor de texto preferido).



Importante

Lembra-te que o TortoiseSVN é um cliente de GUI e este guia de automação mostra-te como fazer as caixas de diálogo do TortoiseSVN aparecer, para colectar a entrada do utilizador. Se tu queres escrever um script que não requer entrada de dados, deverás usar em alternativa, o cliente de linha de comandos oficial do Subversion.

D.1. Comandos TortoiseSVN

O programa de GUI do TortoiseSVN chama-se `TortoiseProc.exe`. Todos os comandos são especificados com o parâmetro `/command:abcd` onde `abcd` é o nome do comando requerido. A maioria desses comandos necessitam pelo menos de um argumento de caminho, que é dado com `/path:"some\path"`. Na tabela seguinte o comando refere-se ao parâmetro `/command:abcd` e o caminho refere-se ao parâmetro `/path:"some\path"`.

Visto que alguns dos comandos podem necessitar de uma lista de caminhos de destinos (e.g. submeter vários ficheiros específicos), o parâmetro `/path` pode requerer vários caminhos, separados pelo caracter `*`.

O TortoiseSVN usa ficheiros temporários para passar múltiplos argumentos entre a extensão da shell e o programa principal. A partir do TortoiseSVN 1.5.0 o parâmetro `/notempfile` é obsoleto e não é mais necessário adicioná-lo.

A caixa de diálogo que é usada para submissões, actualizações e muitos mais comandos fica usualmente aberta depois do comando ter acabado até o utilizador premir o botão **OK**. Isto pode ser alterado ao verificar a opção correspondente na caixa de diálogo preferências. Mas ao usar essa preferência, ir-se-á fechar a caixa de diálogo de progresso independentemente de iniciares o comando a partir de um ficheiro de batch ou a partir do menu de contexto do TortoiseSVN.

Para especificar uma localização diferente para o ficheiro de configuração, usa o parâmetro `/configdir:"path\to\config\directory"`. Isto irá substituir o caminho por defeito, incluindo qualquer configuração do registo.

Para fechar automaticamente a caixa de diálogo de progresso no fim de um comando, sem usar uma configuração permanente, podes passar o parâmetro `/closeonend`.

- `/closeonend:0` não fechar automaticamente a caixa de diálogo
- `/closeonend:1` fechar automaticamente se não existem erros
- `/closeonend:2` fechar automaticamente se não existem erros e conflitos
- `/closeonend:3` fechar automaticamente se não existem erros, conflitos e integrações
- `/closeonend:4` fecha automaticamente se não existirem erros, conflitos e integrações para operações locais

A tabela abaixo lista todos os comandos que podem ser acedidos usando a linha de comandos TortoiseProc.exe. Como descrito acima, esses deverão ser usados na forma `/command:abcd`. Na tabela é omitido o prefixo `/command` por questões de espaço.

Comando	Descrição
:about	Mostra a caixa de diálogo Sobre. Esta é também mostrada se não é dado nenhum comando.
:log	Opens the log dialog. The <code>/path</code> specifies the file or folder for which the log should be shown. Three additional options can be set: <code>/startrev:xxx</code> , <code>/endrev:xxx</code> and <code>/strict</code>
:checkout	Abre a caixa de diálogo checkout. O <code>/path</code> especifica a directoria destino e o <code>/url</code> especifica o URL de onde efectuar o checkout.
:import	Abre a caixa de diálogo importar. O <code>/path</code> especifica a pasta com os dados a importar.
:update	Updates the working copy in <code>/path</code> to HEAD. If the option <code>/rev</code> is given then a dialog is shown to ask the user to which revision the update should go. To avoid the dialog specify a revision number <code>/rev:1234</code> . Other options are <code>/nonrecursive</code> and <code>/ignoreexternals</code> .
:commit	Abre a caixa de diálogo submeter. O <code>/path</code> especifica a pasta de destino ou a lista de ficheiros a submeter. Tu podes também especificar a opção <code>/logmsg</code> para passar uma mensagem de registo pré-definida para a caixa de diálogo submeter. Ou se não queres passar a mensagem de registo na linha de comandos, usa <code>/logmsgfile:path</code> , onde o caminho aponta para um ficheiro que contém a mensagem de registo. Para pré-preencher a caixa do ID-Problema (no caso de teres configurado correctamente a integração com os seguidores de bugs), podes usar para isso o <code>/bugid:"o id do bug aqui"</code> .
:add	Adiciona ao controlo de versões os ficheiros em <code>/path</code>
:revert	Reverte as modificações locais de uma cópia de trabalho. O <code>/path</code> diz quais os itens a reverter.
:cleanup	Cleans up interrupted or aborted operations and unlocks the working copy in <code>/path</code> .
:resolve	Marca um ficheiro em conflito em <code>/path</code> como resolvido. Se for dado o <code>/noquestion</code> então a resolução é feita sem perguntar primeiro ao utilizador se ela deverá ser mesmo feita.
:repopulate	Cria um repositório em <code>/path</code>
:switch	Abre a caixa de diálogo trocar. O <code>/path</code> especifica a pasta destino.
:export	Exporta a cópia de trabalho em <code>/path</code> para outra pasta. Se o <code>/path</code> apontar para uma pasta não versionada, a caixa de diálogo perguntará por um URL para exportar para a pasta em <code>/path</code> .
:merge	Abre a caixa de diálogo integrar. O <code>/path</code> especifica a pasta de destino. Para integrar um intervalo de revisões estão disponíveis as seguintes opções: <code>/fromurl:URL</code> , <code>/revrange:string</code> . Para integrar duas árvores de repositório, estão disponíveis as seguintes opções: <code>/fromurl:URL</code> , <code>/tourl:URL</code> , <code>/fromrev:xxx</code> e <code>/torev:xxx</code> . Estas pré-preencher os campos relevantes na caixa de diálogo integrar.
:mergeall	Abre a caixa de diálogo integrar todos. O <code>/path</code> especifica a pasta de destino.
:copy	Mostra a caixa de diálogo ramo/etiqueta. O <code>/path</code> é a cópia de trabalho de onde efectuar o ramo/etiqueta. E o <code>/url</code> é o URL destino. Tu também podes especificar a opção <code>/logmsg</code> para passar uma mensagem de registo pré-

Comando	Descrição
	definida para a caixa de diálogo ramo/etiqueta. Ou se não queres passar a mensagem de registo na linha de comandos, usa o <code>/logmsgfile:path</code> , onde o caminho aponta para um ficheiro que contém a mensagem de registo.
<code>:settings</code>	Abre a caixa de diálogo preferências.
<code>:remove</code>	Remove do controlo de versões o(s) ficheiro(s) em <code>/path</code> .
<code>:rename</code>	Altera o nome do ficheiro em <code>/path</code> . O novo nome do ficheiro será pedido numa caixa de diálogo. Para evitar a pergunta sobre alterar o nome a ficheiros similares num só passo, passa <code>/noquestion</code> .
<code>:diff</code>	Starts the external diff program specified in the TortoiseSVN settings. The <code>/path</code> specifies the first file. If the option <code>/path2</code> is set, then the diff program is started with those two files. If <code>/path2</code> is omitted, then the diff is done between the file in <code>/path</code> and its BASE. To explicitly set the revision numbers use <code>/startrev:xxx</code> and <code>/endrev:xxx</code> . If <code>/blame</code> is set and <code>/path2</code> is not set, then the diff is done by first blaming the files with the given revisions.
<code>:showcompare</code>	Dependendo dos URLs e revisões a comparar, isto ou mostra uma comparação unificada (se a opção <code>unified</code> for usada), uma caixa de diálogo com a lista de ficheiros que foram alterados, ou se os URLs apontam para ficheiros arranca o visualizador de comparação para esses dois ficheiros. As opções <code>url1</code> , <code>url2</code> , <code>revision1</code> e <code>revision2</code> devem ser especificadas. As opções <code>pegrevision</code> , <code>ignoreancestry</code> , <code>blame</code> and <code>unified</code> são opcionais.
<code>:conflicteditor</code>	Arranca o editor de conflitos especificado nas preferências do TortoiseSVN com os ficheiros correctos para o ficheiro em conflito no <code>/path</code> .
<code>:relocate</code>	Abre a caixa de diálogo reposicionar. O <code>/path</code> especifica o caminho da cópia de trabalho a reposicionar.
<code>:help</code>	Abre o ficheiro de ajuda.
<code>:repostatus</code>	Abre a caixa de diálogo verificar-por-modificações. O <code>/path</code> especifica a pasta da cópia de trabalho.
<code>:repobrowser</code>	Arranca o navegador de repositório apontando para o URL da cópia de trabalho dada pelo <code>/path</code> , ou o <code>/path</code> aponta directamente para um URL. Uma opção adicional <code>/rev:xxx</code> , pode ser usada para especificar qual a revisão que o navegador de repositório deve mostrar. Se for omitido o <code>/rev:xxx</code> , será por defeito HEAD. Se o <code>/path</code> apontar para um URL, o <code>/projectpropertiespath:path/to/wc</code> especifica o caminho de onde deve ler e usar as propriedades do projecto.
<code>:ignore</code>	Adiciona todos os objectos em <code>/path</code> à lista de ignorados, i.e. adiciona a propriedade <code>svn:ignore</code> a esses ficheiros.
<code>:blame</code>	Abre a caixa de diálogo responsabilizar paa o ficheiro especificado em <code>/path</code> . Se as opções <code>/startrev</code> e <code>/endrev</code> são usadas, então a caixa de diálogo que pergunta pelo intervalo de responsabilidade não será mostrada, e em alternativa será usado os valores dessas opções. Se a opção <code>/line:nnn</code> for usada, o TortoiseBlame irá abrir mostrando a linha cujo número foi especificado. As opções <code>/ignoreeol</code> , <code>/ignorespaces</code> e <code>/ignoreallspaces</code> são também suportadas.

Comando	Descrição
:cat	Guarda um ficheiro de um URL ou cópia de trabalho dada em /path para a localização especificada em /savepath:path. A revisão é dada em /revision:xxx. Isto pode ser usado para obter um ficheiro com uma revisão específica.
:createpatch	Cria um ficheiro de correção para o caminho dado em /path.
:revisiongraph	Mostra o gráfico de revisões para o caminho dado em /path.
:lock	Bloqueia um ficheiro ou todos os ficheiros numa pasta dada por /path. É mostrada a caixa de diálogo 'Bloquear' para que o utilizador possa introduzir um comentário para o bloqueio.
:unlock	Desbloqueia um ficheiro ou todos os ficheiros numa pasta dada por /path.
:rebuildiconcache	Reconstrói a cache de ícones do windows. Usa só isto no caso de os ícones do Windows ficarem corrompidos. Um efeito secundário disto (que não pode ser evitado) é que os ícones no ambiente de trabalho são reordenados. Para suprimir a caixa de mensagem, passa o /noquestion.
:properties	Mostra a caixa de diálogo propriedades para o caminho dado por /path.

Tabela D.1. Lista de comandos e opções disponíveis

Exemplos (que deveram ser introduzidos numa só linha):

```
TortoiseProc.exe /command:commit
                 /path:"c:\svn_wc\file1.txt*c:\svn_wc\file2.txt"
                 /logmsg:"test log message" /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. Comandos TortoiseIDiff

A ferramenta de comparação de imagem tem algumas opções de linha de comandos que tu podes usar para controlar o modo como a ferramenta arranca. O programa chama-se `TortoiseIDiff.exe`.

A tabela abaixo lista todas as opções que podem ser passadas, a partir da linha de comandos, para a ferramenta de comparação de imagens.

Opção	Descrição
:left	Caminho para o ficheiro mostrado à esquerda.
:lefttitle	A string de título. Esta string é usada no título da vista da imagem em vez do caminho completo para o ficheiro da mesma.
:right	Caminho para o ficheiro mostrado à direita.
:righttitle	A string de título. Esta string é usada no título da vista da imagem em vez do caminho completo para o ficheiro da mesma.
:overlay	Se especificada, a ferramenta de comparação de imagens comuta para o modo sobreposição (alpha blend).
:fit	Se especificada, a ferramenta de comparação de imagens encaixa as duas imagens juntas.

Opção	Descrição
:showinfo	Mostra a caixa de informações da imagem.

Tabela D.2. Lista de opções disponíveis

Exemplo (que deverá ser introduzido numa linha só):

```
TortoiseIDiff.exe /left:"c:\images\img1.jpg" /lefttitle:"image 1"  
                /right:"c:\images\img2.jpg" /righttitle:"image 2"  
                /fit /overlay
```

Apêndice E. Referência Cruzada da Interface de Linha de Comandos

Por vezes este manual refere-se à principal documentação do Subversion, que descreve o Subversion em termos da Interface de Linha de Comandos (ILC). Para te ajudar a compreender o que o tortoise faz nos bastidores, nós compilámos uma lista mostrando os comandos ILC equivalentes, para cada comando do GUI do TortoiseSVN.

Nota

Mesmo que hajam ILC equivalentes para o que o TortoiseSVN faz, lembra-te que o TortoiseSVN *não* chama o ILC mas usa directamente a biblioteca do Subversion.

Se pensas que encontres um bug no TortoiseSVN, podemos pedir-te para o reproduzir, usando o ILC, para que possamos distinguir os problemas do TortoiseSVN dos do Subversion. Esta referência diz qual o comando a tentar.

E.1. Convenções e Regras Básicas

Nas descrições que se seguem, o URL de uma localização do repositório é mostrado simplesmente como URL, e um exemplo poderá ser `http://tortoisesvn.googlecode.com/svn/trunk`. O caminho da cópia de trabalho é simplesmente mostrado como PATH, e um exemplo poderá ser `C:\TortoiseSVN runk`.



Importante

Porque o TortoiseSVN é uma extensão da Shell do Windows, não é capaz de usar a noção de pasta corrente de trabalho. Todos os caminhos da cópia de trabalho deverão ser fornecidos usando o caminho absoluto e não o caminho relativo.

Certos itens são opcionais, e esses no TortoiseSVN são frequentemente controlados por caixas de verificação ou botões de rádio. Essas opções são mostradas em [parênteses rectos] nas definições da linha de comandos.

E.2. Comandos TortoiseSVN

E.2.1. Checkout

```
svn checkout [-N] [--ignore-externals] [-r rev] URL PATH
```

Se estiver assinalada a caixa de verificação Checkout apenas à pasta de topo, usar a opção `-N`.

Se estiver verificada a Omitir externos, usa o parâmetro `--ignore-externals`

Se estiveres efectuar checkout a uma revisão específica, especifica o URL usando o parâmetro `-r`.

E.2.2. Actualizar

```
svn info URL_of_WC
svn update [-r rev] PATH
```

O actualizar de itens múltiplos não é correntemente uma actividade atómica no Subversion. Então o TortoiseSVN procura primeiro a revisão HEAD do repositório e só então actualiza todos os itens para o número de revisão em particular, para evitar criar de uma cópia de trabalho com revisões misturadas.

Se só um item está seleccionado para actualizar ou os itens seleccionados não são todos do mesmo repositório, o TortoiseSVN apenas actualiza para a HEAD.

Não são usadas aqui opções de linha de comandos. O Actualizar para revisão também implementa o comando actualizar, mas oferece mais opções.

E.2.3. Actualizar para Revisão

```
svn info URL_of_WC
svn update [-r rev] [-N] [--ignore-externals] PATH
```

Se estiver verificada a Actualizar apenas a pasta de topo, usa apenas o parâmetro `-N`.

Se estiver verificada a Omitir externos, usa o parâmetro `--ignore-externals`

E.2.4. Submeter

No TortoiseSVN a caixa de diálogo submeter usa vários comandos do Subversion. A primeira fase é uma verificação de estado, que determina quais os itens, na tua cópia de trabalho, que podem ser potencialmente submetidos. Tu podes rever a lista, compara os ficheiros com a BASE, e seleccionar quais os itens que queres incluir na submissão.

```
svn status -v PATH
```

Se a **Mostrar ficheiros não versionados** está verificada, o TortoiseSVN irá também mostrar todos os ficheiros e pastas não versionadas na hierarquia da cópia de trabalho, tendo em conta as regras de ignorar. Esta funcionalidade em particular, não tem equivalente no Subversion já que o comando `svn status` não desce às pastas não versionadas.

Se tu seleccionares qualquer pasta ou ficheiro não versionado, esses itens serão primeiro adicionados à tua cópia de trabalho

```
svn add PATH...
```

Quando clicas no OK é efectuada a submissão Subversion. Se deixaste todas as caixas de selecção de ficheiros no seu estado por defeito, o TortoiseSVN usa uma única submissão recursiva da cópia de trabalho. Se desseleccionaste alguns ficheiros, então será usada uma submissão não recursiva (`-N`), e cada caminho terá de ser especificado individualmente na linha de comandos da submissão.

```
svn commit -m "MensagemRegisto" [-N] [--no-unlock] PATH...
```

A `LogMessage` representa aqui o conteúdo da caixa de edição da mensagem de registo. Esta pode ser vazia.

Se a **Manter bloqueios** estiver verificada, usa o parâmetro `--no-unlock`.

E.2.5. Comparar

```
svn diff PATH
```

Se usa o Comparar a partir do menu de contexto principal, estarás a comparar o ficheiro modificado com a sua revisão BASE. A saída do comando ILC acima, também faz isto e produz a saída no formato

comparação-unificada. No entanto este não é o que o TortoiseSVN está a usar. O TortoiseSVN usa o TortoiseMerge (ou um programa de comparação à tua escolha) para mostrar visualmente as diferenças entre os ficheiros de texto integral, pelo que não existe nenhum ILC equivalente.

Tu também podes comparar quaisquer 2 ficheiros usando o TortoiseSVN, estejam eles ou não sob controlo de versões. O TortoiseSVN apenas carrega os dois ficheiros, para o programa de comparação escolhido, e deixa-o descobrir onde estão as diferenças.

E.2.6. Mostrar Registo

```
svn log -v -r 0:N --limit 100 [--stop-on-copy] PATH
  or
svn log -v -r M:N [--stop-on-copy] PATH
```

Por defeito, o TortoiseSVN tenta obter 100 mensagens de registo usando o método `--limit`. Se as preferências lhe dizem para usar as APIs antigas, então uma segunda forma é usada para obter as mensagens de registo de 100 revisões do repositório.

Se Parar ao copiar/mudar nome estiver verificado, usa a opção `--stop-on-copy`.

E.2.7. Verificar Modificações

```
svn status -v PATH
  or
svn status -u -v PATH
```

A verificação do estado inicial olha só para a tua cópia de trabalho. Se clicares em **Verificar repositório** então o repositório é também verificado para ver que ficheiros puderam ter sido alterados por uma actualização, o que requer a opção `-u`.

Se a **Mostrar ficheiros não versionados** está verificada, o TortoiseSVN irá também mostrar todos os ficheiros e pastas não versionadas na hierarquia da cópia de trabalho, tendo em conta as regras de ignorar. Esta funcionalidade em particular, não tem equivalente no Subversion já que o comando `svn status` não desce às pastas não versionadas.

E.2.8. Gráfico de Revisões

O gráfico de revisões é uma funcionalidade só do TortoiseSVN. Não existe um equivalente no cliente de linha de comandos.

O que faz o TortoiseSVN é um

```
svn info URL_of_WC
svn log -v URL
```

onde o URL é a *raiz* do repositório, e então analisa os dados retornados.

E.2.9. Navegador de Repositório

```
svn info URL_of_WC
svn list [-r rev] -v URL
```

Podes usar o `svn info` para determinar a raiz do repositório, que é o nível de topo mostrado no navegador de repositório. Tu não podes navegar **Para cima** para além deste nível. Este comando também retorna toda a informação de bloqueio, mostrada no navegador de repositório.

A chamada `svn list` irá listar os conteúdos de uma pasta, dado o URL e revisão.

E.2.10. Editar Conflitos

Este comando não tem equivalente ILC. Ele invoca o TortoiseMerge ou, uma ferramenta externa de comparação/integração de 3 modos, para olhar para os ficheiros envolvidos no conflito e determinar quais as linhas a usar.

E.2.11. Resolvido

```
svn resolved PATH
```

E.2.12. Alterar nome

```
svn rename CURR_PATH NEW_PATH
```

E.2.13. Remover

```
svn delete PATH
```

E.2.14. Reverter

```
svn status -v PATH
```

A primeira fase é uma verificação do estado, que determina quais os itens na tua cópia de trabalho, que podem potencialmente ser revertidos. Tu podes rever a lista, comparar os ficheiros com a BASE e seleccionar os itens que queres que sejam incluídos na reversão.

Quando tu clicas no OK, toma lugar a reversão Subversion. Se deixaste todas as caixas de selecção de ficheiros no seu estado por defeito, o TortoiseSVN usa uma reversão simples e recursiva (-R) da cópia de trabalho. Se desseleccionaste alguns ficheiros, então cada caminho deve ser individualmente especificado na linha de comando da reversão.

```
svn revert [-R] PATH...
```

E.2.15. Limpar

```
svn limpar PATH
```

E.2.16. Obter "Lock"

```
svn status -v PATH
```

A primeira fase é uma verificação de estado, que determina os ficheiros na tua cópia de trabalho que podem potencialmente ser bloqueados. Tu podes seleccionar os itens que queres que sejam bloqueados.

```
svn lock -m "LockMessage" [--force] PATH...
```

LockMessage representa aqui os conteúdos da caixa de edição da mensagem de bloqueio. Isto pode ser vazio.

Se estiver verificada a Roubar bloqueios usa a opção `--force`.

E.2.17. Libertar "Lock"

```
svn unlock PATH
```

E.2.18. Ramo/Etiqueta

```
svn copy -m "LogMessage" URL URL  
or  
svn copy -m "LogMessage" URL@rev URL@rev  
or  
svn copy -m "LogMessage" PATH URL
```

A caixa de diálogo Ramo/Etiqueta executa uma cópia para o repositório. Existem 3 botões de rádio com opções:

- A revisão HEAD do repositório
- Revisão específica no repositório
- Cópia de trabalho

que correspondem às 3 variantes de linha de comando acima mostradas.

A LogMessage representa aqui o conteúdo da caixa de edição da mensagem de registo. Esta pode ser vazia.

E.2.19. Trocar

```
svn info URL_of_WC  
svn switch [-r rev] URL PATH
```

E.2.20. Integrar

```
svn merge [--dry-run] --force From_URL@revN To_URL@revM PATH
```

O Integração de Teste executa a mesma integração com a opção `--dry-run`.

```
svn diff From_URL@revN To_URL@revM
```

A Comparação unificada mostra a operação de comparação que será usada para executar a integração.

E.2.21. Exportar

```
svn export [-r rev] [--ignore-externals] URL Export_PATH
```

Esta forma é usada quando acedido a partir de uma pasta não versionada, e a pasta é usada como destino.

Exportar uma cópia de trabalho para uma localização diferente, é efectuada sem o uso da biblioteca do Subversion, pelo que não existe comando de linha equivalente.

O que o TortoiseSVN faz é copiar todos os ficheiros para uma nova localização, enquanto mostra o progresso da operação. Ficheiros/pastas não versionados poderão ser também exportados, opcionalmente.

Em ambos os casos, se está verificada a Omitir externos, usa a opção `--ignore-externals`.

E.2.22. Reposicionar

```
svn switch --relocate From_URL To_URL
```

E.2.23. Criar Repositório Aqui

```
svnadmin create --fs-type fsfs PATH
```

E.2.24. Adicionar

```
svn add PATH...
```

Se seleccionaste uma pasta, o TortoiseSVN primeiro examina-a recursivamente à procura de itens que podem ser adicionados.

E.2.25. Importar

```
svn import -m LogMessage PATH URL
```

A `LogMessage` representa aqui o conteúdo da caixa de edição da mensagem de registo. Esta pode ser vazia.

E.2.26. Responsabilizar

```
svn blame -r N:M -v PATH  
svn log -r N:M PATH
```

Se usares o TortoiseBlame para ver a informação de responsabilidade, é também requerido o ficheiro de registo para mostrar as mensagens de registo numa etiqueta de dica. Se vires a responsabilidade como um ficheiro de texto esta informação não é então requerida.

E.2.27. Adicionar á list de ítems a ignorar

```
svn propget svn:ignore PATH > tempfile  
{edit new ignore item into tempfile}  
svn propset svn:ignore -F tempfile PATH
```

Porque a propriedade `svn:ignore` é frequentemente um valor multi-linha, é mostrado aqui como sendo alterado via um ficheiro de texto em vez de directamente na linha de comandos.

E.2.28. Criar Correção

```
svn diff PATH > patch-file
```

O TortoiseSVN cria um ficheiro de correcção no formato comparação-unificada ao comparar a cópia de trabalho com a sua versão BASE.

E.2.29. Aplicar Correção

Aplicar correcções é um trabalho complicado, a não ser que a correcção e a cópia de trabalho estejam na mesma revisão. Afortunadamente para ti, poderás usar o TortoiseMerge, que não tem equivalente directo no Subversion.

Apêndice F. Detalhes de Implementação

Este anexo contém uma discussão mais detalhada da implementação de algumas das funcionalidades do TortoiseSVN.

F.1. Sobreposição de Ícones

Cada ficheiro e pasta tem um valor de estado no Subversion como é reportado pela biblioteca do mesmo. No cliente de linha de comando, esse é representado por uma letra de código, mas no TortoiseSVN são mostrados graficamente como sobreposições de ícones. Porque o número de sobreposições é muito limitado, cada sobreposição poderá representar vários valores de estado.



A sobreposição *em conflito* é usada para representar o estado em `conflito`, onde uma actualização ou troca resultou em conflitos entre as alterações locais e as alterações descarregadas do repositório. É também usado para indicar o estado `obstruído`, que pode ocorrer quando não é concluída uma operação.



A sobreposição *Modificado* representa o estado `modificado`, quando efectuaste modificações locais, o estado `integrado`, quando as alterações do repositório foram integradas com as alterações locais, e o estado `substituído`, quando o ficheiro foi removido e substituído por outro ficheiro diferente com o mesmo nome.



A sobreposição *Removido* representa o estado `removido`, quando um item foi escalonado para remoção, ou o estado `desaparecido`, quando um item não está presente. Naturalmente um item que está desaparecido não tem ele próprio uma sobreposição, mas a pasta pai pode ser sinalizada quando um dos seus filhos está desaparecido.



A sobreposição *Adicionado* é usado simplesmente para representar o estado `adicionado` quando um item foi adicionado ao controlo de versões.



A sobreposição *No Subversion* é usada para representar um item que está no estado `normal`, ou um item versionado cujo estado não é ainda conhecido. Porque o TortoiseSVN usa um processo de cache em background para recolher o estado, poderá levar alguns segundos a actualizar as sobreposições.



A sobreposição *Necessita bloqueio* é usada para indicar quando um ficheiro tem configurada a propriedade `svn:needs-lock`. Para cópias de trabalho que foram criadas usando o Subversion 1.4.0 e superior, o estado `svn:needs-lock` é guardado localmente em cache pelo Subversion, e isto é usado para determinar quando se deve mostrar esta sobreposição. Para cópias de trabalho no formato pré-1.4, o TortoiseSVN mostra esta sobreposição quando o ficheiro tem o estado só-de-leitura. Tem atenção que o Subversion actualizada automaticamente a versão das cópias de trabalho quando tu as actualizas,

embora poderá não ser efectuada a cache da propriedade `svn:needs-lock` até o próprio ficheiro ser actualizado.



A sobreposição *Bloqueado* é usado quando a cópia de trabalho local contém um bloqueio para esse ficheiro.



A sobreposição *Ignorado* é usado para representar um item que está no estado ignorado, devido ao padrão global de ignorados, ou à propriedade `svn:ignore` da pasta pai. Esta sobreposição é opcional.



A sobreposição *Não versionado* é usado para representar um item que está no estado não versionado. Isto é um item numa pasta versionada, mas que não está ele próprio sob o controlo de versões. Esta sobreposição é opcional.

Se um item tem o estado subversion nenhum (o item não está dentro da cópia de trabalho) então não é mostrado nenhuma sobreposição. Se decidiste desactivar as sobreposições emphasis>Ignorado

Um item só pode ter um valor de estado no Subversion. Por exemplo, um ficheiro pode ser modificado localmente e pode ser marcado para remoção ao mesmo tempo. O Subversion retorna um único valor de estado - neste caso - removido. Essas propriedades estão definidas dentro do próprio Subversion.

Quando o TortoiseSVN mostra recursivamente o estado (configuração por defeito), cada pasta mostra uma sobreposição que reflete o seu próprio estado e o estado de todos os seus filhos. De modo a mostrar uma simples sobreposição de *sumário*, nós usamos a ordem de prioridade mostrada acima para determinar qual a sobreposição a usar, com a sobreposição *Em conflito* a tomar a prioridade mais elevada.

De facto, poderás descobrir que nem todos esse ícones são usados no teu sistema. Isot é porque o número de sobreposições permitidas pelo Windows é limitada a 15. O Windows usa 4 dessas, e as restantes podem ser usadas por outras aplicações. Se não existirem disponíveis em número suficiente sobreposições vagas, o TortoiseSVN tenta ser um bom cidadão (“Good Citizen (TM)”) e limita o seu uso de sobreposições, para dar hipótese às outras aplicações.

- *Normal*, *Modificado* e *Em conflito* são sempre carregadas e visíveis.
- *Removido* é se possível carregado, mas descai para *Modificado* se não existem vagas suficientes.
- *Só-de-Leitura* é se possível carregado, mas descai para *Normal* se não existem vagas suficientes.
- *Bloqueado* só é carregada se existirem menos de 13 sobreposições já carregadas. Descai para *Normal* se não existirem vagas suficientes.
- *Adicionado* é só carregada se existirem menos de 14 sobreposições já carregadas. Descai para *Modificado* se não existirem vagas suficientes.

Apêndice G. Protegendo o Svnserve ao usar o SSH

Esta secção fornece um guia passo-a-passo para configurar o Subversion e o TortoiseSVN para usar o protocolo `svn+ssh`. Se já usas conexões autenticadas por SSH para efectuares login no teu servidor, então já lá estás e poderás encontrar mais detalhes no livro do Subversion. Se não estás a usar o SSH mas gostarias de o fazer para proteger a tua instalação do Subversion este guia fornece-te um método simples, que não envolve criar uma conta de utilizador SSH separada no servidor, para cada utilizador do Subversion.

Nesta implementação criamos uma única conta de utilizador SSH para todos os utilizadores do Subversion, e usar diferentes chaves de autenticação para diferenciar entre os utilizadores reais do Subversion.

Neste anexo assumimos que já instalaste as ferramentas do Subversion, e que criaste um repositório como detalhado algures neste manual. Tem atenção que tu *não* deves arrancar o `svnserve` como um serviço or daemon quando usado com o SSH.

Muita da informação daqui vem de um tutorial providenciado pelo Marc Logemann, que pode ser encontrado em www.logemann.org [<http://www.logemann.org/2007/03/13/subversion-tortoisesvn-ssh-howto/>]. Informação adicional de como montar um servidor Windows é fornecida pelo Thorsten Müller. Obrigado malta!

G.1. Configuração de um Servidor Linux

Tu precisas de ter o SSH activado no servidor e assumimos que irás usar o OpenSSH. Na maioria das distribuições este já está instalado. Para descobrires escreve:

```
ps xa | grep sshd
```

e procura por jobs `ssh`.

Um ponto a salientar é que ao construíres o Subversion a partir das fontes, e não forneceres qualquer parâmetro para o `./configure`, o Subversion cria uma pasta `bin` sob `/usr/local` e coloca aqui os binários. Se queres usar o modo tunel com o SSH deves ter em atenção que o utilizador autenticado via SSH necessita de executar o programa `svnserve` e outros binários. Por esta razão coloca o `/usr/local/bin` na variável `PATH` ou cria links simbólicos dos teus binários para a pasta `/usr/sbin`, ou para qualquer outra pasta que normalmente está no `PATH`.

Para verificares se está tudo OK, autentica-te através do SSH como um utilizador alvo e digita:

```
which svnserve
```

Este comando deverá informar-te se o `svnserve` está disponível.

Cria um novo utilizador que será então usado para aceder ao repositório `svn`:

```
useradd -m svnuser
```

Tem a certeza que dás a este utilizador permissão total de acesso ao repositório.

G.2. Configurar um Servidor Windows

Instala o daemon SSH Cygwin como descrito aqui: <http://pigtail.net/LRP/printsrv/cygwin-sshd.html> SSH Client Tools for use with TortoiseSVN

Cria uma nova conta de utilizador Windows `svnuser`, que será usada para aceder ao repositório. Tem a certeza que dás a este utilizador todos os privilégios de acesso ao repositório.

Se ainda não existe nenhum ficheiro de palavra-passe, cria um a partir da consola do Cygwin usando o:

```
mkpasswd -l > /etc/passwd
```

G.3. Ferramentas de cliente SSH para uso com o TortoiseSVN

Adquire as ferramentas que necessitas para usar o SSH no cliente Windows a partir deste sítio: <http://www.chiark.greenend.org.uk/~sgtatham/putty/> Vai à secção de descarga e obtêm o Putty, Plink, Pageant e Puttygen.

G.4. Criação de certificados OpenSSH

O próximo passo é criar um par de chaves para autenticação. Existem duas maneiras possíveis para criar as chaves. A primeira é criar as chaves com o PuTTYgen no cliente, carregar a chave pública para o teu servidor e usar a chave privada com o PuTTY. A outra é criar o par de chaves com a ferramenta do OpenSSH `ssh-keygen`, descarrega a chave privada para o teu cliente e converte-a numa chave privada ao estilo do PuTTY.

G.4.1. Criar chaves com o ssh-keygen

Entra no teu servidor como `root` ou `svnuser` e digita:

```
ssh-keygen -b 1024 -t dsa -N passphrase -f keyfile
```

substituindo a palavra-passe real (que só tu conheces) e o ficheiro chave. Nós apenas criámos uma chave SSH2 DSA com uma palavra-chave de 1024 bites. Se digitares

```
ls -l keyfile*
```

irás ver dois ficheiros, `keyfile` e `keyfile.pub`. Como podes adivinhar o ficheiro `.pub` é o ficheiro de chave pública, o outro é o da chave privada.

Anexa a chave pública aquelas que estão na pasta `.ssh` dentro da pasta `home svnuser`:

```
cat keyfile.pub >> /home/svnuser/.ssh/authorized_keys
```

De modo a poder usar a chave privada que gerámos, temos de a converter para um formato putty. Isto é porque o formato de ficheiro da chave privada não é especificado por um organismo de standartização. Depois de descarregares para o teu PC cliente o ficheiro da chave privada, arranca o PuTTYgen e vai ao menu `Conversions` → `Import key`. Navega até ao teu ficheiro `keyfile`, do qual obtiveste do servidor a palavra-passe usada aquando da criação da chave. Finalmente clica no `Save private key` e guarda o ficheiro como `keyfile.PPK`.

G.4.2. Criar chaves usando o PuTTYgen

Usa o PuTTYgen para gerar o par chave-pública/chave-privada e guarda-a. Copia a chave pública para o servidor e anexa-a às que estão na pasta `.ssh`, dentro da pasta mãe `svnuser`:

```
cat keyfile.pub >> /home/svnuser/.ssh/authorized_keys
```

G.5. Testar com o PuTTY

Para testar a conexão iremos usar o PuTTY. Arranca o programa e na aba `Session` configura o `hostname` para o nome ou endereço IP do teu servidor, o protocolo para SSH e guarda a sessão como

`SvnConnection`, ou outro nome que preferires. Na aba **SSH** configura versão preferida do protocolo SSH para 2, e na **Auth** fixa o caminho completo para o ficheiro de chave privada `.PPK`, que converteste anteriormente. Volta à aba **Sessions** e pressiona o botão **Save**. Irás agora ver a `SvnConnection` na lista das sessões guardadas.

Clica no **Open** e irás ver uma janela de login ao estilo do telnet. Usa como utilizador o nome `svnuser` e se tudo estiver bem dever-te-ás ligar directamente, sem seres questionado pela palavra-passe.

Poderás necessitar de editar o `/etc/sshd_config` no servidor. Edita as linhas como mostrado em seguida, e depois re-arranca o serviço SSH.

```
PubkeyAuthentication yes
PasswordAuthentication no
PermitEmptyPasswords no
ChallengeResponseAuthentication no
```

G.6. Testar SSH com o TortoiseSVN

Até agora só testámos se consegues efectuar login usando o SSH. Agora precisamos de ter a certeza que a ligação SSH pode efectivamente correr o svnserve. No servidor modifica o `/home/svnuser/.ssh/authorized_keys` do seguinte modo para permitir que vários autores do Subversion usem a mesma conta, `svnuser`. Tem em atenção que cada autor Subversion usa o mesmo login mas com uma chave de autenticação diferente, pelo que terás de adicionar uma linha por cada autor.

Nota: Esta é tudo numa linha muito longa.

```
command="svnserve -t -r <ReposRootPath> --tunnel-user=<author>",
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,
no-pty ssh-rsa <PublicKey> <Comment>
```

Existe vários valores que necessitas de configurar, de acordo com a tua instalação.

`<ReposRootPath>` deverá ser substituído com o caminho para a pasta onde estão os teus repositórios. Isto evita a necessidade de especificar os caminhos completos de servidor nos URLs. Tem atenção que deverás usar barras direitas, mesmo em servidores Windows, e.g. `c:/svn/reposroot`. Nos exemplos abaixo assumimos que tu tens uma pasta no repositório, dentro da raiz, chamada `repos`.

O `<author>` deverá ser substituído pelo autor `svn` que tu queres que seja armazenado na submissão. Isto permite também que o `svnserve` use os seus direitos de acesso dentro do `svnserve.conf`.

O `<PublicKey>` deverá ser substituído pela chave pública que geraste anteriormente.

O `<Comment>` pode ser qualquer comentário que desejares, mas será útil para mapear o nome de um autor `svn` com o nome real da pessoa.

Clica com o botão direito em qualquer pasta no explorador do Windows, e selecciona **TortoiseSVN** → **Navegador-de-repositório**. Serás solicitado a introduzir o URL, que introduzirás deste modo:

```
svn+ssh://svnuser@SvnConnection/repos
```

O que é que significa este URL? O nome da Schema é `svn+ssh` que diz ao TortoiseSVN como lidar com os pedidos ao servidor. Depois da dupla barra especifica o utilizador a ligar-se ao servidor, que é no nosso caso `svnuser`. Depois do `@` fornecemos o nome da nossa sessão PuTTY. Este nome de sessão contém todos os detalhes, como por exemplo, onde encontrar a chave privada, o IP e DNS do servidor. Por último temos de fornecer o caminho para o repositório, relativamente à raiz do repositório no servidor, como especificado no ficheiro `authorized_keys`.

Clica no OK, e deverás conseguir navegar no conteúdo do repositório. Se tal acontecer, significa que tens um tunel SSH em conjunto com o TortoiseSVN.

Tem em atenção que, por defeito, o TortoiseSVN usa a sua própria versão do Plink para se ligar. Isto evita que, para cada tentativa de autenticação, apareça uma janela de consola, mas também significa que não aparecerá em nenhum lado mensagens de erro. Se receberes um erro “Não foi possível escrever para a saída padrão”, poderás tentar especificar o Plink como cliente nas preferências de rede do TortoiseSVN. Isto permite-a ver as mensagens de erro reais, geradas pelo Plink.

G.7. Variantes da Configuração do SSH

Uma maneira de simplificar o URL no TortoiseSVN é configurar o utilizador dentro da sessão PuTTY. Para tal terás de carregar no PuTTY as tuas sessões já definidas `SvnConnection` e na aba **Connection** inserir para nome de utilizador `Auto login user`, e.g. `svnuser`. Guarda de novo a tua sessão PuTTY e tenta o seguinte URL no TortoiseSVN:

```
svn+ssh://SvnConnection/repos
```

Desta vez só fornecemos a sessão PuTTY `SvnConnection` ao cliente SSH que o TortoiseSVN usa (TortoisePlink.exe). Este cliente irá verificar na sessão todos os detalhes necessários.

Até à altura da escrita o PuTTY na verificava todas as configurações guardadas, pelo que se tiveres múltiplas configurações para o mesmo nome de servidor, ele irá pegar na primeira que condizer com este. Se também editares a configuração por defeito e a guardares, o nome de utilizador `auto login` *não* será guardado.

Muitas pessoas gostam de usar o Pageant para armazenarem as suas chaves. Porque uma sessão de PuTTY é capaz de armazenar uma chave, tu nem sempre precisas do Pageant. Mas imagina que tu queres armazenar chaves para vários servidores diferentes; nesse caso terias de editar a sessão do Putty vez após vez, para cada servidor que estejas a tentar-te ligar. Nessa situação o Pageant tem todo o sentido porque quando o PuTTY, Plink, TortoisePlink e qualquer outra ferramenta baseada em PuTTY está a tentar-se ligar a um servidor SSH, esta verifica todas as chaves privadas que o Pageant armazena, para iniciar a ligação.

Para esta tarefa corre simplesmente o Pageant e adiciona a chave privada. Deverá ser a mesma chave privada que tu definiste acima na sessão do PuTTY. Se usares o Pageant para armazenar a chave privada, tu podes remover na tua sessão de PuTTY guardada, a referência ao ficheiro de chave privada. Tu podes adicionar mais chaves para outros servidores ou, é claro, para outros utilizadores.

Se não queres repetir este procedimento após cada reinício do sistema do teu cliente, deverás colocar o Pageant no grupo de auto-arranque do teu sistema Windows. Tu podes acrescentar as chaves com caminhos completos, como parâmetros de linha de comandos do Pageant.exe.

A última maneira de se conectar a um servidor SSH, será simplesmente através do uso deste URL dentro do TortoiseSVN:

```
svn+ssh://svnuser@100.101.102.103/repos  
svn+ssh://svnuser@mydomain.com/repos
```

Como podes ver, não usamos uma sessão PuTTY guardada, mas um endereço IP (ou nome de domínio) como o destino da ligação. Também fornecemos o utilizador, mas poderás perguntar como será encontrado o ficheiro de chave privada. Porque o TortoisePlink.exe é apenas uma versão modificada da ferramenta Plink padrão da suite PuTTY, o TortoiseSVN irá também tentar todas as chaves armazenadas no Pageant.

Se usares o último método, tem a certeza que não tens configurado por defeito um nome de utilizador no PuTTY. Temos tido relatórios de um problema no PuTTY que causa, neste caso, perdas de ligação.

Para remover o utilizador por defeito, simplesmente limpa a `HKEY_CURRENT_USER\Software\SimonTatham\Putty\Sessions\Default%20Settings\HostName`

Glossário

Actualizar	Este comando do Subversion adquire as últimas alterações do repositório para a tua cópia de trabalho, integrando quaisquer alterações efectuadas por outros com as alterações locais.
Adicionar	Comando Subversion que é usado para adicionar um ficheiro ou pasta à tua cópia de trabalho. Os novos itens são adicionados ao repositório quando os submeteres.
BDB	BD Berkeley. Uma base de dados extensamente testada em aplicações de "backend" que não é utilizável em "shares" de rede. Opção automática para repositórios anteriores à versão 1.2.
Bloquear	Quando efectuas um bloqueio no item versionado, tu irás marcá-lo no repositório como não submetível, à excepção da cópia de trabalho de onde foi efectuado o bloqueio.
Checkout	Comando Subversion que cria uma cópia de trabalho local numa simples pasta, através da descarga de ficheiros versionados a partir do repositório.
Comparar	Abreviatura para "Mostrar Diferenças". Muito útil quando queres ver exactamente que alterações foram feitas.
Conflito	Quando as alterações do repositório são integradas com as tuas alterações locais, por vezes essas alterações ocorrem nas mesmas linhas. Neste caso o Subversion não pode decidir automaticamente qual a versão a usar e o ficheiro é tido como em conflito. Terás de editar manualmente esse ficheiro e resolver o conflito antes de submeteres posteriormente mais alterações.
Cópia	No repositório Subversion tu podes criar uma cópia de um simples ficheiro ou de uma árvore inteira. Essas são implementadas como "cópias baratas" que funcionam um pouco como um apontador para o original, pelo que não ocupam quase nenhum espaço. Criar cópias preserva o histórico do item na cópia, para que possas seguir as alterações efectuadas antes de a teres feito.
Cópia de Trabalho	Esta é a tua "caixa de areia" local, a área onde tu trabalhas nos ficheiros versionados que normalmente reside no teu disco local. Tu crias uma cópia local a partir do repositório ao efectuar o comando "Checkout", e envias as tuas alterações de volta para o repositório com o comando "Submeter".
Exportar	Este comando produz uma cópia de uma pasta versionada, tal como uma cópia de trabalho, mas sem as pastas locais <code>.svn</code> .
FSFS	É um sistema de ficheiros proprietário do Subversion para suporte aos repositórios. Pode ser usado em partilhas de rede. Por defeito para repositórios na versão 1.2 e nos mais recentes.
GPO	Objecto de política de grupo
História	Mostra o histórico de revisões de um ficheiro ou pasta. Também conhecido como "Registo".
Importar	Comando do Subversion para importar uma hierarquia completa de pastas, para o repositório numa única revisão.

Integrar	<p>O processo pelo qual as alterações do repositório são adicionadas à tua cópia de trabalho destruir quaisquer alterações que tenhas efectuado localmente. Por vezes essas alterações não podem ser conciliadas automaticamente e, é dito que a cópia de trabalho está em conflito.</p> <p>A integração ocorre automaticamente quando actualizas a tua cópia de trabalho. Tu podes também integrar alterações específicas de outro ramo usando o comando TortoiseSVN's Merge.</p>
Limpar	<p>Para citar o livro do Subversion: “ Recursively clean up the working copy, removing locks and resuming unfinished operations. If you ever get a <i>working copy locked</i> error, run this command to remove stale locks and get your working copy into a usable state again. ” Tem atenção que neste contexto o <i>lock</i> (bloqueio) refere-se ao bloqueio do sistema de ficheiros local e não ao bloqueio do repositório.</p>
Propriedade	<p>Além de versionar as tuas pastas e ficheiros, o Subversion permite-te adicionar metadados versionados - referidos como “propriedades” a cada um das tuas pastas e ficheiros versionados. Cada propriedade tem um nome e valor, tal como uma chave do registo. O Subversion tem algumas propriedades especiais que usa internamente, como as <code>svn:eol-style</code>. O TortoiseSVN também tem algumas, tais como <code>tsvn:logminsize</code>. Tu podes também adicionar as tuas próprias propriedades com qualquer nome e valor que escolhas.</p>
Propriedade de Revisão (revprop)	<p>Tal como os ficheiros podem ter propriedades, assim pode cada revisão no repositório. Algumas revprops especiais são automaticamente adicionadas quando é criada a revisão, nomeadamente: <code>svn:date</code> <code>svn:author</code> <code>svn:log</code> que representa a data/hora de submissão, o nome de quem submeteu e a respectiva mensagem de registo. Essas propriedades podem ser editadas, mas não estão versionadas pelo que qualquer alteração é permanente, e não poderá ser desfeita.</p>
Ramo	<p>Um termo usado frequentemente em sistemas de controlo de versões para descrever o que acontece quando o desenvolvimento se divide, no determinado ponto, e segue dois caminhos separados. Tu podes criar um ramo a partir da linha principal de desenvolvimento, para o desenvolvimento de uma nova funcionalidade sem tornar instável a linha principal. Ou tu podes criar um ramo estável de uma entrega para o qual só efectuas reparações de problemas, enquanto novos desenvolvimentos têm lugar no instável trunk. No Subversion um ramo é implementado como “cópia barata”.</p>
Registo	<p>Mostra o histórico de revisões de um ficheiro ou pasta. Também conhecido como “Histórico”.</p>
Remendo	<p>Se uma cópia de trabalho tem alterações, só em ficheiros de texto, é possível usar o comando <code>comparar</code> do Subversion para gerar único ficheiro de sumário dessas alterações, no formato comparação-unificada. Um ficheiro desse tipo é normalmente referido como “Correcção”, e pode ser enviado por email para alguém (ou para uma lista de correio) e aplicado noutra cópia de trabalho. Alguém sem permissão para submeter pode efectuar as alterações, e submeter o ficheiro de correcção para alguém com autorização para submeter, para seja este a aplicar a correcção. Ou se não estiveres seguro da alteração efectuada, podes submeter a correcção para que outros a possam rever.</p>

Remover	Quando tu removes um item versionado (e submetes a alteração) o item deixa de existir no repositório, após a revisão submetida. Mas é claro que continuará a existir em revisões anteriores do repositório, pelo que poderás continuar a acedê-lo. Se necessário, tu podes copiar um item removido e “ressuscitá-lo” por completo com histórico.
Reposicionar	<p>Se o teu repositório mudou de localização, talvez porque o alteraste para uma pasta diferente no teu servidor, ou o nome de domínio do servidor foi alterado, será necessário “reposicionar” a tua cópia de trabalho para que o seu URL de repositório aponte para a nova localização.</p> <p>Nota: só deverás usar este comando se a tua cópia de trabalho se refere à mesma localização no mesmo repositório, mas apenas o próprio repositório foi movido. Em qualquer outra circunstância, o que provavelmente necessitarás, é do comando “Trocar”.</p>
Repositório	Um repositório e um local central onde são armazenados e mantidos dados. Um repositório pode ser um local onde múltiplas bases de dados ou ficheiros estão localizados para distribuição pela rede, ou poderá também ser um local que está acessível ao utilizador directamente, sem o obrigar a navegar pela rede.
Resolver	Quando na cópia de trabalho, são deixados ficheiros num estado de conflito após uma integração, esses conflitos têm de ser observados por um humano com o recurso a um editor (ou talvez o TortoiseMerge). Este processo é referido como “Resolução de Conflitos”. Quando estiver concluído, poderás marcar os ficheiros em conflito como estando resolvidos, o que permite que estes possam ser submetidos.
Responsabilizar	Este comando dirige-se apenas a ficheiros de texto, e anota cada linha de modo a mostrar a revisão do repositório em que foi alterada por último, o autor que efectuou a alteração. A nossa implementação de GUI chama-se TortoiseBlame, e também mostra a data/Hora de submissão e a mensagem de registo, quando passas com o rato sobre o número de revisão.
Reverter	O Subversion mantém um cópia “prístina” local de cada ficheiro como era, quando tu actualizaste pela última vez a tua cópia de trabalho. Se efectuaste alterações e decidiste que queres desfazê-las, podes então usar o comando “reverter” para retornares à cópia prístina.
Revisão	<p>Cada vez que submetes um conjunto de alterações, tu crias uma nova “revisão” no repositório. Cada revisão representa o estado da árvore do repositório, num determinado ponto da sua história. Se quiseres voltar atrás, poderás examinar o repositório tal e qual estava na revisão N.</p> <p>De outro modo, uma revisão refere-se a um conjunto de alterações que foram efectuadas, quando essa mesma revisão foi criada.</p>
Revisão BASE	A revisão base corrente para o ficheiro ou pasta na tua <i>cópia de trabalho</i> . Esta é a revisão que o ficheiro ou pasta tinha na última execução do checkout, actualização ou submissão. A revisão BASE é normalmente diferente da revisão HEAD.
Revisão HEAD	A última revisão de um ficheiro ou pasta no <i>repositório</i> .

Submeter	Este comando Subversion é usado para transferir as alterações, na tua cópia de trabalho local, de volta para o repositório, criando uma nova revisão do mesmo.
SVN	Uma abreviatura para Subversion, frequentemente utilizada. O nome do protocolo personalizado do Subversion, usado pelo servidor de repositório “svnserve”.
Trocar	Tal como o “Actualizar-para-revisão” altera a janela temporal da cópia de trabalho, para olhar para um ponto diferente na história, o “Trocar” altera a janela de espaço da cópia de trabalho, para que aponte para uma parte diferente do repositório. É particularmente útil quando se está a trabalhar com o trunk e com ramos, onde apenas alguns ficheiros diferem. Podes assim trocar a tua cópia de trabalho entre esses dois e só os ficheiros alterados serão transferidos.

Índice Remissivo

Símbolos

"plugin", 175
"Shell" de Windows, 1
'cópia de trabalho' não versionada, 127

A

a bloquear, 112
acções lado-do-servidor, 120
Aceder, 17
actualizar, 52, 181
adicionar, 82
adicionar ficheiros ao repositório, 41
alteração de tamanho de letra, 89
alterações, 182
anotar, 117
Apache, 26
arrastar, 39
arrastar com o botão direito, 39
arrastar-e-largar, 39
atalho, 184
autenticação, 40
Autenticação múltipla, 33
auto-props, 95
automação, 188, 191
Autorização, 31

B

bugtracker, 130

C

cache de registo, 155
caminhos UNC, 17
caracteres de substituição, 85
CLI, 193
cliente de linha de comando, 193
clique com o botão direito, 37
colunas do Explorador, 59
COM, 166, 175
COM SubWCRRev interface, 168
comparar, 62, 77
comparar ficheiros, 182
comparar imagens, 81
comparar pastas, 182
comparar revisões, 79
conflito, 9, 54
conflito de árvore, 54
controlador de assuntos, 130, 175
controlador de domínio, 185
controlador de problemas, 130
controlo de problema, 130
Controlo de versões, 1
cópia, 100, 120
cópia de segurança, 19

cópia de trabalho, 10
copiar ficheiros, 83
correção, 116
corretor ortográfico, 3
correspondência de padrões, 85
Criar
 Cliente de linha de comando, 16
 TortoiseSVN, 16
criar repositório, 16
criar uma cópia de trabalho, 44
culpa, 117

D

dar entrada, 46
desfazer, 90
desfazer alteração, 181
desfazer submissão, 181
desligar do repositório, 184
desversionar, 129, 184
dicionário, 3
dif, 77, 116
diff unificado, 116
domaincontroller, 31
Domínio Windows, 31

E

editar registo/autor, 72
elogiar, 117
entradas do menu de contexto, 186
enviar alterações, 46
estado, 58, 60
estado da cópia de trabalho, 58
estatísticas, 73
etiqueta, 83, 100
expande as palavras-chave, 93
explorador, 1
exportar, 127
exportar alterações, 79
externos, 98, 182
extracção de versão, 166

F

FAQ, 179
ferramentas de comparação, 82
ferramentas de integração, 82
ficheiros especiais, 43
ficheiros temporários, 41
ficheiros/pastas não-versionadas, 84
filtro, 73
funções desactivadas, 186

G

ganchos, 20
ganchos de cliente, 159
GPO, 185
gráfico, 123
gráfico de revisões, 123

H

hiperligação, 20
hiperligação de checkout, 20
hiperligação TortoiseSVN, 20
história, 64

I

IBugtraqProvider, 175
ícones, 58
ignorados globais, 136
ignorar, 84
Importar, 41
importar no local, 43
Índice de projectos, 30
instalar, 3
Instalar, 185
integrar, 103

- duas árvores, 107
- faixa de revisão, 105
- re-integrar, 106

integrar conflitos, 110

L

limpeza, 91
linha de comando, 188, 191
lista de alterações, 62
Livro sobre Subversion, 5

M

marcar versão, 100
maximizar, 41
mensagem de registo, 180
mensagem de submissão, 180
mensagem vazia, 180
mensagens de registo, 64
mensagens de submissão, 64
menu de contexto, 37
modificações, 60
mod_authz_svn, 28, 31
Monitorizar a integração, 110
mover, 88, 180
mover ficheiros, 83
msi, 185

N

navegador-de-repositório, 120
NTLM, 31
números de versão nos ficheiros, 166

O

obter alterações, 52

P

pacotes de linguas, 3
padrão de exclusão, 136
palavras-chave, 93

Partilha de rede, 17
pasta .svn, 165
pasta _svn, 165
políticas de grupo, 185, 186
preferências, 135
prioridade de sobreposição, 199
projectos ASP, 186
projectos comuns, 182
projectos proprietários, 182
propriedades, 91
propriedades de projecto, 96
propriedades de revisão, 72
Propriedades do TortoiseSVN, 96
Propriedades Subversion, 92

R

ramo, 83, 100
re-integrar integração, 111
readonly, 112
rebobinar, 181
registo, 64, 163
registo de controlo de integração, 71
remover, 87, 87
remover versionamento, 184
renomear, 88, 120, 180
renomear ficheiros, 83
reorganizar, 180
reposicionar, 129
repositório, 5, 41
repositórios externos, 98
resolver, 54
reverter, 90, 181
revisão, 12, 123
revprops, 72

S

SASL, 25
scripts de gancho, 20, 159
scripts de gancho do lado do servidor, 20
servidor movido, 129, 129
servidor proxy, 149
sítio de internet, 20
sobreposições, 58, 199
sons, 135
SSL, 34
SSPI, 31
submiter, 46
SubWCRev, 166
SVN exportar, 44
SVNParentPath, 29, 30
SVNPath, 29
svnserv, 21, 23
SVN_ASP_DOT_NET_HACK, 186

T

TortoiseIDiff, 81
traduções, 3

trocar, 102

U

unidades SUBST, 148

URL alterado, 129

URL do repositório alterado, 129

V

ver alterações, 58

verificar actualizações, 185

verificar nova versão, 185

versão, 185

versionar ficheiros novos, 82

ViewVC, 134

visualizador de repositório, 134

visualizador de servidor, 120

visualizador web, 134

VS2003, 186

W

WebDAV, 26

WebSVN, 134