

# **TortoiseSVN**

**Cliente de Subversion para Windows**

**Version 1.12**

**Stefan Küng  
Lübbe Onken  
Simon Large**

---

# **TortoiseSVN: Cliente de Subversion para Windows: Version 1.12**

por Stefan Küng, Lübbe Onken e Simon Large

Tradução: Paulo M. Santana Justino, Luis X. B. Mourão

data de publicação 2019/03/11 21:25:21 (r28524)

---

# Índice

Prefácio .....	xi
1. O que é o TortoiseSVN? .....	xi
2. Funcionalidades do TortoiseSVN .....	xi
3. Licença .....	xii
4. Desenvolvimento .....	xii
4.1. A história do TortoiseSVN .....	xiii
4.2. Agradecimentos .....	xiii
5. Guia de leitura .....	xiii
6. Terminologia utilizada neste documento .....	xiv
1. Começar .....	1
1.1. Instalar o TortoiseSVN .....	1
1.1.1. Requisitos de sistema .....	1
1.1.2. Instalação .....	1
1.2. Conceitos básicos .....	1
1.3. Ir para um teste de condução .....	2
1.3.1. Criar um repositório .....	2
1.3.2. Importar um projeto .....	2
1.3.3. Verificar uma cópia de trabalho .....	3
1.3.4. Fazer alterações .....	4
1.3.5. Adicionar mais ficheiros .....	4
1.3.6. Ver a história do projeto .....	5
1.3.7. Desfazer alterações .....	5
1.4. Avançar... .....	6
2. Conceitos básicos de controlo de versões .....	7
2.1. O repositório .....	7
2.2. Modelos de Controlo de Versões .....	7
2.2.1. O Problema da Partilha de Ficheiros .....	8
2.2.2. A Solução Bloquear-Modificar-Desbloquear .....	8
2.2.3. A solução Copiar-Modificar-Integrar .....	9
2.2.4. O que faz o Subversion? .....	12
2.3. O Subversion em Acção .....	12
2.3.1. Cópias de Trabalho. ....	12
2.3.2. URLs do Repositório .....	14
2.3.3. Revisões .....	14
2.3.4. Como as Cópias de Trabalho Seguem o Repositório .....	16
2.4. Resumo .....	16
3. O repositório .....	18
3.1. Criação do Repositório .....	18
3.1.1. Criar um Repositório com o Cliente de Linha de Comandos .....	18
3.1.2. Criar O Repositório Com o TortoiseSVN .....	18
3.1.3. Acesso Local ao Repositório .....	19
3.1.4. Aceder ao Repositório numa Partilha de Rede .....	19
3.1.5. Estrutura do Repositório .....	19
3.2. Cópia de Segurança do Repositório .....	21
3.3. Scripts de gancho de servidor .....	21
3.4. Ligações de Checkout .....	22
3.5. Aceder ao Repositório .....	22
4. Guia de Uso Diário .....	24
4.1. Funcionalidades gerais .....	24
4.1.1. Sobreposição de Ícones .....	24
4.1.2. Menus de Contexto .....	24
4.1.3. Arrastar e Largar .....	26
4.1.4. Atalhos comuns .....	27
4.1.5. Autenticação .....	27
4.1.6. Maximizar Janelas .....	28

---

4.2. Importar Dados Para Um Repositório .....	29
4.2.1. Importar .....	29
4.2.2. Importar no local .....	30
4.2.3. Ficheiros Especiais .....	31
4.3. SVN Exportar Para Uma Cópia de Trabalho .....	31
4.3.1. Profundidade do Checkout .....	32
4.4. Submetendo as tuas alterações para o Repositório .....	34
4.4.1. A Caixa de Diálogo Submeter .....	34
4.4.2. Listas de Alterações .....	37
4.4.3. Submeter apenas secções de ficheiros .....	37
4.4.4. Excluir itens da lista a Submeter .....	38
4.4.5. Mensagens de Registo de Submeter .....	38
4.4.6. Progresso do Submeter .....	40
4.5. Actualizar a Tua Cópia de Trabalho Com Alterações de Outros .....	41
4.6. Resolvendo Conflitos .....	43
4.6.1. Conflitos de Ficheiro .....	43
4.6.2. Conflitos de Propriedade .....	44
4.6.3. Conflitos de Árvore .....	44
4.7. Obter informação de Estado .....	47
4.7.1. Sobreposição de Ícones .....	47
4.7.2. Estado detalhado .....	49
4.7.3. Estado Remoto e Local .....	50
4.7.4. Vendo diferenças .....	52
4.8. Listas de Alterações .....	53
4.9. Shelving .....	55
4.10. Caixa de Diálogo Registo de Revisões .....	57
4.10.1. Invocando a Caixa de Diálogo Registo de Revisão .....	58
4.10.2. Acções de Registo de Revisões .....	58
4.10.3. Obtendo Informação Adicional .....	60
4.10.4. Obtendo mais mensagens de registo .....	66
4.10.5. Revisão Actual da Cópia de Trabalho .....	67
4.10.6. Funcionalidades de Rastreamento de Integração .....	67
4.10.7. Alterando a Mensagem de Registo e Autor .....	68
4.10.8. Filtrando Mensagens de Registo .....	69
4.10.9. Informação Estatística .....	70
4.10.10. Modo Fora de Linha .....	74
4.10.11. Refrescar a Vista .....	74
4.11. Ver Diferenças .....	75
4.11.1. Diferenças em ficheiros .....	75
4.11.2. Opções de Fim-de-Linha e Espaços-Brancos .....	76
4.11.3. Comparando Pastas .....	76
4.11.4. Comparando Imagens usando o TortoiseDiff .....	78
4.11.5. Comparando Documentos Office .....	79
4.11.6. Ferramentas de Comparação/Integração .....	79
4.12. Adicionar Novos Ficheiros e Pastas .....	80
4.13. Copiando/Movendo/Renomeando Ficheiros e Pastas .....	80
4.14. Ignorando Ficheiros E Pastas .....	82
4.14.1. Correspondência de Padrões em Listas de Ignorados .....	82
4.15. Removendo, Movendo e Renomeando .....	83
4.15.1. Removendo ficheiros e pastas .....	84
4.15.2. Movendo ficheiros e pastas .....	85
4.15.3. Lidando com conflitos de maiúsculas e minúsculas no nome do ficheiro .....	85
4.15.4. Reparando Renomeações de Ficheiros .....	86
4.15.5. Removendo Ficheiros Não Versionados .....	86
4.16. Desfazer Alterações .....	86
4.17. Limpar .....	88
4.18. Configurações de Projecto .....	89
4.18.1. Propriedades Subversion .....	89

---

4.18.2. Propriedades de Projecto TortoiseSVN .....	93
4.18.3. Editores de Propriedades .....	98
4.19. Itens Externos .....	104
4.19.1. Pastas Externas .....	104
4.19.2. Ficheiros Externos .....	106
4.19.3. Creating externals via drag and drop .....	107
4.20. Ramificando/Etiquetando .....	107
4.20.1. Criando um Ramo ou Etiqueta .....	107
4.20.2. Outras formas de criar um ramo ou etiqueta .....	110
4.20.3. SVN Exportar ou Trocar... .....	110
4.21. Integrar .....	111
4.21.1. A Integrar Um Intervalo de Revisões .....	112
4.21.2. A Integrar Duas Árvores Diferentes .....	115
4.21.3. Opções de Integração .....	116
4.21.4. Rever os Resultados de Integração .....	116
4.21.5. Rastreamento de Integração .....	117
4.21.6. Handling Conflicts after Merge .....	118
4.21.7. Manutenção do Ramo de Funcionalidade .....	119
4.22. A bloquear .....	120
4.22.1. Como Funciona o Sistema de Bloqueio no Subversion .....	120
4.22.2. Obter um Bloqueio .....	121
4.22.3. Libertar um Bloqueio .....	122
4.22.4. Verificar o Estado dos Bloqueio .....	122
4.22.5. Tornar os Ficheiros Não-Bloqueados Só de Leitura .....	123
4.22.6. Os Scripts do Gancho de Bloqueio .....	123
4.23. Criar e Aplicar Correções .....	123
4.23.1. Criar um Ficheiro de Correção .....	123
4.23.2. Aplicar um Ficheiro de Correção .....	125
4.24. Quem Alterou Que Linha .....	125
4.24.1. Responsabilidade para Ficheiros .....	126
4.24.2. Diferenças de responsabilidade .....	128
4.25. O Navegador de Repositório .....	128
4.26. Grafo de Revisões .....	131
4.26.1. Nós do Grafo de Revisões .....	132
4.26.2. Alterando a Vista .....	132
4.26.3. Utilizar o Gráfico .....	134
4.26.4. Refrescar a Vista .....	135
4.26.5. Podar as Árvores .....	135
4.27. Exportar uma Cópia de Trabalho do Subversion .....	136
4.27.1. Remover uma cópia de trabalho do controlo de versões .....	137
4.28. Reposicionar uma cópia de trabalho .....	137
4.29. Integração com Sistemas de controlo de problemas/Gestores de Problemas .....	138
4.29.1. Adicionar Números de Problemas nas Mensagens de Registo .....	139
4.29.2. Obter Informações do Gestor de Problemas .....	142
4.30. Integração com visualizadores de repositório Web-based .....	143
4.31. Preferências do TortoiseSVN .....	144
4.31.1. Preferências Gerais .....	144
4.31.2. Preferências do Gráfico de Revisões .....	153
4.31.3. Preferências de Sobreposição de Ícones .....	156
4.31.4. Preferências de Rede .....	160
4.31.5. Preferências de Programas Externos .....	162
4.31.6. Preferências de Dados Guardados .....	167
4.31.7. Cache de Registo .....	168
4.31.8. Scripts de Gancho do Lado do Cliente .....	171
4.31.9. Preferências do TortoiseBlame .....	176
4.31.10. TortoiseUDiff Settings .....	177
4.31.11. Exportando preferências TSVN .....	178
4.31.12. Preferências Avançadas .....	178

---

4.32. Passo Final .....	183
5. Project Monitor .....	185
5.1. Adding projects to monitor .....	185
5.2. Monitor dialog .....	186
5.2.1. Main operations .....	186
6. O Programa SubWCRev .....	188
6.1. A Linha de Comando SubWCRev .....	188
6.2. Substituição de Palavra-Chave .....	190
6.3. Exemplo de Palavra-Chave .....	191
6.4. Interface COM .....	193
7. Interface IBugtraqProvider .....	196
7.1. Convenções de nomenclatura .....	196
7.2. O interface do IBugtraqProvider .....	196
7.3. A interface IBugtraqProvider2 .....	198
A. Questões Mais Frequentes (FAQ) .....	201
B. Como Farei Para... .....	202
B.1. Mover/copiar muitos ficheiros de uma vez só .....	202
B.2. Forçar utilizadores a introduzir uma mensagem de registo .....	202
B.2.1. Script-gancho no servidor .....	202
B.2.2. Propriedades do projecto .....	202
B.3. Actualizar ficheiros seleccionados a partir do repositório .....	202
B.4. Reverter (Anular) revisões no repositório .....	203
B.4.1. Usar a caixa de diálogo registo de revisão .....	203
B.4.2. Usa a caixa de diálogo integrar .....	203
B.4.3. Usa o <code>svndumpfilter</code> .....	203
B.5. Comparar duas revisões de um ficheiro ou pasta .....	204
B.6. Incluir um subprojecto comum .....	204
B.6.1. Usa o <code>svn:externals</code> .....	204
B.6.2. Usar uma cópia de trabalho aninhada. ....	204
B.6.3. Usa uma localização relativa .....	205
B.6.4. Adicionar projecto ao repositório .....	205
B.7. Criar um atalho para um repositório .....	205
B.8. Ignorar ficheiros que já estão versionados .....	205
B.9. Remover uma cópia de trabalho do controlo de versões .....	206
B.10. Remover uma cópia de trabalho .....	206
C. Dicas Úteis para Administradores .....	207
C.1. Instalar o TortoiseSVN via politicas de grupo .....	207
C.2. Redireccionar a verificação de actualização .....	207
C.3. Configurar a variável de ambiente <code>SVN_ASP_DOT_NET_HACK</code> .....	208
C.4. Desactivar entradas do menu de contexto .....	208
D. Automatizar o TortoiseSVN .....	211
D.1. Comandos TortoiseSVN .....	211
D.2. manípulo URL do <code>Tsvncmd</code> .....	217
D.3. Comandos TortoiseIDiff .....	218
D.4. TortoiseUDiff Commands .....	219
E. Referência Cruzada da Interface de Linha de Comandos .....	220
E.1. Convenções e Regras Básicas .....	220
E.2. Comandos TortoiseSVN .....	220
E.2.1. Checkout .....	220
E.2.2. Actualizar .....	220
E.2.3. Actualizar para Revisão .....	221
E.2.4. Submeter .....	221
E.2.5. Comparar .....	221
E.2.6. Mostrar Registo .....	222
E.2.7. Verificar Modificações .....	222
E.2.8. Gráfico de Revisões .....	222
E.2.9. Navegador de Repositório .....	222
E.2.10. Editar Conflitos .....	223

---

---

E.2.11. Resolvido .....	223
E.2.12. Alterar nome .....	223
E.2.13. Remover .....	223
E.2.14. Reverter .....	223
E.2.15. Limpar .....	223
E.2.16. Obter "Lock" .....	223
E.2.17. Libertar "Lock" .....	224
E.2.18. Ramo/Etiqueta .....	224
E.2.19. Trocar .....	224
E.2.20. Integrar .....	224
E.2.21. Exportar .....	225
E.2.22. Reposicionar .....	225
E.2.23. Criar Repositório Aqui .....	225
E.2.24. Adicionar .....	225
E.2.25. Importar .....	225
E.2.26. Responsabilizar .....	225
E.2.27. Adicionar á list de ítems a ignorar .....	225
E.2.28. Criar Correção .....	226
E.2.29. Aplicar Correção .....	226
F. Detalhes de Implementação .....	227
F.1. Sobreposição de Ícones .....	227
G. Pacotes de Linguagens e Verificadores de Sintaxe .....	229
G.1. Pacotes de Linguas .....	229
G.2. Corretor Ortográfico .....	229
Glossário .....	231
Índice Remissivo .....	234

---

## Lista de Figuras

1.1. O menu TortoiseSVN para pastas não versionadas .....	2
1.2. Caixa de diálogo Importar .....	3
1.3. Visualizador de diferenças em ficheiros .....	4
1.4. A caixa de diálogo Registo .....	5
2.1. Um sistema típico Cliente/Servidor .....	7
2.2. O Problema a Evitar .....	8
2.3. A Solução Bloquear-Modificar-Desbloquear .....	9
2.4. A solução Copiar-Modificar-Integrar .....	10
2.5. ...Continuação do Copiar-Modificar-Integrar .....	11
2.6. O Sistema de Ficheiros do Repositório .....	13
2.7. O repositório .....	15
3.1. O menu TortoiseSVN para pastas não versionadas .....	18
4.1. O explorador mostrando os ícones sobrepostos .....	24
4.2. Menu de Contexto de uma pasta sob controlo de versões. ....	25
4.3. Menu de ficheiro do Explorador para um atalho numa pasta versionada. ....	26
4.4. Menu arrastar com o botão direito para uma pasta sob controlo de versões. ....	27
4.5. Caixa de diálogo de autenticação .....	28
4.6. Caixa de diálogo Importar .....	30
4.7. A caixa de diálogo Checkout .....	32
4.8. A Caixa de Diálogo Submeter .....	35
4.9. A Verificação de Sintaxe na Caixa de Diálogo Submeter .....	39
4.10. A caixa de dialogo de Progresso, mostrando a submissão em progresso .....	40
4.11. Caixa de diálogo de progresso mostrando uma actualização finalizada. ....	41
4.12. O explorador mostrando os ícones sobrepostos .....	47
4.13. Página de propriedades do explorador, aba do Subversion .....	49
4.14. Verificar Modificações .....	50
4.15. Caixa de diálogo Submeter com Listas de Alterações .....	54
4.16. Shelve dialog .....	56
4.17. Unshelve dialog .....	57
4.18. A Caixa de Diálogo Registo de Revisão .....	58
4.19. O Painel de Topo da Caixa de Diálogo Registo de Revisões com Menu de Contexto .....	60
4.20. The Code Collaborator Settings Dialog .....	63
4.21. Menu de Contexto do Painel Superior para 2 revisões seleccionadas .....	63
4.22. O Painel de fundo da Caixa de Diálogo Registo com o Menu de Contexto .....	64
4.23. The Log Dialog Bottom Pane with Context Menu When Multiple Files Selected. ....	65
4.24. A Caixa de Diálogo Registo Mostra Rasto das Revisões de Integração .....	68
4.25. Histograma de Submissões-por-Autor .....	71
4.26. Gráfico de queijo Submissões-por-Autor .....	72
4.27. Gráfico de Submissões-por-data .....	73
4.28. Caixa de diálogo Colocar-se em Fora de Linha .....	74
4.29. A Caixa de Diálogo Comparar Revisões .....	77
4.30. O leitor de diferenças de imagem .....	78
4.31. Menu de contexto do Explorador para ficheiros não versionados .....	80
4.32. Menu arrastar com o botão direito para uma pasta sob controlo de versões. ....	81
4.33. Menu de contexto do Explorador para ficheiros não versionados .....	82
4.34. Menu de contexto do Explorador para ficheiros versionados .....	84
4.35. Caixa de Diálogo Reverter .....	87
4.36. The Cleanup dialog .....	88
4.37. Página de propriedades do Subversion .....	89
4.38. Adicionando propriedades .....	91
4.39. Caixa de diálogo de propriedades para scripts gancho .....	96
4.40. Caixa de diálogo da propriedade tipos de utilizador booleano .....	96
4.41. Página de propriedade svn:externals .....	98
4.42. Página de propriedade svn:keywords .....	99
4.43. Página de propriedade svn:eol-style .....	99



---

4.44. Página de propriedade tsvn:bugtraq .....	100
4.45. Página de propriedade do tamanho das mensagens de registo .....	101
4.46. Página de propriedade da Língua .....	101
4.47. Página de propriedade svn:mime-type .....	102
4.48. Página de propriedade svn:needs-lock .....	102
4.49. Página de propriedade svn:executable .....	102
4.50. Caixa de propriedades integra templates de mensagens de registo .....	103
4.51. A Caixa de Diálogo de Ramificar/Etiquetar .....	108
4.52. A Caixa de Diálogo Trocar .....	111
4.53. O Assistente de Integração - Selecciona o Intervalo de Revisões .....	113
4.54. O Assistente de Integração - Integração de Árvores .....	115
4.55. The Merge Conflict Dialog .....	118
4.56. The Merge Tree Conflict Dialog .....	119
4.57. The Merge-All Dialog .....	119
4.58. A Caixa de Diálogo Bloquear .....	121
4.59. A Caixa de Diálogo Verificar Alterações .....	122
4.60. A caixa de diálogo de Criar Correção .....	124
4.61. A Caixa de Diálogo de Anotar/Responsabilizar .....	126
4.62. TortoiseBlame .....	127
4.63. O Navegador de Repositório .....	129
4.64. O Grafo de Revisões .....	131
4.65. A Caixa de Diálogo Exportar-do-URL .....	136
4.66. A Caixa de Diálogo Reposicionar .....	137
4.67. Caixa de diálogo de Propriedades do Bugtraq .....	139
4.68. Caixa de diálogo de exemplo da consulta ao gestor de problemas .....	143
4.69. A Caixa de Diálogo Preferências, Página Geral .....	145
4.70. A Caixa de Diálogo Preferências, Página Menu de Contexto .....	147
4.71. A Caixa de Diálogo Preferências, Página Diálogos 1 .....	148
4.72. A Caixa de Diálogo Preferências, Página Diálogos 2 .....	150
4.73. Caixa de diálogo de preferências, página de diálogos 3 .....	151
4.74. A Caixa de Diálogo Preferências, Página de Cores .....	152
4.75. A Caixa de Diálogo Preferências, Página Gráfico de Revisões .....	153
4.76. A Caixa de Diálogo preferências, Página Cores do Gráfico de Revisões .....	154
4.77. A Caixa de Diálogo Preferências, Página Sobreposição de Ícones .....	156
4.78. A Caixa de Diálogo preferências, Página Conjunto de Ícones .....	159
4.79. A Caixa de Diálogo preferências, Página de Manípulos de Ícones .....	160
4.80. A Caixa de Diálogo Preferências, Página de Rede .....	161
4.81. A Caixa de Diálogo Preferências, Página Visualizador de Comparação .....	162
4.82. A Caixa de Diálogo Preferências, Caixa de Diálogo Comparar/Integrar Avançados .....	166
4.83. A Caixa de Diálogo, Página de Dados Guardados .....	167
4.84. A Caixa de Diálogo Preferências, Página Cache de Registo .....	168
4.85. A Caixa de Diálogo Preferências, Estatísticas da Cache de Registo .....	170
4.86. A Caixa de Diálogo Preferências, Página Scripts de Gancho .....	171
4.87. A Caixa de Diálogo Preferências, Configurar Scripts de Gancho .....	172
4.88. A Caixa de Diálogo Preferências, Página Integração com Controlador de Problemas .....	175
4.89. A Caixa de Diálogo Preferências, Página do TortoiseBlame .....	176
4.90. The Settings Dialog, TortoiseUDiff Page .....	177
4.91. The Settings Dialog, Sync Page .....	178
4.92. Barra de tarefas com agrupamento padrão .....	180
4.93. Barra de tarefas com agrupamento de repositório .....	180
4.94. Barra de tarefas com agrupamento de repositório .....	181
4.95. Agrupamento da barra de tarefas com sobreposições de cores do repositório .....	181
5.1. The edit project dialog of the project monitor .....	185
5.2. The main dialog of the project monitor .....	186
C.1. A caixa de diálogo submeter, mostrando a notificação de actualização .....	207

---

## Lista de Tabelas

2.1. URLs de Acesso ao Repositório .....	14
4.1. Pinned Revision .....	109
6.1. Lista de opções de linha de comando disponíveis .....	189
6.2. List of SubWCRev error codes .....	189
6.3. Lista de palavras-chave disponíveis .....	190
6.4. Métodos COM/automação suportados .....	193
C.1. Entradas de menu e seus valores .....	208
D.1. Lista de comandos e opções disponíveis .....	212
D.2. Lista de opções disponíveis .....	218
D.3. Lista de opções disponíveis .....	219

---

# Prefácio



# TortoiseSVN

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

## 1. O que é o TortoiseSVN?

O tortoiseSVN é um cliente de Windows gratuito e de código aberto para o sistema de controlo de versões *Apache™ Subversion®*. Isto é, o TortoiseSVN gere as pastas e os ficheiros ao longo do tempo. Os ficheiros são armazenados num *repositório* central. O repositório é muito parecido com um vulgar servidor de ficheiros à exceção de que se lembra de toda e qualquer alteração que fez nos seus ficheiros e pastas. Isto permite que recupere versões antigas dos seus ficheiros, e examine o histórico de como e quando os seus dados foram alterados, e quem os alterou. Esta é razão porque muita gente pensa que o Subversion e os sistemas de controlo de versões em geral, como um espécie de “máquina do tempo”.

Alguns sistemas de controlo de versões são também sistemas de programa de gestão de configurações (software configuration management - SCM). Estes sistemas são especificamente desenhados para gerir árvores 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* coleção de ficheiros, incluindo os de código-fonte.

## 2. Funcionalidades do TortoiseSVN

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

### Integração na "shell"

O TortoiseSVN integra-se sem problemas com a shell do Windows (i.e. o explorador). Isto significa que pode continuar a trabalhar com as ferramentas com que está familiarizado, e não terá de mudar de programa de cada vez que necessita das funcionalidades do controlo de versões!

E não está limitado a utilizar o Explorador do Windows. Os menus de contexto do TortoiseSVN funcionam em muitos outros gestores de ficheiros, e também na caixa de diálogo de Ficheiro/Abrir, que é comum à maioria das aplicações de referência do Windows. Deverá no entanto ter em conta que, o TortoiseSVN foi intencionalmente desenvolvido como uma extensão para o Explorador do Windows. Pelo que é possível que noutras aplicações esta integração não seja tão completa, como por exemplo os ícones sobrepostos poderão não ser mostrados.

### Sobreposição 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 sua cópia de trabalho.

### Interface gráfica de utilizador

Quando lista as alterações a um ficheiro ou pasta, pode clicar numa determinada revisão para visualizar os comentários dessa submissão. Pode também ver a lista de ficheiros alterados - faça apenas duplo clique num ficheiro para ver exatamente o que foi alterado.

A caixa de diálogo de submeter, lista todos os itens que irão ser incluídos numa submissão, cada item tem uma caixa de verificação, para que possa escolher quais os itens que quer incluir. Os ficheiros não versionados também podem ser listados, para o caso de se esquecer de adicionar aquele novo ficheiro.

#### Acesso 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. Os ficheiros e as pastas são versionadas. Como resultado, existem comandos reais do lado do cliente de, **mover** e **copiar** que atuam em ficheiros e pastas.

#### Submissões atômicas

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

#### Metadados versionados

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

#### Escolha de camadas de rede

O Subversion tem uma noção abstrata 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 fornecido. Este servidor fala um protocolo personalizado 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) como para 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 direções através da rede.

#### Ramificação e etiquetação eficiente

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

## 3. Licença

O TortoiseSVN é um projeto de código-fonte aberto (open source) desenvolvido sob a GNU General Public License (GPL). Pode ser descarregado gratuitamente e utilizado também gratuitamente, tanto para uma utilização pessoal como comercial, em qualquer número de computadores.

Embora a maioria das pessoas apenas descarregam o instalador, se quiser ainda tem acesso ao código-fonte deste programa. Pode explorá-lo nesta hiperligação: <https://sourceforge.net/p/tortoisesvn/code/HEAD/tree/>. A versão em desenvolvimento atual está localizada em `/trunk/`, e as versões já lançadas estão em `/tags/`.

## 4. Desenvolvimento

Tanto o TortoiseSVN como o Subversion são desenvolvidos por uma comunidade de pessoas que estão a trabalhar nesses projetos. Elas são provenientes de diferentes países por todo o mundo, trabalhando juntos para criar bom programa.

## 4.1. A história do TortoiseSVN

Em 2002, o Tim Kemp descobriu que o Subversion era um sistema de controlo de versões muito bom, mas faltava-lhe um bom cliente com interface gráfica (GUI). A ideia de um cliente para o Subversion com integração na shell do Windows foi inspirada pelo seu cliente similar para o CVS, o TortoiseCVS. O Tim estudou o código-fonte do TortoiseCVS e usou-o como base para o TortoiseSVN. Então ele arrancou com o projeto registando o domínio `tortoisesvn.org` e colocando o código-fonte disponível na Internet.

Por volta dessa altura o Stefan Küng andava à procura de um bom e gratuito sistema de controlo de versões, tendo encontrado o Subversion e o código-fonte para o TortoiseSVN. Visto que o TortoiseSVN ainda não estava pronto a usar, ele juntou-se ao projeto e começou a programar. Depressa refez a maior parte do código existente e começou a adicionar comandos e funcionalidades, até ao ponto em que já nada restava do código original.

À medida que o Subversion se tornou mais estável, atraiu mais e mais utilizadores que começaram a utilizar o TortoiseSVN como o seu cliente do Subversion. A base de utilizadores cresceu rapidamente (e continua a crescer a cada dia). Foi então quando Lübbe Onken ofereceu-se para ajudar com uns ícones atraentes, e um logótipo para o TortoiseSVN. Agora ele gere o sítio de Internet e também as traduções.

Com o tempo, todos os outros sistemas de controlo de versões obtiveram os seus clientes Tortoise o que causou um problema com a sobreposição dos ícones no Explorador : o número das sobreposições é limitado e mesmo um cliente Tortoise pode facilmente exceder esse limite. Foi então quando Stefan Küng implementou o componente TortoiseOverlays que permite que todos os clientes Tortoise usem a mesma sobreposição de ícones. Agora todos os clientes de código-fonte aberto Tortoise e mesmo alguns cliente não Tortoise usam esse componente partilhado.

## 4.2. Agradecimentos

Tim Kemp  
por ter fundado o projeto TortoiseSVN

Stefan Küng  
pelo trabalho árduo realizado para fazer o que TortoiseSVN é hoje, e a sua liderança do projeto

Lübbe Onken  
pelos ícones maravilhosos, o logotipo, as caças aos erros, as traduções e gestão das traduções

Simon Large  
por manter a documentação

Stefan Fuhrmann  
pela cache de registo e grafo de revisões

O livro sobre o Subversion  
pela excelente introdução a Subversion e ao capítulo 2 que copiamos para aqui

O projeto de estilo da Tigris  
por alguns dos estilos reutilizados nesta documentação

Os nossos colaboradores  
para correções, relatórios de erros e novas ideias, e também para ajudar os outros ao responderem às questões da nossa lista de correio.

Os nossos doadores  
por muitas horas de prazer com a música que nos enviaram

## 5. Guia de leitura

Este livro foi escrito para pessoal com conhecimentos de computação que querem utilizar o Subversion para gerir os seus dados, mas que preferem o uso de um GUI em vez do cliente de linha de comandos. Visto que TortoiseSVN é uma extensão de "shell", é assumido que o utilizador está familiarizado com o Explorador do Windows e como o utilizar.

Este **Prefácio** explica o que é o TortoiseSVN, fala um pouco sobre o TortoiseSVN e a comunidade de pessoas que trabalham nele e também nas condições de licenciamento para usá-lo e distribuí-lo.

O **Capítulo 1, Começar** explica como instalar o TortoiseSVN no seu computador e como começar a usá-lo de imediato.

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 projeto do Subversion, e explica as diferentes abordagens ao controlo de versões e como funciona o Subversion.

O capítulo em **Capítulo 3, O repositório** como instalar e configurar um repositório, que é útil para testar o Subversion e o TortoiseSVN utilizando apenas um computador. Também explica um pouco sobre administração de repositórios que também é relevante para repositórios localizados num servidor.

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 por verificar uma cópia de trabalho, alterá-la, submeter as tuas alterações, etc. Então avança-se para tópicos mais avançados.

O **Capítulo 5, Project Monitor** explica como pode monitorizar os seus projetos Subversion para que não perca submissões importantes de outros membros da sua equipa.

O **Capítulo 6, O Programa SubWCRev** é um programa separado e incluído no TortoiseSVN, que pode extrair informação a partir da sua cópia de trabalho e gravá-la num ficheiro. Isto é útil para incluir informação de construção (build) nos seus projetos.

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 pode necessitar de interação 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`.

## 6. Terminologia utilizada neste documento

Para tornar mais fácil a leitura dos documentos, os nomes de todos os ecrãs e menus do TortoiseSVN estão escritos com um tipo de letra diferente. Por exemplo a Caixa de diálogo de registos.

O menu escolhido é indicado com uma seta. TortoiseSVN → **Mostrar registo** significa: seleccione *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 da interface de utilizador são indicados da seguinte forma: Premir **OK** para continuar.

As ações do utilizador são sinalizadas com a fonte a negrito. **Alt+A**: premir a tecla **Alt** no seu teclado e enquanto a mantém premida, prima também a tecla **A**. Arrastar para a direita: premir o botão direito do rato e enquanto o prime *arraste* 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 lhe facilitam a vida.



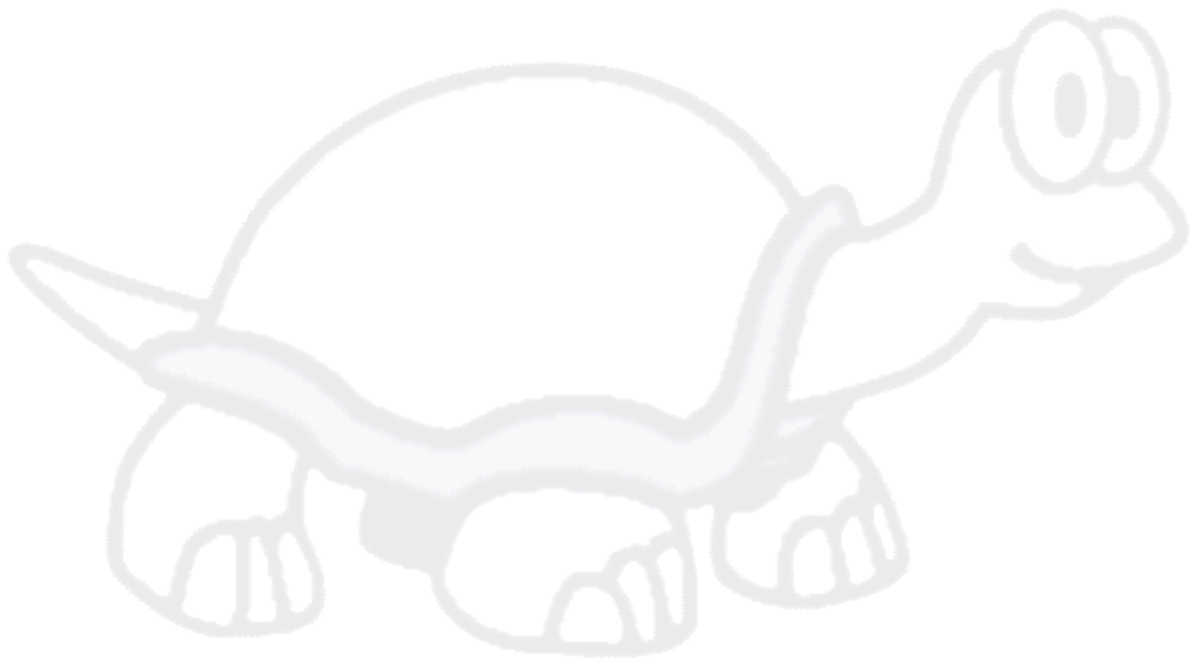
## **Cuidado**

Lugares onde é preciso cuidado com o que faz.



## **Atenção**

Onde tem de ter cuidado extremo. Pode ocorrer corrupção de dados ou outras coisas desagradáveis se esses avisos forem ignorados.



---

# Capítulo 1. Começar

Esta secção é destinada às pessoas que gostariam de descobrir o que é realmente o TortoiseSVN e dar uma volta de teste. Ela explica como instalar o TortoiseSVN e instalar um repositório local, e caminha também através das operações mais usadas.

## 1.1. Instalar o TortoiseSVN

### 1.1.1. Requisitos de sistema

O TortoiseSVN corre em Windows Vista ou mais recente e está disponível para 32-bit e 64-bit. O instalador para Windows 64-bit também inclui as partes da extensão de 32-bit. O que significa que não tem de instalar a versão 32-bits separadamente para obter o menu de contexto do TortoiseSVN e sobreposições em programas 32-bit.

O suporte para Windows 98, Windows ME e Windows NT4 foi descontinuado na versão 1.2.0, e o suporte ao Windows 2000 e XP até ao SP2 foi descontinuado na versão 1.7.0. O suporte ao Windows XP com SP3 foi descontinuado na versão 1.9.0. Ainda pode descarregar e instalar versões antigas se precisar.

### 1.1.2. Instalação

O TortoiseSVN vem com um instalador fácil de usar. Faça duplo clique no ficheiro de instalação e siga as instruções. O instalador deverá tomar conta do resto do processo de instalação. Não se esqueça de reiniciar após a instalação.



#### Importante

Necessita de privilégios de administrador para instalar o TortoiseSVN. O instalador pergunta pelas suas credenciais de administrador se for necessário.

Estão disponíveis pacotes de linguagens que traduzem o interface de utilizador do TortoiseSVN em muitas línguas diferentes. Consulte [Apêndice G, Pacotes de Linguagens e Verificadores de Sintaxe](#) para mais informações em como instalá-las.

Se tiver problemas durante ou após a instalação do TortoiseSVN por favor veja a nossa página de perguntas frequentes disponível em <https://tortoisesvn.net/faq.html>.

## 1.2. Conceitos básicos

Antes de ficarmos atarefados com alguns ficheiros reais, é importante ter uma visão geral de como o Subversion funciona e dos termos que são usados.

### O repositório

O Subversion usa uma base de dados central que contém todos os seus ficheiros com controlo de versões, com o seu histórico completo. Esta base de dados é referida como o *repositório*. O repositório reside normalmente num servidor de ficheiros correndo o programa de servidor Subversion, que fornece a pedido, conteúdos para os clientes Subversion (como o TortoiseSVN). Se só efetuar cópias de segurança de uma coisa, efetue uma cópia de segurança do seu repositório já que é a cópia-mestre definitiva de todos os seus dados.

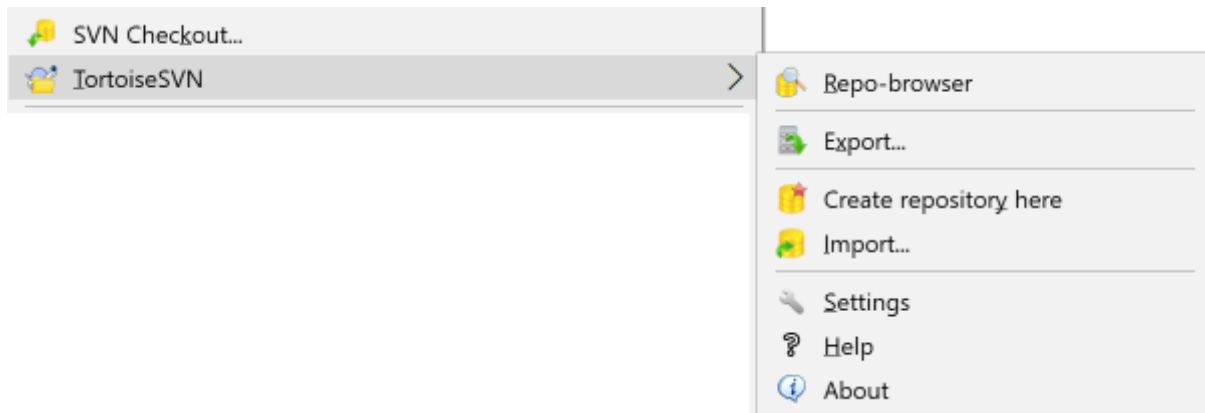
### Cópia de trabalho

É aqui que fará realmente o trabalho. Cada programador terá a sua própria cópia de trabalho no seu PC local, por vezes conhecida como a caixa de areia. Poderá obter a versão mais recente do repositório, trabalhar nela localmente sem afetar ninguém, e então quando se sentir satisfeito com as alterações que fez, submetê-las de volta para o repositório.

A cópia de trabalho do Subversion não contém o histórico do projeto, mas mantém uma cópia dos ficheiros como eles existem no repositório antes de começar a fazer alterações. Isto significa que é fácil verificar exatamente que alterações fez.



Também precisa de saber como encontrar o TortoiseSVN, porque não há muito para ver a partir do menu Iniciar. Isto é porque o TortoiseSVN é uma extensão da shell, pelo que em primeiro lugar, inicie o Explorador do Windows. Clique com o botão direito no explorador e deverá ver algumas novas entradas no menu de contexto, tal como:



**Figura 1.1. O menu TortoiseSVN para pastas não versionadas**

## 1.3. Ir para um teste de condução

Esta secção mostra-lhe como experimentar algumas das funcionalidades mais usadas num pequeno repositório de teste. Naturalmente não explicará tudo - afinal, isto é apenas um Guia de Início Rápido. Após estar familiarizado deverá tirar algum tempo para ler o resto deste guia de utilização, que o levará através das coisas com muito mais detalhe. Também explicará mais sobre como instalar corretamente um servidor Subversion.

### 1.3.1. Criar um repositório

Para um projeto real, terá um repositório instalado algures num local seguro e com um servidor Subversion a controlá-lo. Para os objetivos deste tutorial iremos utilizar a funcionalidade do repositório Subversion local, que te permitirá acesso direto a um repositório criado no seu disco duro, sem nenhuma necessidade de servidor.

Primeiro crie uma nova pasta vazia no seu PC. Poderá ir para qualquer lado, mas neste tutorial vamos dar-lhe o nome C:\svn\_repos. Agora clique com o botão direito do rato na nova pasta e a partir do menu de contexto escolha TortoiseSVN → Criar repositório aqui.... É então criado o repositório dentro da pasta, pronto a usar. Iremos também criar a estrutura interna de pastas por defeito, clicando no botão Criar estrutura de pastas.

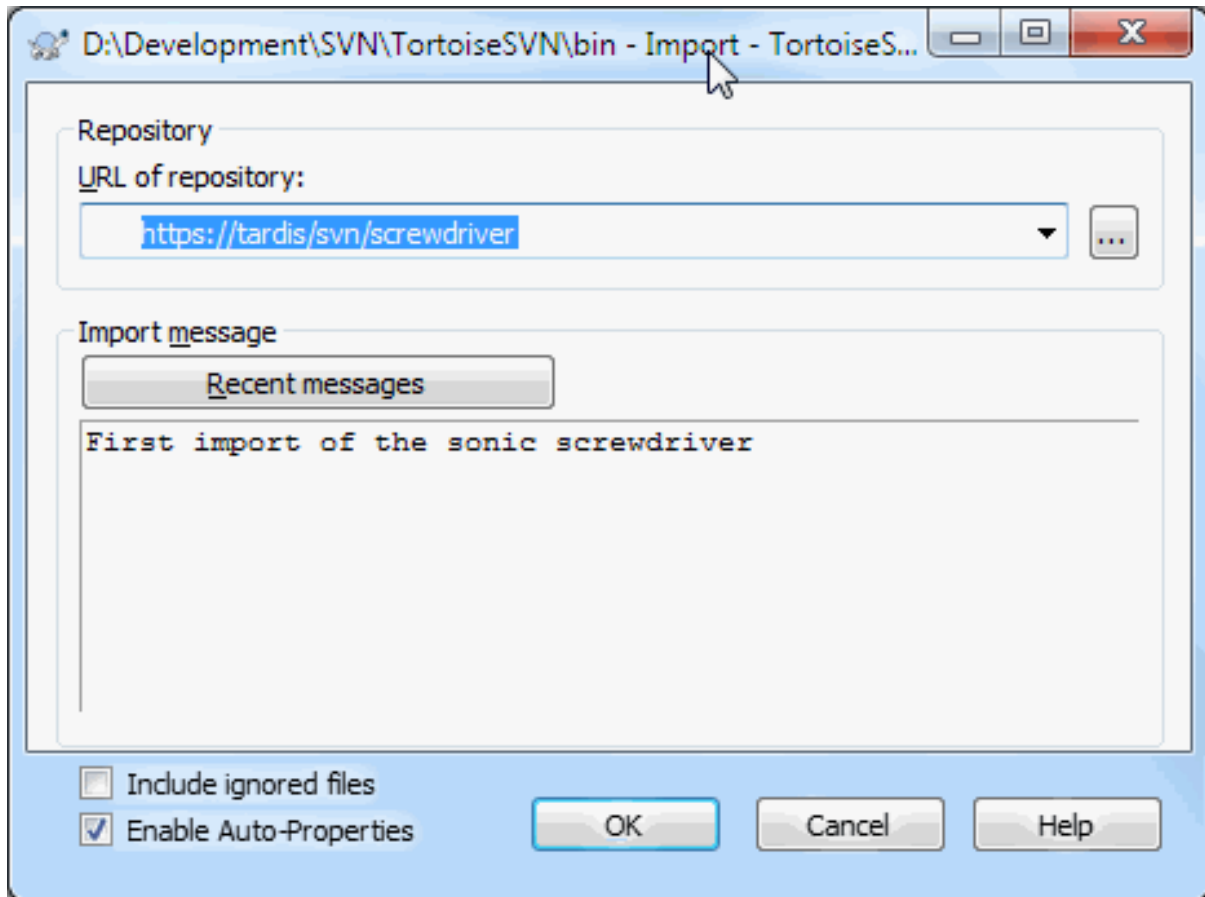


#### Importante

A funcionalidade do repositório local é muito útil apenas para testes e avaliação, a não ser que esteja a trabalhar sozinho num único PC, deverá usar sempre um servidor Subversion corretamente configurado. É tentador numa empresa pequena evitar o trabalho de instalar um servidor, apenas para aceder ao seu repositório situado numa partilha de rede. Nunca faça isso. Irá perder dados. Leia [Secção 3.1.4, “Aceder ao Repositório numa Partilha de Rede”](#) para descobrir porque isso é uma má ideia e saber como instalar um servidor.

### 1.3.2. Importar um projeto

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



**Figura 1.2. Caixa de diálogo Importar**

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

A outra funcionalidade importante desta caixa de diálogo é a caixa **Importar mensagem**, que lhe permite introduzir uma mensagem descrevendo o que está a fazer. Quando voltar a olhar para a história do projeto, essas mensagens de submissão serão um guia valioso para as alterações que foram feitas e o porquê. Neste caso podemos dizer algo como “Importar o projeto `Widget1`”. Clique em **OK** e a pasta é adicionada ao seu repositório.

### 1.3.3. Verificar uma cópia de trabalho

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



#### **Importante**

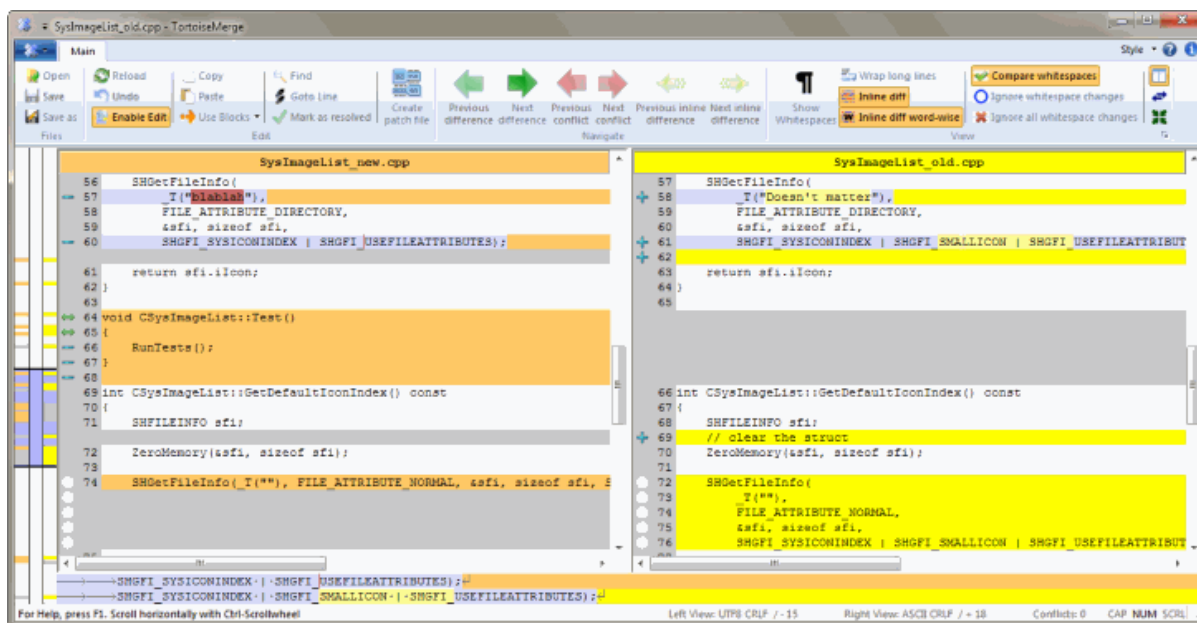
In the default setting, the checkout menu item is not located in the TortoiseSVN submenu but is shown at the top explorer menu. TortoiseSVN commands that are not in the submenu have **SVN** prepended: **SVN Checkout...**

Irá notar que a aparência desta pasta é diferente da nossa original. Cada ficheiro tem uma marca de verificação verde no quanto inferior esquerdo. Isso são ícones de estado do TortoiseSVN que apenas estão presentes numa cópia de trabalho. O estado verde indica que o ficheiro está inalterado em relação à versão do repositório.

### 1.3.4. Fazer alterações

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

Mas quais são as alterações? Clique com o botão direito num dos ficheiros alterados e selecione TortoiseSVN → Comparar. A ferramenta de comparação de ficheiros do TortoiseSVN arranca mostrando exatamente quais as linhas que foram alteradas.



**Figura 1.3. Visualizador de diferenças em ficheiros**

OK, agora estamos satisfeitos com as alterações e vamos atualizar o repositório. Esta cação é conhecida como a Submissão das alterações. Clique com o botão direito na pasta Widget1-Dev e selecione TortoiseSVN → Submeter. A caixa de diálogo submeter irá listar os ficheiros alterados, cada um com a sua caixa de verificação. Poderá querer escolher um sub-grupo desses ficheiros, mas neste caso iremos submeter as alterações de ambos os ficheiros. Introduza uma mensagem, descrevendo a razão da alteração e clique no botão OK. A caixa de diálogo de progresso mostrará os ficheiros a serem carregados para o repositório, e está terminado.

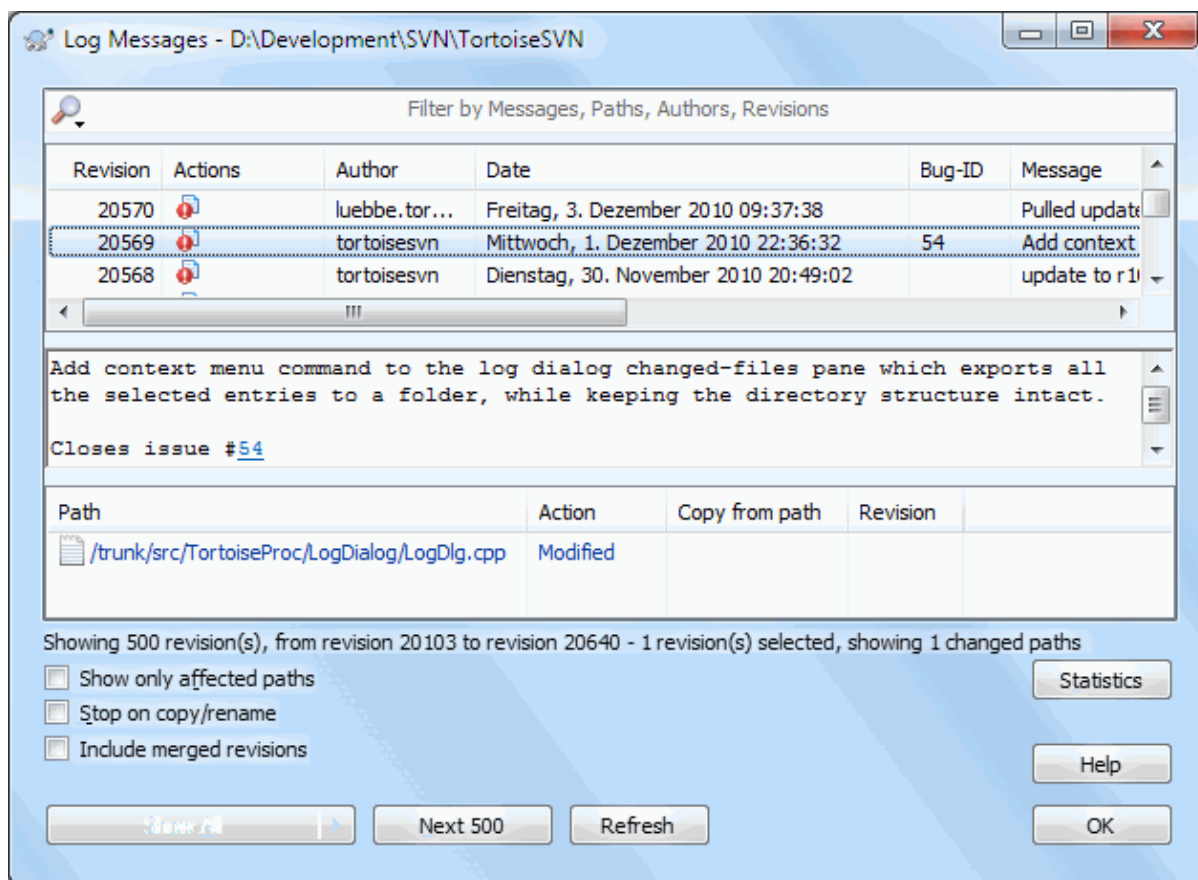
### 1.3.5. Adicionar mais ficheiros

À medida que o projeto se desenvolve irá necessitar de adicionar novos ficheiros - digamos que adiciona novas funcionalidades em Extras.c e adiciona uma referência no Makefile, já existente. Clique com o botão direito na pasta e TortoiseSVN → Adicionar. A caixa de diálogo Adicionar mostra agora todos os ficheiros não versionados e pode escolher quais quer adicionar. Outra maneira de adicionar ficheiros é clicar no próprio ficheiro e selecionar TortoiseSVN → Adicionar.

Agora quando for submeter a pasta, o novo ficheiros aparecerá como *Adicionado* e o ficheiro existente como *Alterado*. Observe que poderá fazer um clique duplo no ficheiro alterado para verificar quais foram exatamente as alterações feitas.

### 1.3.6. Ver a história do projeto

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



**Figura 1.4. A caixa de diálogo Registo**

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

O painel de topo mostra a lista de revisões submetidas juntamente com o início da mensagem de submissão. Se seleccionar uma das revisões, o painel do meio irá mostra a mensagem de registo completa para essa revisão, e o painel do fundo mostrará a lista de ficheiros e pastas alterados.

Cada um desses painéis tem um menu de contexto que lhe fornece muitas mais maneiras de usar a informação. No painel do fundo poderá fazer um clique duplo num ficheiro para ver exatamente que alterações foram feitas nessa revisão. Consulte [Secção 4.10, “Caixa de Diálogo Registo de Revisões”](#) para obter toda a informação.

### 1.3.7. Desfazer alterações

Uma funcionalidade comum a todos os sistemas de controlo de versões, é deixarem-lhe desfazer as alterações que fez anteriormente. Como poderia esperar, o TortoiseSVN faz com que seja de fácil acesso.

Se quiser livrar-se das alterações que ainda não submeteu e voltar a ter o seu ficheiro como estava, antes de o começar a editar, o TortoiseSVN → Reverter será o seu melhor amigo. Este descarta as suas alterações (para o lixo, por precaução) e reverte para a revisão submetida com que iniciou a edição.

Se quiere desfazer os efeitos de uma revisão em particular, inicie a caixa de diálogo de registo e encontre a revisão comprometida. Selecione Menu de contexto → Reverter as alterações desta revisão e essas alterações serão desfeitas.

## 1.4. Avançar...

Este guia deu-lhe uma volta muito rápida por algumas das mais úteis e importantes funcionalidades do TortoiseSVN, mas está claro que existe muito mais para além do que abordámos. Recomendamos vivamente que tire algum tempo para ler o resto deste manual, em especial o [Capítulo 4, Guia de Uso Diário](#), que lhe dará mais detalhe nas operações do dia-a-dia.

Tivemos muito trabalho em assegurar que este documento é ao mesmo tempo informativo e fácil de ler, mas reconhecemos que existe muito de ambos! Leve o seu tempo e não tenha medo de experimentar as coisas num repositório de teste, enquanto progride. A melhor maneira de aprender é usando-o.

---

# Capítulo 2. Conceitos básicos de controlo de versões

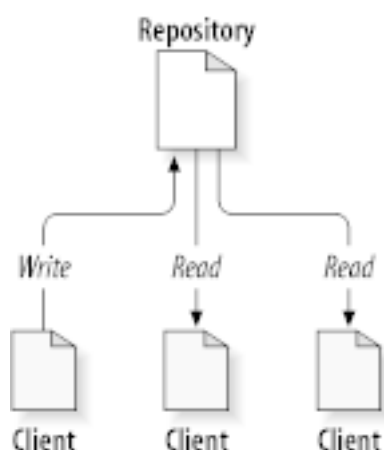
Este capítulo é uma versão ligeiramente modificada do mesmo capítulo no livro do Subversion. Uma versão em linha do livro do Subversion pode ser encontrada aqui: <http://svnbook.red-bean.com/>.

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.

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

## 2.2. 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!

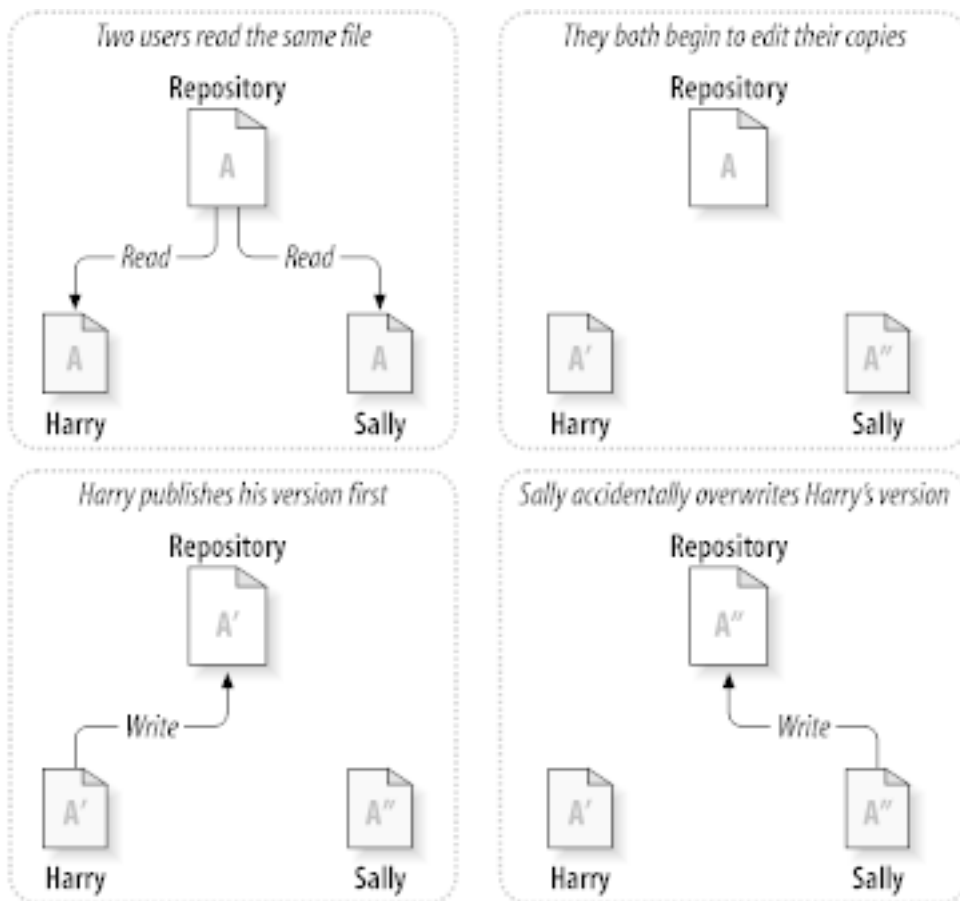
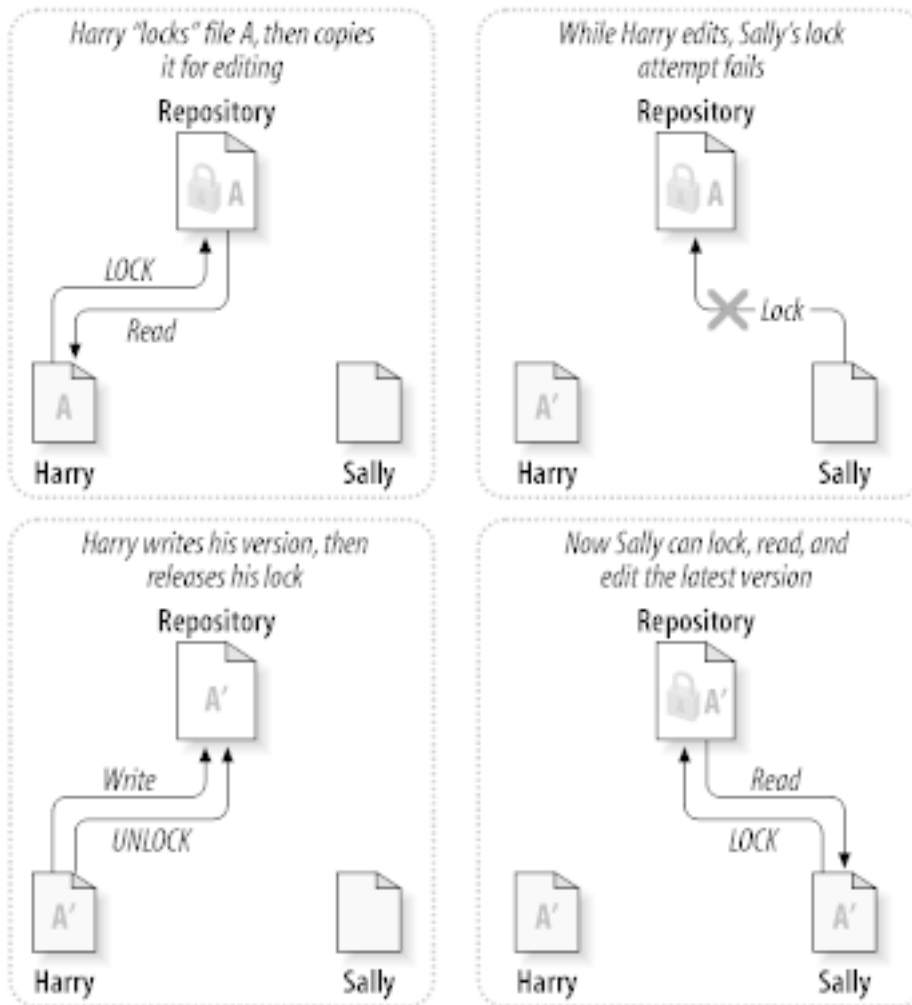


Figura 2.2. O Problema a Evitar

### 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 sobreporiam 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

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



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

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.

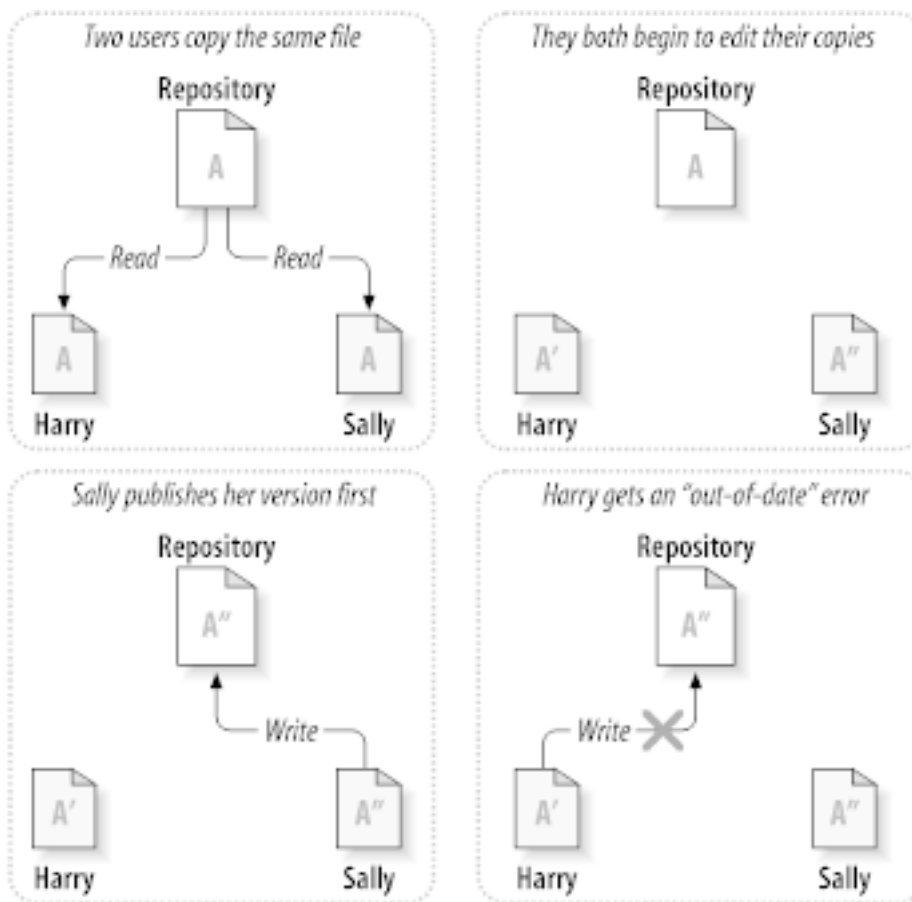
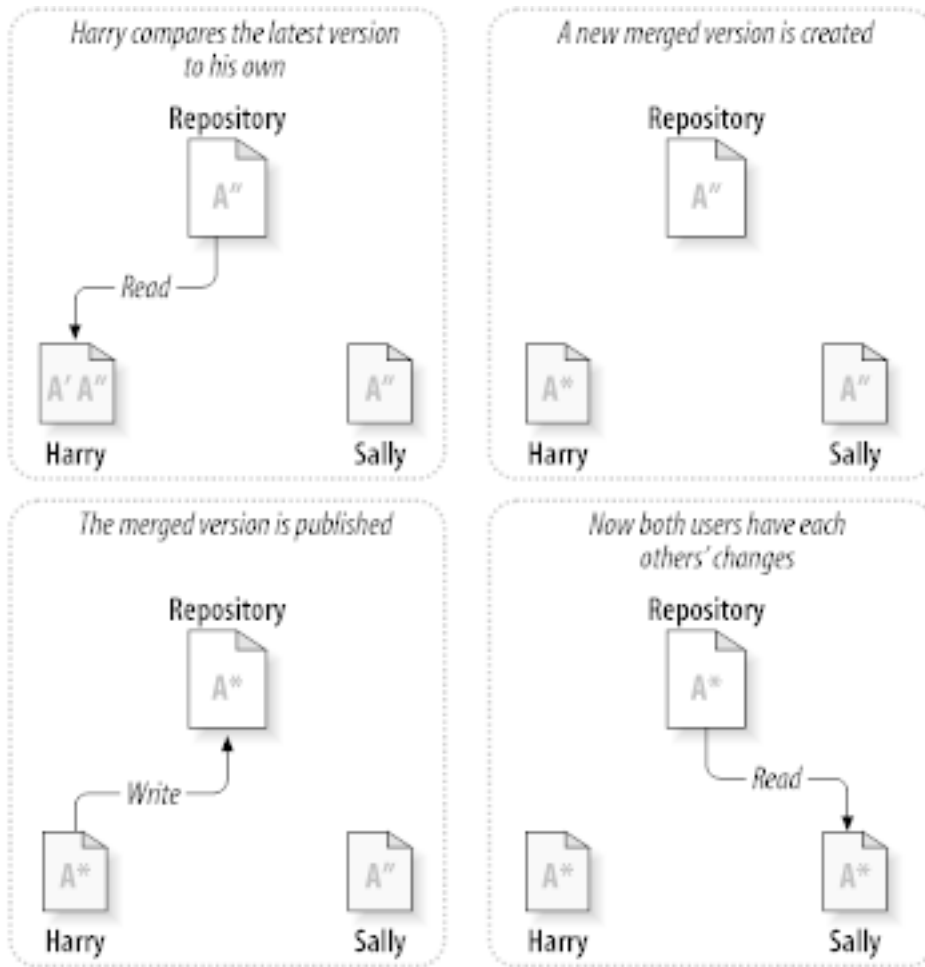


Figura 2.4. A solução Copiar-Modificar-Integrar



**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 pessoas 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.

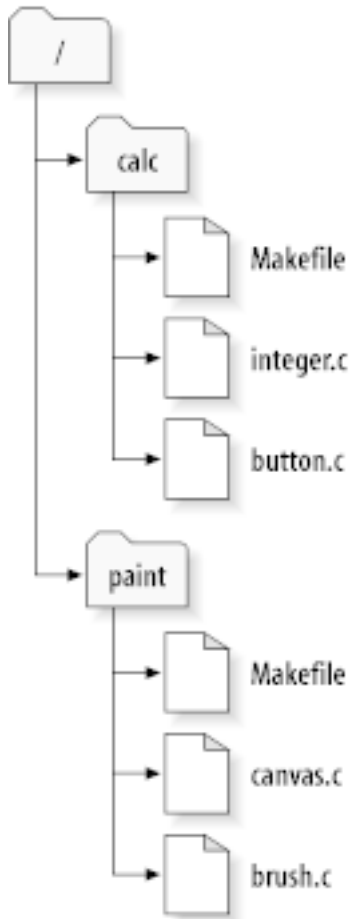
Uma cópia de trabalho do Subversion é uma normal árvore de pastas no teu sistema local contendo uma colecção de ficheiros. Tu podes editar esses ficheiros como desejares, e no caso de serem ficheiros de código fonte, poderás compilar o teu programa a partir deles da forma habitual. A tua cópia de trabalho 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 working copy also contains some extra files, created and maintained by Subversion, to help it carry out these commands. In particular, your working copy contains a subdirectory named `.svn`, also known as the working copy *administrative directory*. The files in this administrative directory help Subversion recognize which files contain unpublished changes, and which files are out-of-date with respect to others' work. Prior to 1.7 Subversion maintained `.svn` administrative subdirectories in every versioned directory of your working copy. Subversion 1.7 takes a completely different approach and each working copy now has only one administrative subdirectory which is an immediate child of the root of that working copy.

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`.

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

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

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

**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.

You can access a FSFS repository via a network share, but this is *not* recommended for various reasons:

- You are giving direct write access to all users, so they could accidentally delete or corrupt the repository file system.
- Not all network file sharing protocols support the locking that Subversion requires. One day you will find your repository has been subtly *corrupted*.
- You have to set the access permissions in just the right way. SAMBA is particularly difficult in this respect.
- If one person installs a newer version of the client which upgrades the repository format, then everyone else will be unable to access the repository until they also upgrade to the new client version.

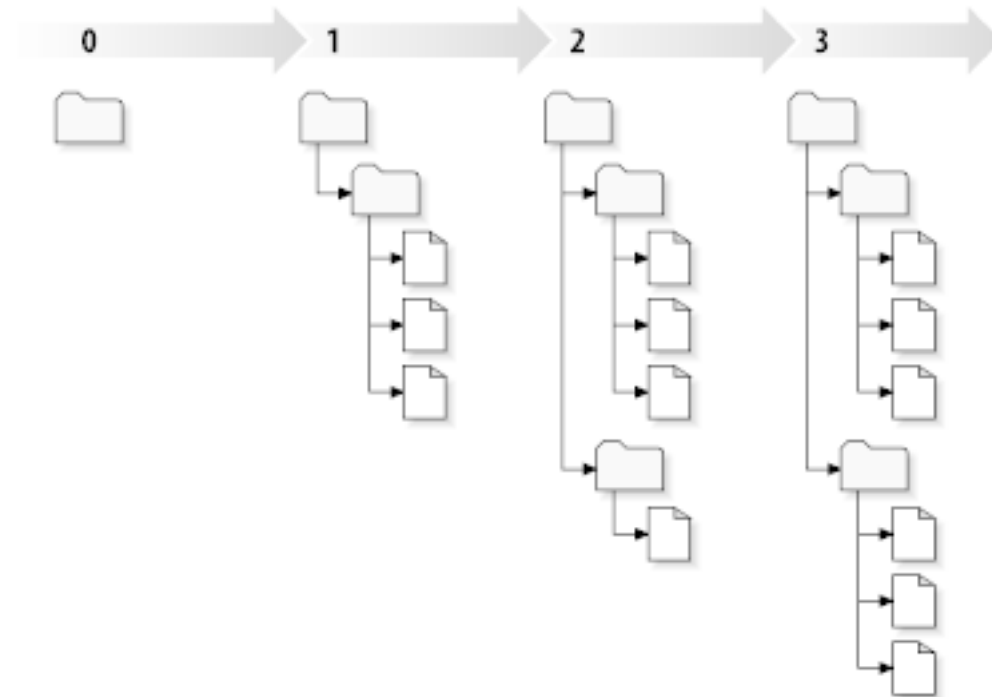
### 2.3.3. Revisões

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

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

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/`:

- what revision your working file is based on (this is called the file's *working revision*), and
- 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** o 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** o 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

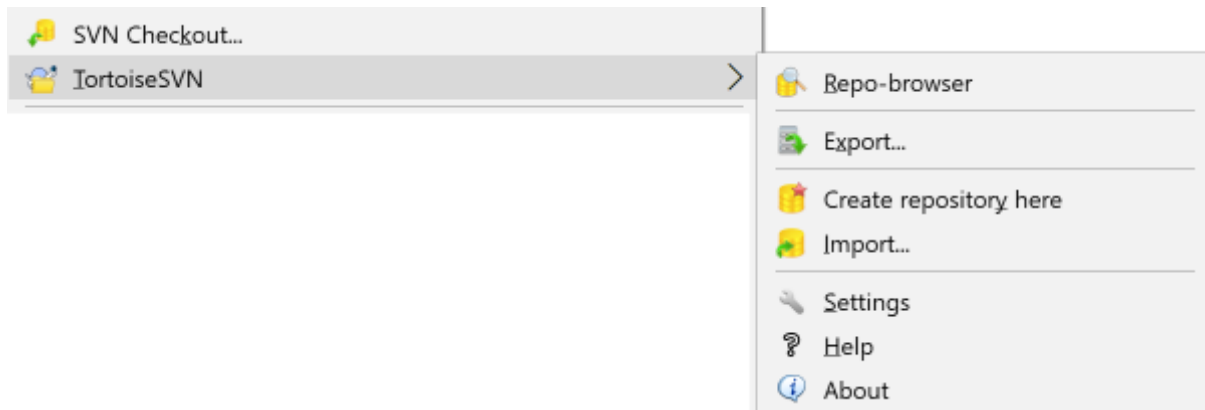
### 3.1.1. Criar um Repositório com o Cliente de Linha de Comandos

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. Open the command prompt (or DOS-Box), change into D:\SVN\ and type

```
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.

Serás também questionado se irás querer criar uma estrutura de pastas no repositório. Pesquisa mais sobre opções de formato em [Secção 3.1.5, “Estrutura do Repositório”](#).

O TortoiseSVN irá criar um ícone de pasta personalizado quando cria um repositório, para que possas identificar mais facilmente os repositórios locais. Se criares um repositório usando o cliente de linha de comandos oficial, este ícone de pasta não será atribuído.



## Dica

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

### 3.1.3. 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.

### 3.1.4. Aceder ao Repositório numa Partilha de Rede

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

- 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.
- If one person installs a newer version of the client which upgrades the repository format, then everyone else will be unable to access the repository until they also upgrade to the new client version.

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
/branches
/tags
```

Porque sendo tão normalmente usado este formato, ao criares um novo repositório através do TortoiseSVN este irá também se oferecer para criar por ti a estrutura de pastas.

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

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

...ou por projecto:

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

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 desejares, dentro dessa pasta - não ponhas lá ainda nenhum ficheiro!
3. Importa esta estrutura para o repositório com um clique com o botão direito na pasta que contém esta estrutura de pastas, e seleccionando TortoiseSVN → Importar... Na caixa de diálogo importar, introduz o URL do teu repositório e clica OK. Então a tua pasta temp será importada para a raiz do repositório, para criar a estrutura básica do mesmo.

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

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

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

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

## 3.2. 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.

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

The recommended method is to run

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

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

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

## 3.3. Scripts de gancho de servidor

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

Esses scripts de gancho são executados pelo servidor que hospeda o repositório. O TortoiseSVN também permite configurar scripts de gancho de cliente, que são executados localmente devido a certos eventos. Consultar [Secção 4.31.8, “Scripts de Gancho do Lado do Cliente”](#) para mais informações.

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

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

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



## Ganchos de Substituição

Se um script de gancho rejeitar a tua submissão então essa decisão é final. Mas tu podes criar no próprio script um mecanismo de substituição, usando a técnica *Magic Word*. Se o script pretender rejeitar a operação então primeiro, pesquisa na mensagem de registo à procura de uma palavra passe especial, seja ela uma frase fixa ou talvez um nome de ficheiro com um prefixo. Se encontrar a tal palavra mágica, então permite o progresso da submissão. Se a frase não é encontrada, então poderá bloquear a submissão com uma mensagem do tipo “Tu não disseste a palavra mágica”. :-)

## 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>
```

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

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

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` > `http://`

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

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

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

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

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

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

---

# 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.

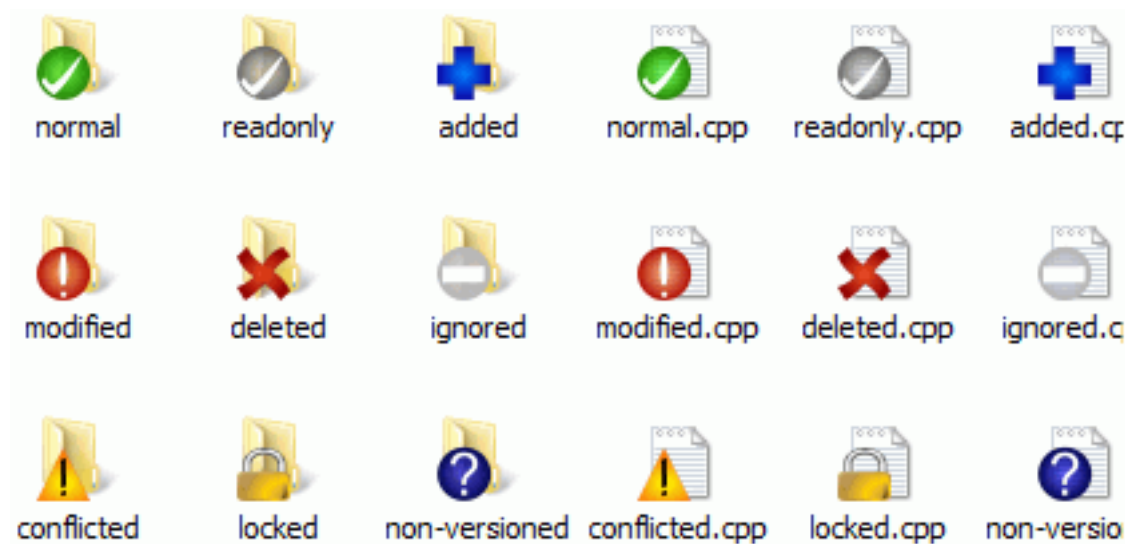
De modo a conseguir o máximo proveito do Guia de Uso Diário:

- 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. Funcionalidades gerais

Esta secção descreve algumas das funcionalidades do TortoiseSVN que se aplicam a praticamente tudo neste manual. Ter em atenção que muitas dessas funiconalidades apenas aparecerão no interior de uma cópia de trabalho Subversion.

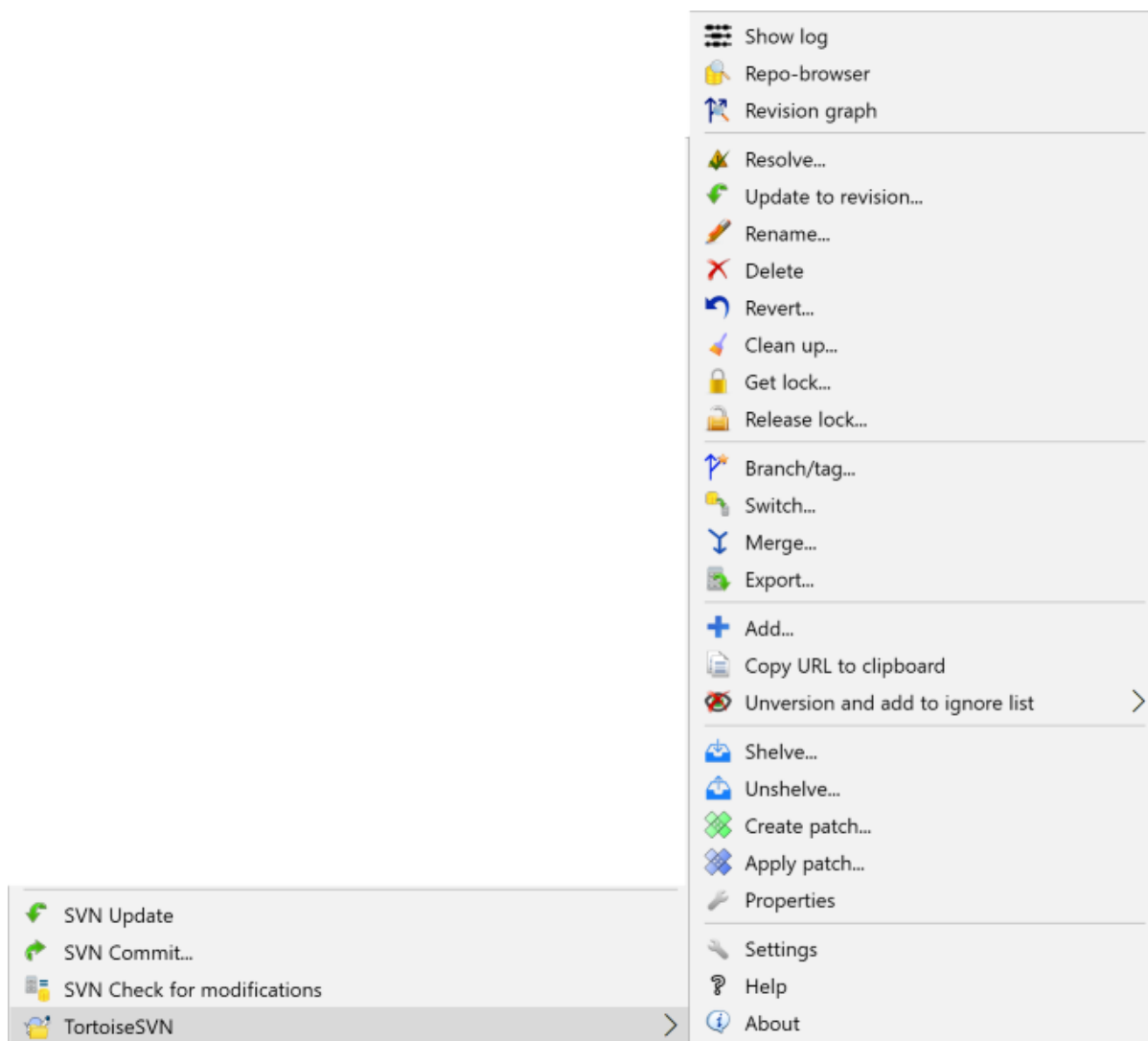
### 4.1.1. Sobreposição de Ícones



**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, apenas estão disponíveis no menu de contexto extendido. Para acederes a esse menu de contexto extendido, mantém premida a tecla **Shift** enquanto clicas à direita.

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

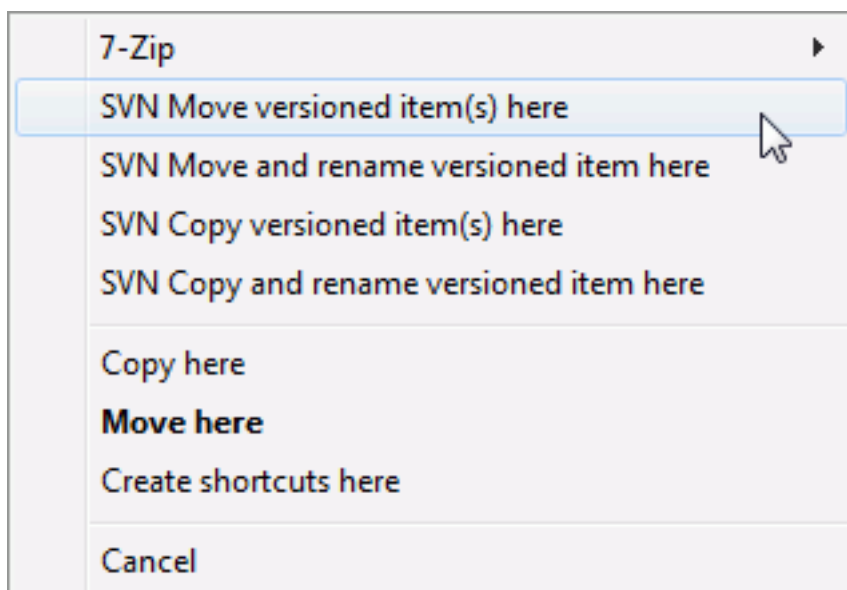




**Figura 4.3. Menu de ficheiro do Explorador para um atalho numa pasta versionada.**

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.

### 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

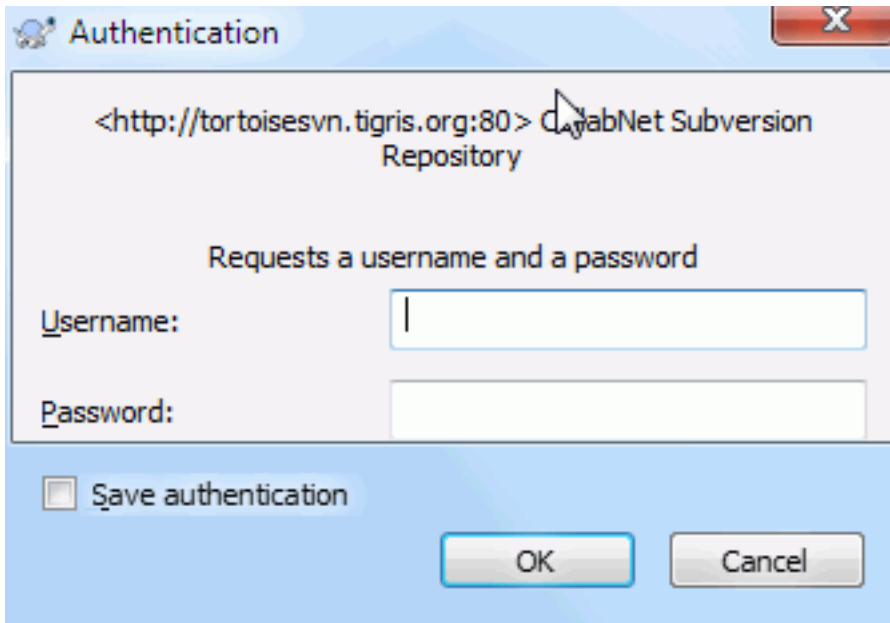
Seleccionar todos. Poderá ser utilizado em caso de receberes uma mensagem de erro quiseres copiar e colar para um email. Usa Ctrl-A para seleccionar a mensagem de erro e...

Ctrl-C

Copy the selected text. In case no text is selected but e.g. a list entry or a message box, then the content of that list entry or the message box is copied to the clipboard.

#### 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 a autenticação básica (nome de utilizador/palavra-passe). De notar que, as palavras-passe são guardadas usando a API WinCrypt e não em texto simples
- `svn.ssl.server` contém os certificados SSL de servidor.
- `svn.username` contém as credenciais para autenticação apenas por utilizador (sem necessidade de palavra-passe).

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

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

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

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

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

#### 4.1.6. Maximizar 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 ecran. 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. Importar Dados Para Um Repositório

### 4.2.1. Importar

Se estás a importar um repositório existente, que já contém alguns projectos então a sua estrutura já está definida. Se estás a importar dados para um novo repositório, então vale a pena tirar algum tempo para pensar no modo em como será organizado. Ler [Secção 3.1.5, “Estrutura do Repositório”](#) para aconselhamento posterior.

Esta secção descreve o comando importar do Subversion, que foi desenhado para importar uma hierarquia de pastas para o repositório de uma só vez. Apesar de efectuar o trabalho, tem algumas deficiências:

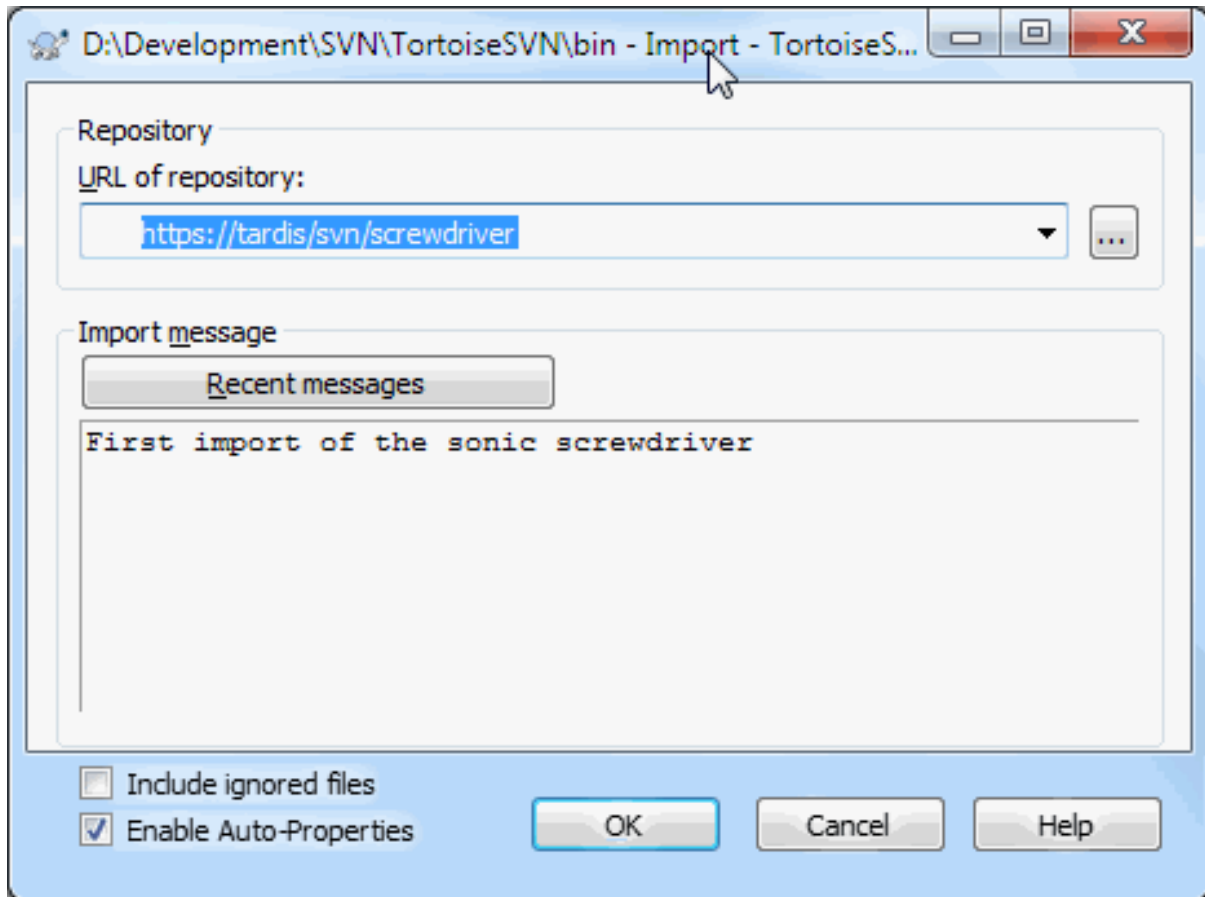
- Não existe modo de seleccionar ficheiros e pastas a incluir, tal como utilizar as configurações do padrão global para ignorar arquivos.
- A pasta importada não se torna uma cópia de trabalho. Terás de efectuar um SVN exportar para copiar os ficheiros de volta do servidor.
- É fácil importar a pasta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

Por essas razões recomendamos que não uses de modo algum o comando importar mas em alternativa segue o metodo de dois-passos descrito em [Secção 4.2.2, “Importar no local”](#), a não ser que estejas a executar o único passo de criar uma estrutura inicial `/trunk /tags /branches` no teu repositório. Uma vez que aqui estás, é assim como funciona o importar 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**

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

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

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

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 4.31.1, “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 repositório do navegador para criar uma nova pasta de projecto directamente no repositório. Se estiveres a usar um dos layouts padrão, provavelmente vais querer criar este como uma sub-pasta do trunk em vez de na raiz do repositório. O repositório do navegador mostra a estrutura do repositório assim como o explorador do Windows, assim podes ver como as coisas são organizadas.
2. Efectua checkout da nova pasta sobre a pasta que queres importar. Receberás um aviso que a pasta local não está vazia, ignora-o. Terás agora 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.

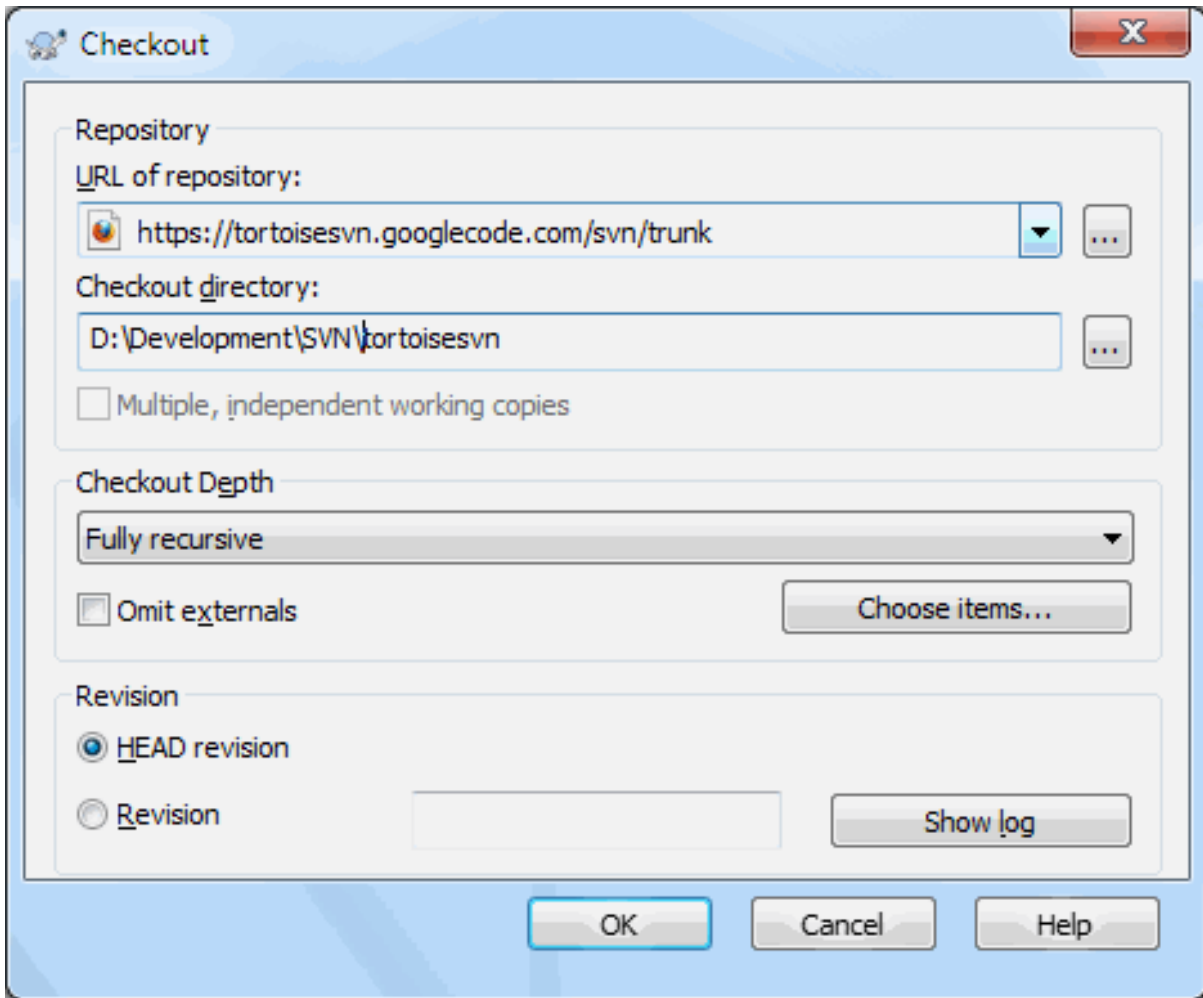
As an example, you can have a look at TortoiseSVN's build script. It calls a file named `default.build.user` which doesn't exist in the repository. Only the file `default.build.user.tpl`. `default.build.user.tpl` is the template file which every developer has to create a copy from and rename that file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

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

### 4.3. 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.



### Importante

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

#### 4.3.1. Profundidade do Checkout

Podes escolher *0 nível* que queres para o checkout, o que te permite especificar o nível da recursividade nas pastas filho. Se quiseres apenas algumas secções de uma grande árvore, podes efectuar checkout 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 o nível especificado na cópia de trabalho. Esta opção não é usada na caixa de diálogo do checkout, mas é a opção por defeito para todas as outras caixas de diálogo que têm a configuração de nível.

#### 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**.

Para facilmente seleccionar apenas os itens que pretendes para o checkout e forçar a cópia de trabalho a manter apenas esses itens, clica no botão **Escolher itens....** Isto abre uma nova caixa de diálogo onde podes verificar todos os itens que pretendes da tua cópia de trabalho e desseleccionar todos os itens que não pretendes. A cópia de trabalho resultante é conhecida como um `checkout disperso`. Uma actualização a tal cópia de trabalho não irá obter os ficheiros e pastas em falta, mas apenas actualizar o que já tens actualmente na tua cópia de trabalho.

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

#### 4.3.1.1. Actualização dispersa usando Actualizar para revisão

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

#### 4.3.1.2. Actualização dispersa usando o Navegador de Repositório

Right click on the checked out folder, then use TortoiseSVN → Repo-Browser to bring up the repository browser. Find the sub-folder you would like to add to your working copy, then use **Context Menu → Update item to revision....**

#### 4.3.1.3. Actualização dispersa usando Verificar Modificações

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

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. Checkout na pasta pai com o nível “Só este item”. Terás agora 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 o nível da cópia de trabalho

Uma vez que tenhas efectuado o checkout de uma cópia de trabalho, com um nível em particular, poderás modificar o mesmo mais tarde de modo a obter maior ou menor conteúdo utilizando, Context menu → Actualizar item para revisão.... Nessa caixa de diálogo, tem a certeza que verificas a caixa Tornar o nível persistente.



## Usando um servidor mais antigo

Servidores anteriores à versão 1.5 não compreendem o pedido de nível da cópia de trabalho, não lidando eficientemente com os pedidos. O comando continuará a funcionar, mas um servidor mais antigo poderá enviar todos os dados, deixando para o cliente a filtragem do que não precisa, o que poderá gerar muito tráfego de rede. Se possível debes actualizar o teu servidor para pelo menos 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 nível, 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 o nível 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 incluindo cada ramo e etiqueta do teu projecto!



## Exportando

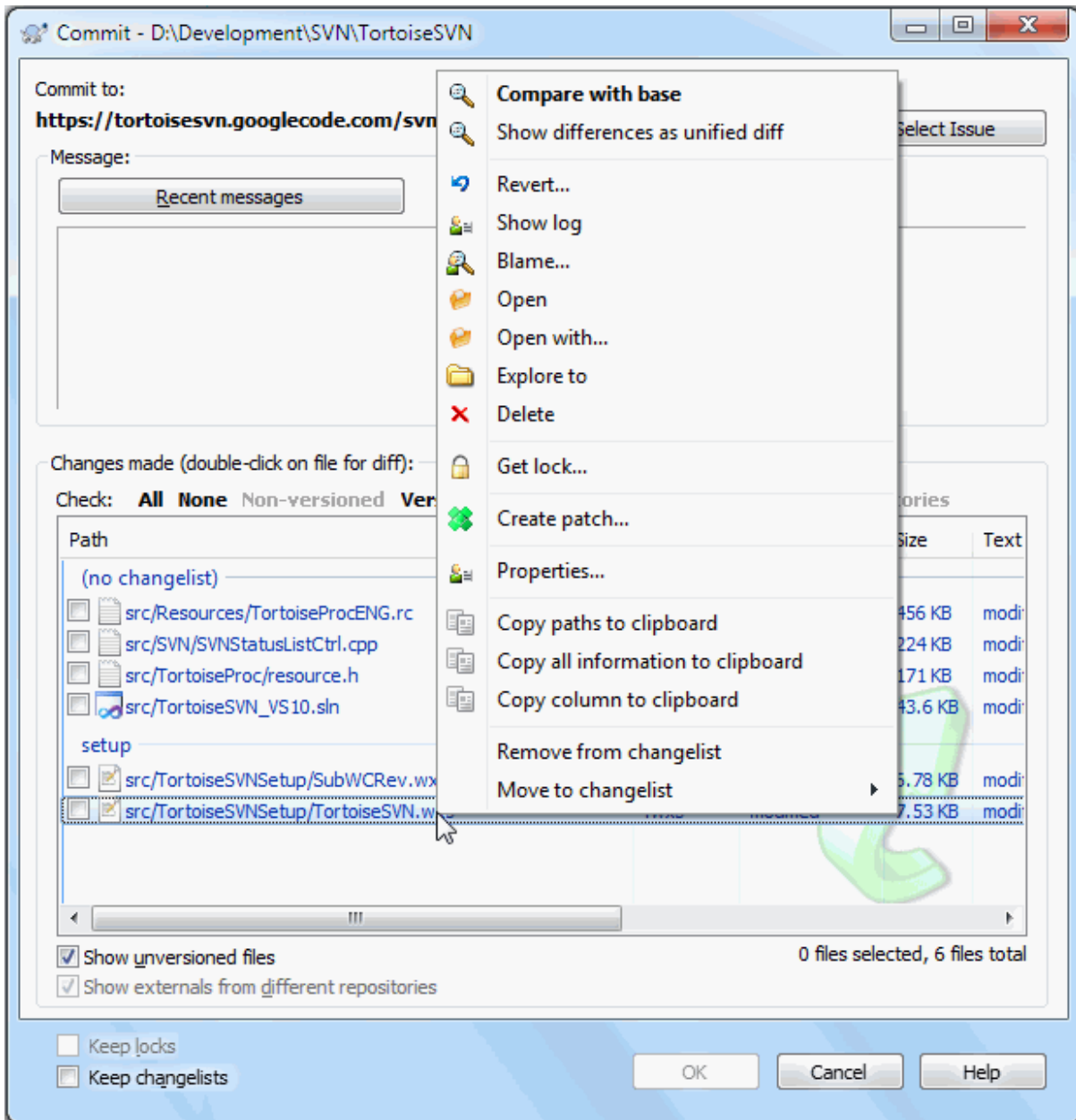
Por vezes quererás criar uma cópia local sem nenhuma das pastas .svn, i.e. criar uma tarball zipada do teu código fonte. Lê [Secção 4.27, “Exportar uma Cópia de Trabalho do Subversion”](#) para descobrires como o fazer.

## 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.

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

Para informação sobre as colorações das sobreposições de itens de acordo com o seu estado, por favor consulta [Secção 4.7.3, “Estado Remoto e Local”](#).

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.



## 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.
- add the file to the `svn:global-ignores` list using TortoiseSVN → Add to ignore list (recursively) This will affect the directory on which you set the `svn:global-ignores` property and all subfolders as well.

Ler [Secção 4.14, “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 4.31.1, “Preferências Gerais”](#) para mais informações em como editar o ficheiro de configuração do Subversion.



## Warning when committing to a tag

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

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

However, this check only works if the repository layout is one of the recommended ones, meaning it uses the names `trunk`, `branches` and `tags` to mark the three main areas. In case the setup is different, the detection of what is a tag/branch/trunk (also known as `classification`

patterns), can be configured in the settings dialog: [Secção 4.31.2, “Preferências do Gráfico de Revisões”](#)



## 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 desejas 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.

Arrastar os ficheiros da lista ao fundo da caixa de diálogo submeter para a caixa de edição das mensagens de registo, irá inserir nesta os caminhos de ficheiro como texto simples.



## 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 acerca 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



## Reparar Cópias Externas

Se fizeste uma cópia de um ficheiro e esqueceste de usar o comando do Subversion para o fazeres, poderás corrigir essa cópia para que o novo ficheiro não perca o seu histórico. Selecciona simplesmente ambos, o com o nome antigo (normal ou modificado) e o com o novo nome (não versionado) e usa Menu de Contexto → Reparar cópia para emparelhar os dois ficheiros com uma cópia.

### 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. Submeter apenas secções de ficheiros

Às vezes, poderás querer efectuar a submissão de apenas uma parte das alterações que efectuaste num ficheiro. Essa situação acontece com frequência quando estás a trabalhar em algo, e surge uma correcção urgente para ser submetida, e essa correcção calha estar no mesmo ficheiro em que estás a trabalhar.

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



## A Utilizar TortoiseMerge

If you use TortoiseMerge to edit the file, you can either edit the changes as you're used to, or mark all the changes that you want to include. right click on a modified block and use Context Menu

→ Mark this change to include that change. Finally right click and use Context Menu → Leave only marked changes which will change the right view to only include the changes you've marked before and undo the changes you have not marked.

Após submeter, a cópia do ficheiro é automaticamente restaurada e terás o ficheiro com todas as tuas modificações que não foram ainda submetidas.

#### 4.4.4. Excluir itens da lista a Submeter

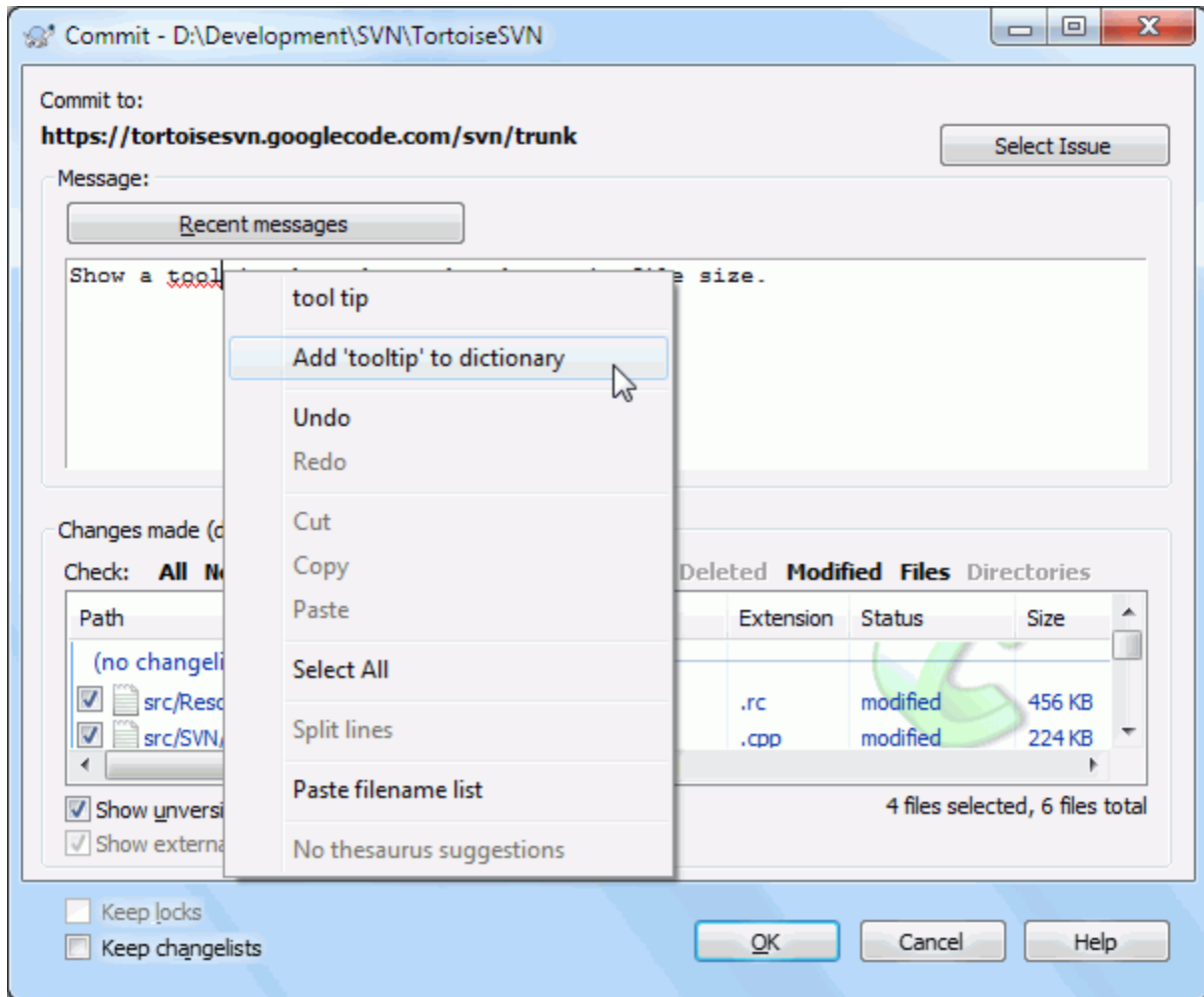
Por vezes, tens ficheiros versionados que são alterados frequentemente mas que não desejas submeter. Por vezes isso indica uma falha no processo de construção - porque são estes ficheiros versionados? devias ter usado ficheiros template? Mas por vezes é inevitável. A razão clássica é a mudança, pelo 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.

#### 4.4.5. 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.

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

Acertar com a regex correcta pode ser um bocado complicado, pelo que para te ajudar a descobrir qual a expressão mais indicada, existe uma caixa de diálogo de teste que te permite introduzir uma expressão e então digitar nomes de ficheiros para a testar. Arranca-a a partir da linha de comandos, com o comando `TortoiseProc.exe /command:autotexttest`.

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

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

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 4.18, “Configurações de Projecto”](#) para mais informação.

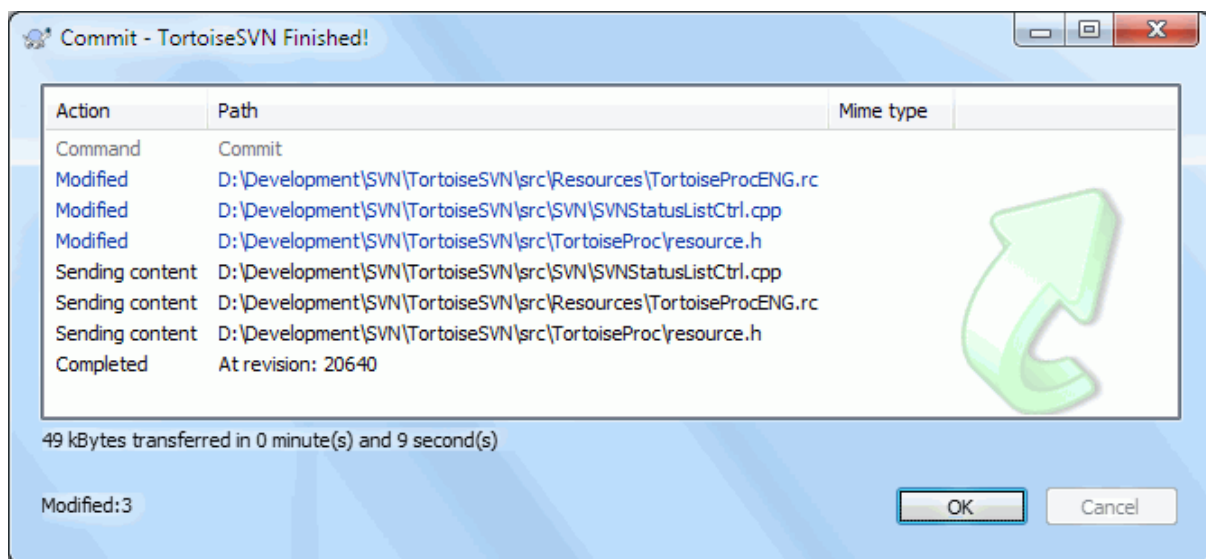


### Integração com Ferramentas de Controlo de Problemas

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 4.29, “Integração com Sistemas de controlo de problemas/Gestores de Problemas”](#).

## 4.4.6. Progresso do Submeter

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



**Figura 4.10. A caixa de diálogo 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

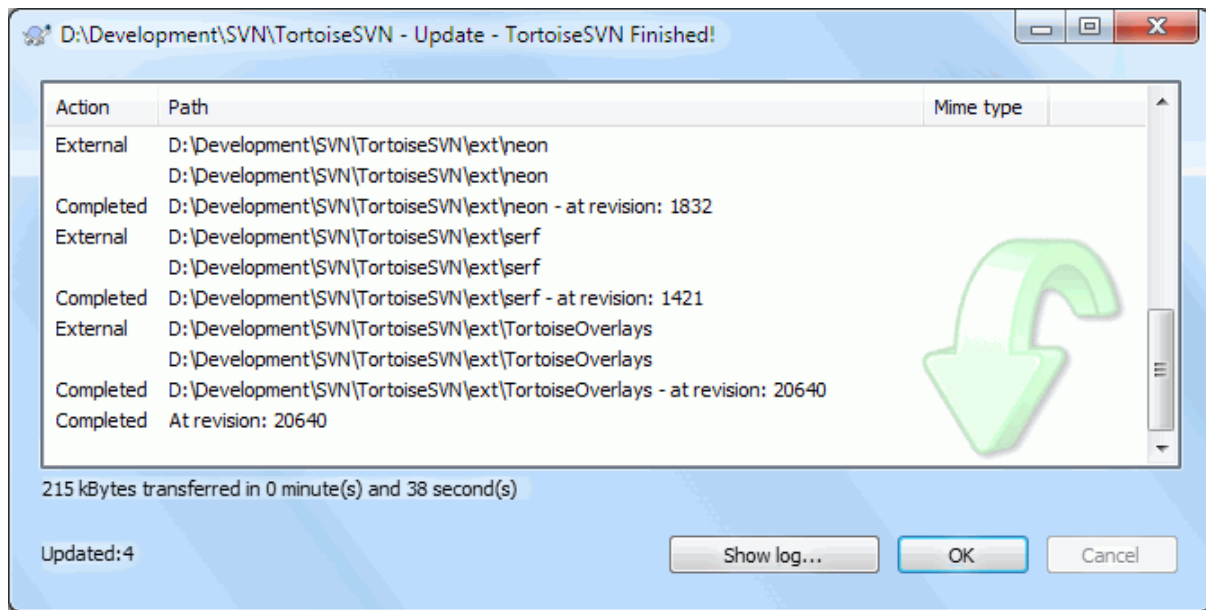
Submetendo uma substituição ou apagamento.

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 4.31.1.5, “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 4.31.1.5, “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 padrão Actualizar não tem opções e actualiza apenas a tua cópia de trabalho para a revisão HEAD do repositório, que é o caso mais comum. Se quiseres mais controlo sobre o processo de actualização, deverás então utilizar TortoiseSVN → Actualizar para revisão.... Este permite-te actualizar a tua cópia de trabalho para uma revisão específica, não apenas para a mais recente. Supõem que a tua cópia de trabalho está na revisão 100, mas queres reflectir o estado que tinha na revisão 50 - então actualiza simplesmente para a revisão 50.

Na mesma caixa de diálogo podes também escolher a *nível* para a qual actualizas a pasta corrente. Os termos usados são descritos em [Secção 4.3.1, “Profundidade do Checkout”](#). O nível por defeito é Cópia de trabalho, que preserva a definição de nível existente. Podes também fixar o nível persistente o que significa que actualizações subsequentes irão usar essa novo nível, i.e. esse nível será então usado como a nível por defeito.

Para tornar mais fácil incluir ou excluir itens específicos no checkout, clica no botão Escolher itens.... Isto abre uma nova caixa de diálogo onde podes seleccionar todos os itens que quiseres na tua cópia de trabalho, e desseleccionar is itens que não quiseres.

Poderás escolher se queres ignorar qualquer projecto externo na actualização (i.e projectos referenciados com `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.

Actualizar 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.

## 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. The conflicting area in a text file is marked like this:

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

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

filename.ext.mine

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 lançar uma ferramenta externa de integração/ editor de conflitos com TortoiseSVN → Editar Conflitos ou poderás utilizar um qualquer editor de texto para resolver o conflito manualmente. Deverás decidir como é que irá ficar o código, efectua as alterações necessárias e guarda o ficheiro. Usando uma ferramenta de integração como o TortoiseMerge ou uma das outras ferramentas populares, é na generalidade a opção mais fácil já que apresentam na generalidade os ficheiros envolvidos numa painel de 3 vistas e não têm de se preocupar com os marcadores de conflitos. Se usares um editor de ficheiro terás então de pesquisar por linhas começadas pela string <<<<<<<.

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 Propriedade

Um conflito de propriedade ocorre quando dois (ou mais) programadores alteraram a mesma propriedade. Tal como no caso dos ficheiros, deve-se exclusivamente aos programadores a resolução do conflito.

Se uma das alterações tiver de se sobrepor a outra, então escolhe a opção para Resolver usando a propriedade local ou Resolver usando a propriedade remota. Se as alterações tem de ser integradas selecciona então Edita manualmente a propriedade, decide o valor que a propriedade deve ter e marca como resolvido.

#### 4.6.3. 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.

##### 4.6.3.1. 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.

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

##### 4.6.3.2. 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.

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

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

#### 4.6.3.3. 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.

#### 4.6.3.4. 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.

#### 4.6.3.5. 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.
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.

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

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

#### 4.6.3.6. 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.

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

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.

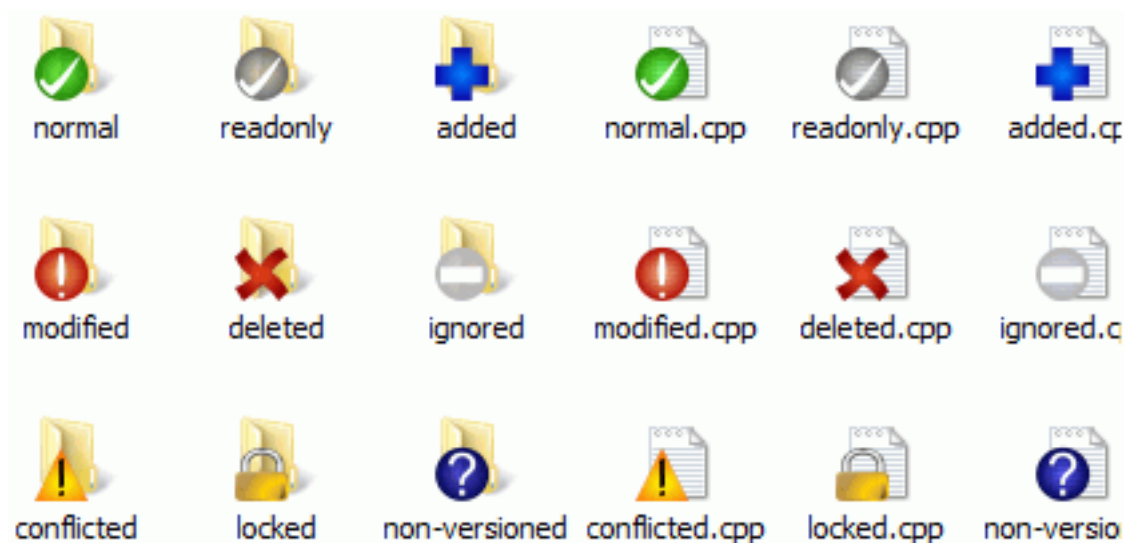
#### 4.6.3.7. Outros conflitos de árvore

Existem outros casos que estão etiquetados como conflitos de árvore, simplesmente porque o conflito envolve uma pasta ao invés de um ficheiro. Por exemplo, se adicionastes uma pasta com o mesmo nome para o trunk e para um ramo, e tentares integrar, irás então obter um conflito de árvore. Se quiseres manter a pasta do destino da integração, marca o conflito como resolvido. Se quiseres usar a pasta da origem da integração, então precisas de SVN remover primeiro a do destino de integração, e correr de novo a integração. Se precisares de algo mais complicado, então tens de resolver o conflito manualmente.

### 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 se 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.



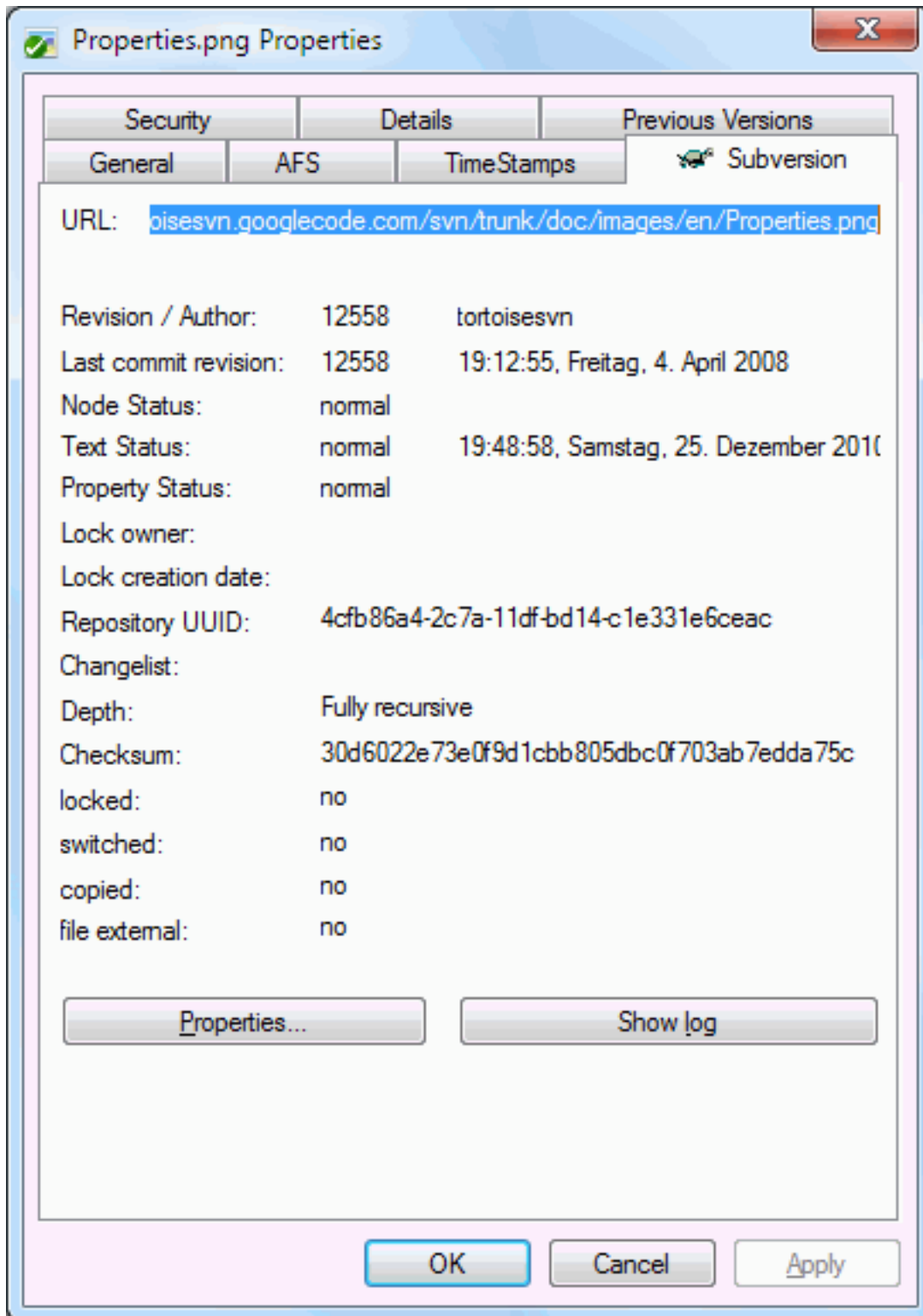
Este ícone mostra os ficheiros e pastas que não estão sob controle de versões, mas não foram ignorados. Esta sobreposição é opcional.

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 vários clientes Tortoise (TortoiseCVS, TortoiseHg, ...) a limitação do ícone torna-se um problema real. Para contornar isso, o projecto TortoiseSVN introduziu um conjunto de ícones comuns e partilhados, carregados como uma DLL, que podem ser usados por todos os clientes Tortoise. Verifique com o seu provedor de cliente para ver se isso 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. Estado detalhado



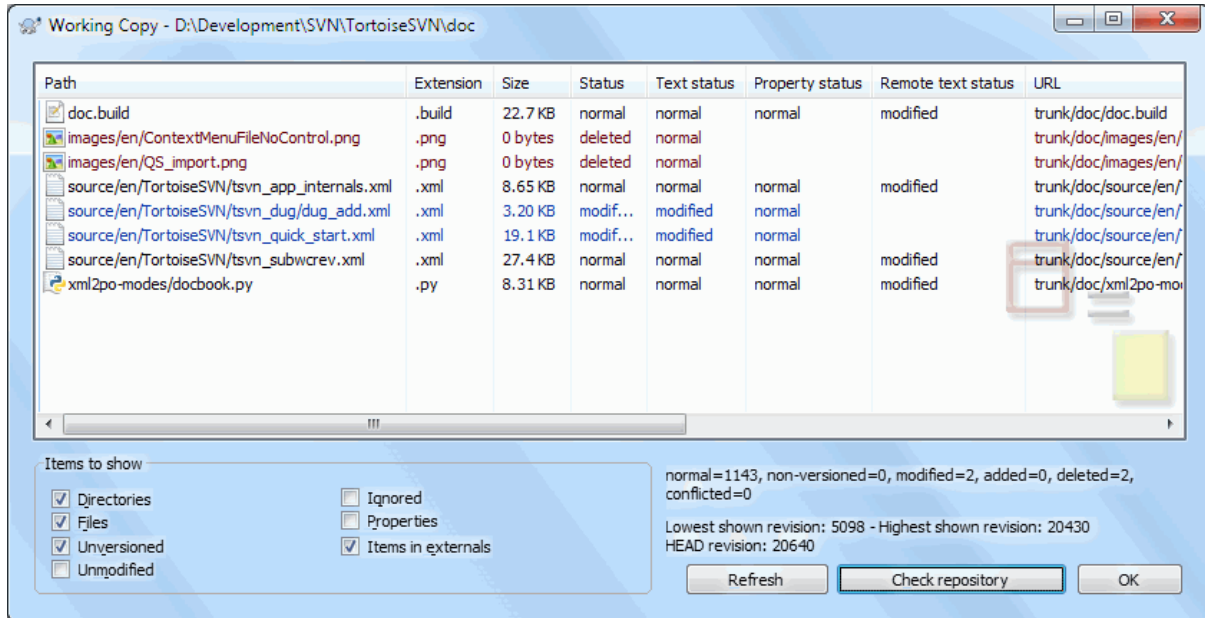
**Figura 4.13. Página de propriedades do explorador, aba do Subversion**

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.

### 4.7.3. Estado Remoto e Local



**Figura 4.14. 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

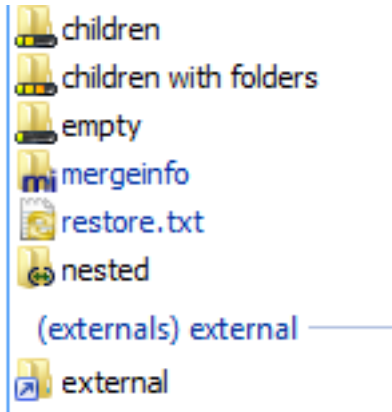
Itens modificados localmente e apagados no repositório, ou modificados no repositório e apagados localmente. Estes *irão* produzir conflitos na actualização.

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 4.31.1.5, “Preferências de cor do TortoiseSVN”](#) para mais informação.

Ícones de sobreposição são usados para indicar também outros estados. A captura de ecrã abaixo mostra todas as sobreposições possíveis que poderão ser mostradas se necessário.



São mostradas sobreposições para os seguintes estados:

- Nível de checkout vazia, significa apenas o item.
- Nível de checkout ficheiros, significa só o próprio item e todos os ficheiros sem pastas filho.
- Nível de checkout imediatos, significa só próprio item, todos os ficheiros e pastas filho, mas sem os filhos das pastas filho.
- Itens aninhados, i.e., cópias de trabalho dentro da cópia de trabalho.
- Itens externos, i.e., todos os itens que são adicionados via a propriedade `svn:externals`.
- Itens que são restaurados após uma submissão. Consulta [Secção 4.4.3, “Submeter apenas secções de ficheiros”](#) para detalhes.
- Itens que têm modificações de propriedade, mas apenas na propriedade `svn:mergeinfo`. Se qualquer outra propriedade estiver modificada, não será usada a sobreposição.

Itens que foram trocados para um caminho diferente no repositório são também sinalizados com um marcador (s). Podes ter trocado algo enquanto trabalhavas num ramo e te esqueceste de trocar de novo para o trunk. Isto é o teu sinal de aviso! O menu de contexto permite-te trocar de volta para o caminho normal mas uma vez.

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 → Remover. Se quiseres remover permanentemente os ficheiros ( ignorando o cesto de reciclagem) mantém premido a tecla **Shift** enquanto clicas no Remover.

Se tu quiseres examinar um ficheiro em detalhe, podes arrastá-lo daqui para outro aplicativo, como um editor de texto ou IDE, ou podes salvar uma cópia simplesmente arrastando-o para uma pasta do navegador.

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 preservadas 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**



### Reparar Cópias Externas

Se fizeste uma cópia de um ficheiro e esqueceste de usar o comando do Subversion para o fazeres, poderás corrigir essa cópia para que o novo ficheiro não perca o seu histórico. Selecciona simplesmente ambos, o com o nome antigo (normal ou modificado) e o com o novo nome (não versionado) e usa **Menu de Contexto → Reparar cópia** para emparelhar os dois ficheiros com uma cópia.

## 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.

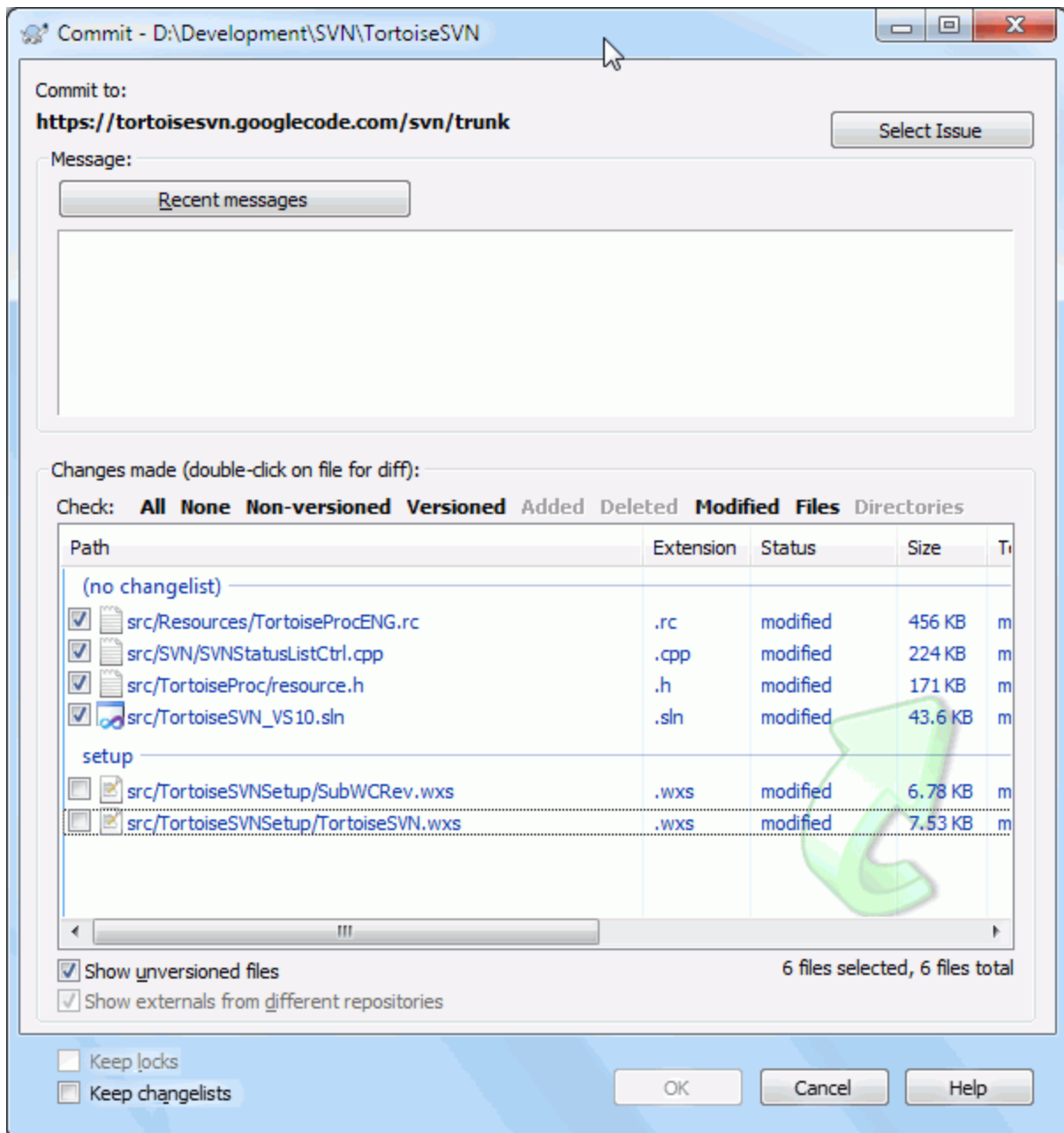
## 4.8. Listas de Alterações

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.

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 4.15. 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.

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 4.4.4, “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.



### Atenção

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

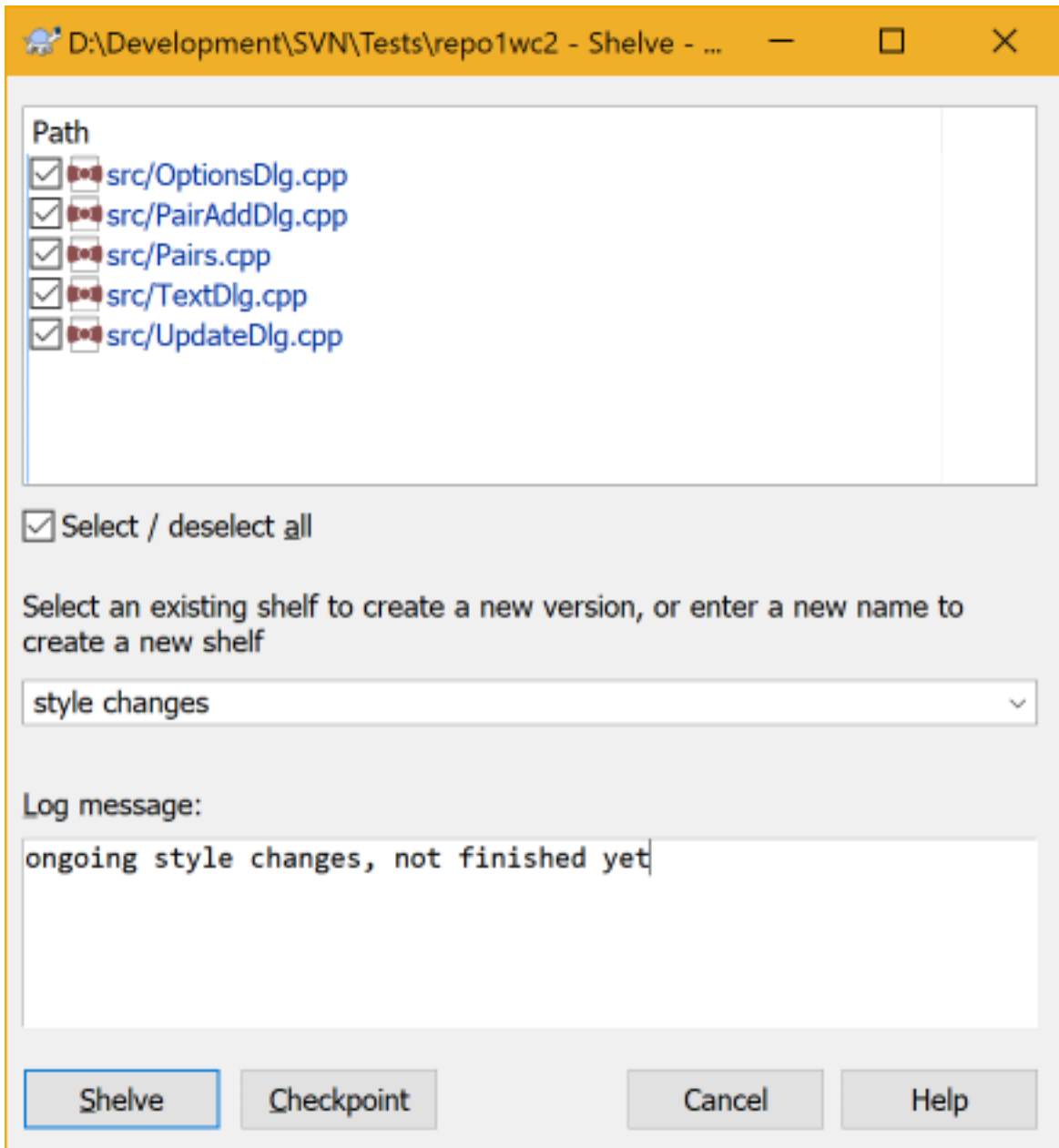
## 4.9. Shelving

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

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

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

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



**Figura 4.16. Shelve dialog**

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

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

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

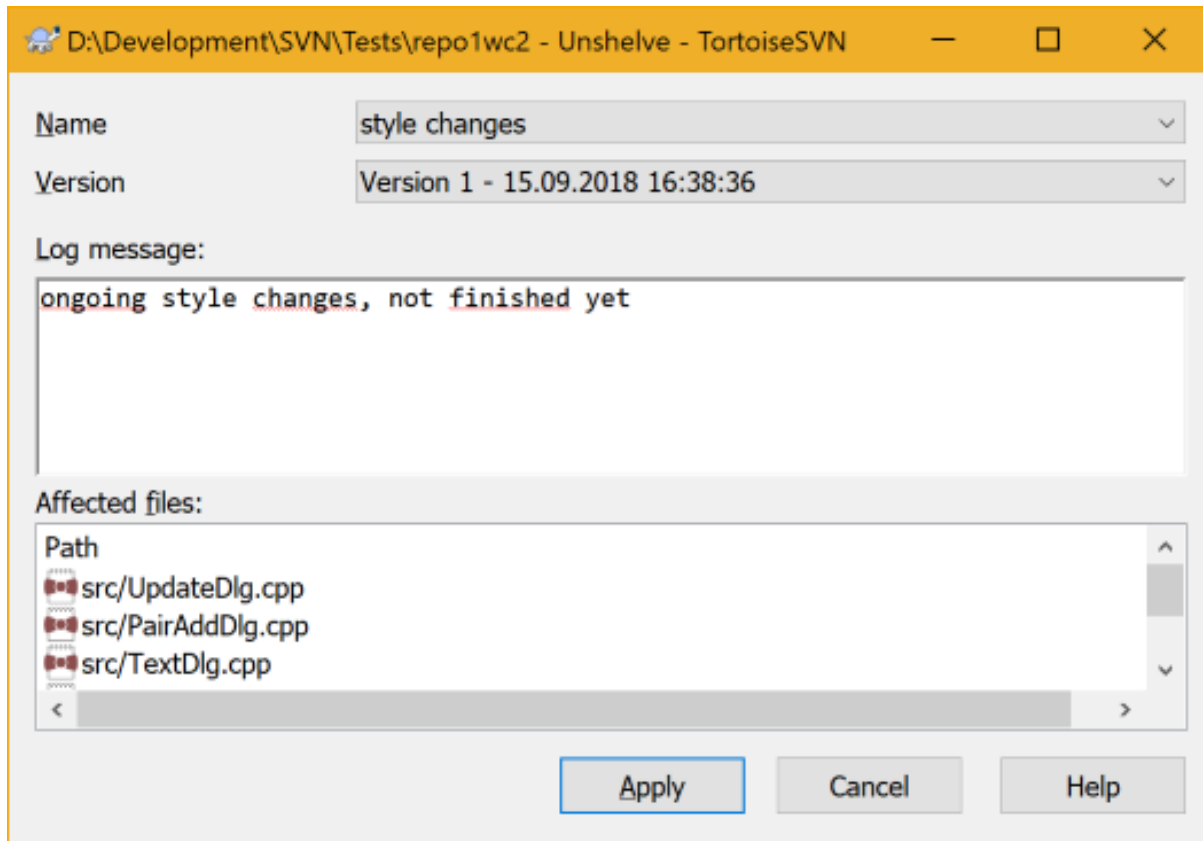


Figura 4.17. Unshelve dialog



### Dica

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



### Experimental

The shelving feature is still marked as experimental.

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

## 4.10. Caixa de Diálogo Registo de Revisões

Para cada alteração que fazes e submetes, deves 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.

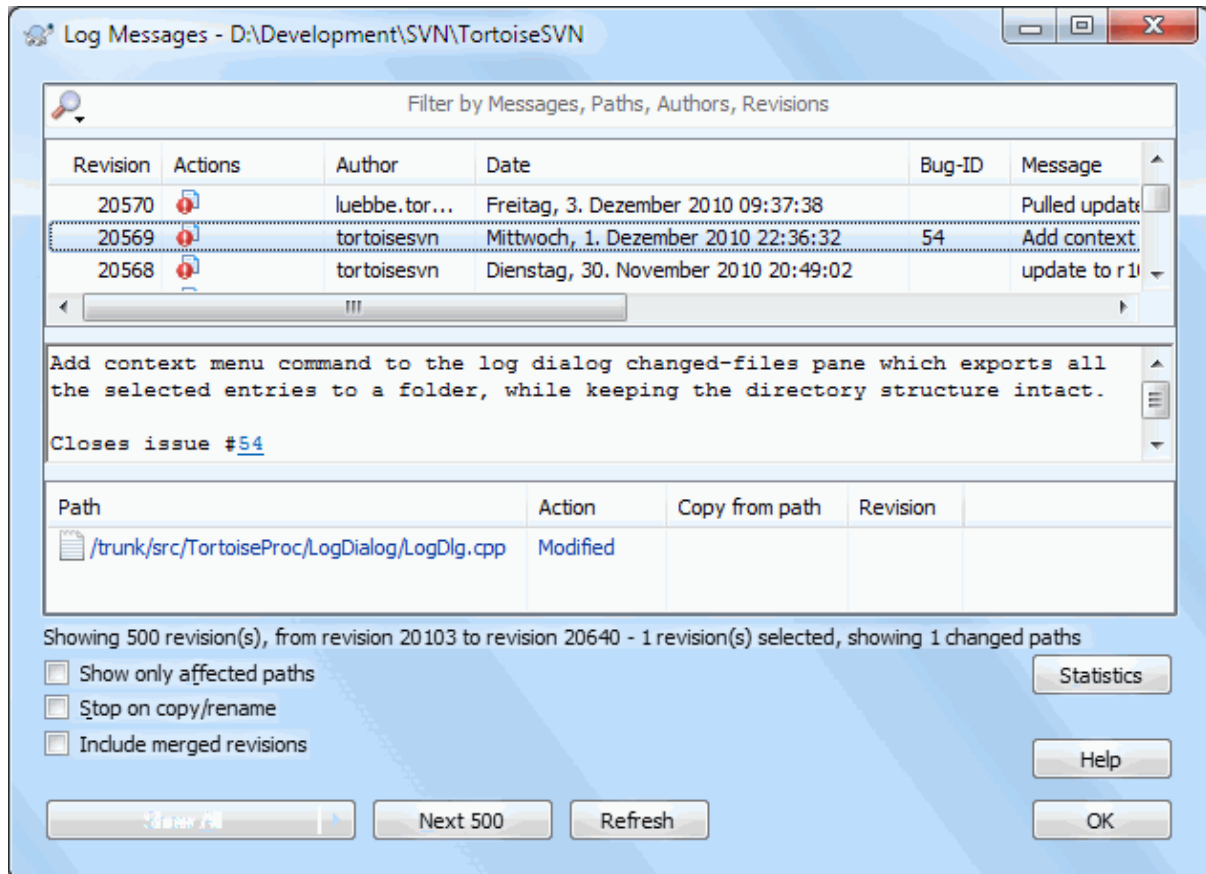
Linhas mostradas a azul indicam que algo foi copiado para esta linha de desenvolvimento (se calhar de um ramo).



- 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.

#### 4.10.1. Invocando a Caixa de Diálogo Registo de Revisão



**Figura 4.18. 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
- From the repository browser

Se o repositório não está disponível, verás a caixa de diálogo *Queres ficar fora de linha?* descrita em [Secção 4.10.10, “Modo Fora de Linha”](#).

#### 4.10.2. 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.



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



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

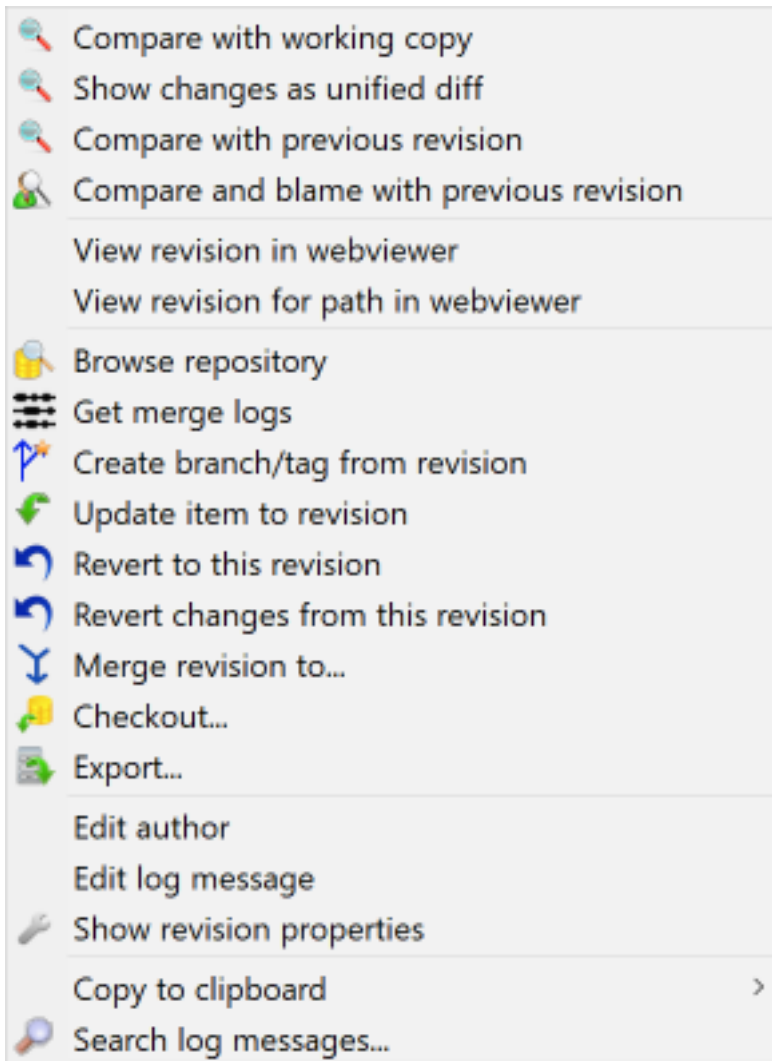


If a revision merged a file or directory, the *merged* icon is shown in the fourth column.



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

### 4.10.3. Obtendo Informação Adicional



**Figura 4.19. O Painel de Topo da Caixa de Diálogo Registo de Revisões com Menu de Contexto**

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

#### 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, então compara os relatórios de responsabilidade usando a ferramenta de comparação visual. Lê [Secção 4.24.2, “Diferenças de responsabilidade”](#) para mais detalhes. (apenas 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.

Se mantiveres premida a tecla **Shift** enquanto clicas no item do menu, aparecerá primeiro uma caixa de diálogo onde podes configurar as opções para a comparação unificada. Essas opções incluem a possibilidade de ignorar alterações em fins-de-linha e espaços-em-branco.

#### 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, então compara os resultados usando a ferramenta de comparação. (apenas pastas)

#### Guardar revisão em...

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

#### Responsabilizar...

Responsabiliza o ficheiro até à revisão seleccionada (apenas 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. Isot 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 4.10.7, "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 4.10.7, "Alterando a Mensagem de Registo e Autor"](#).

#### Copiar para o clipboard

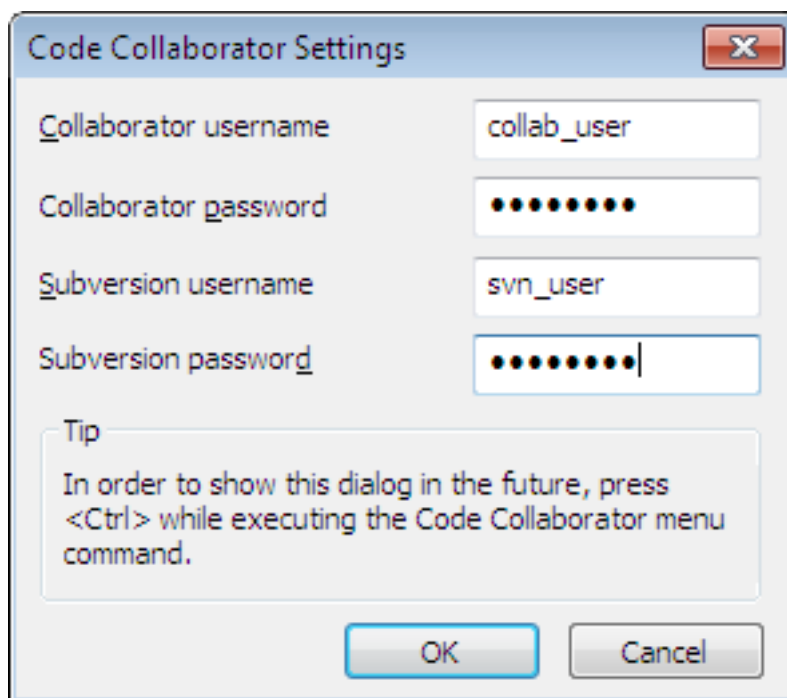
Copia os detalhes do registo das revisões seleccionadas para o clipboard. Isto irá copiar o número de revisão, autor, data, mensagem de registo e a lista de itens alterados para cada revisão.

#### Pesquisar mensagens de registo...

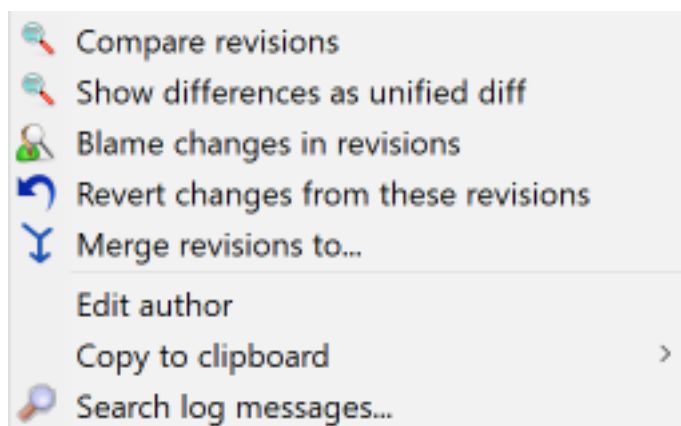
Pesquisar mensagens de registo com o texto que inseriste. É feita a pesquisa das mensagens de registo que inseriste e também o sumário de acções criada pelo Subversion (mostrado no painel inferior). A pesquisa não é sensível à altura da letra.

#### Create code collaborator review...

This menu is shown only if the SmartBear code collaborator tool is installed. When invoked for the first time, a dialog is shown prompting the user to enter user credentials for both code collaborator and SVN. Once the settings are stored, the settings dialog is no longer shown when the menu is invoked, unless the user holds **Ctrl** while executing the menu item. The configuration and the chosen revision(s) are used to invoke the code collaborator graphical user interface client, which creates a new review with the selected revisions.



**Figura 4.20. The Code Collaborator Settings Dialog**



**Figura 4.21. 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 4.11.3, “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 4.24.2, “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.

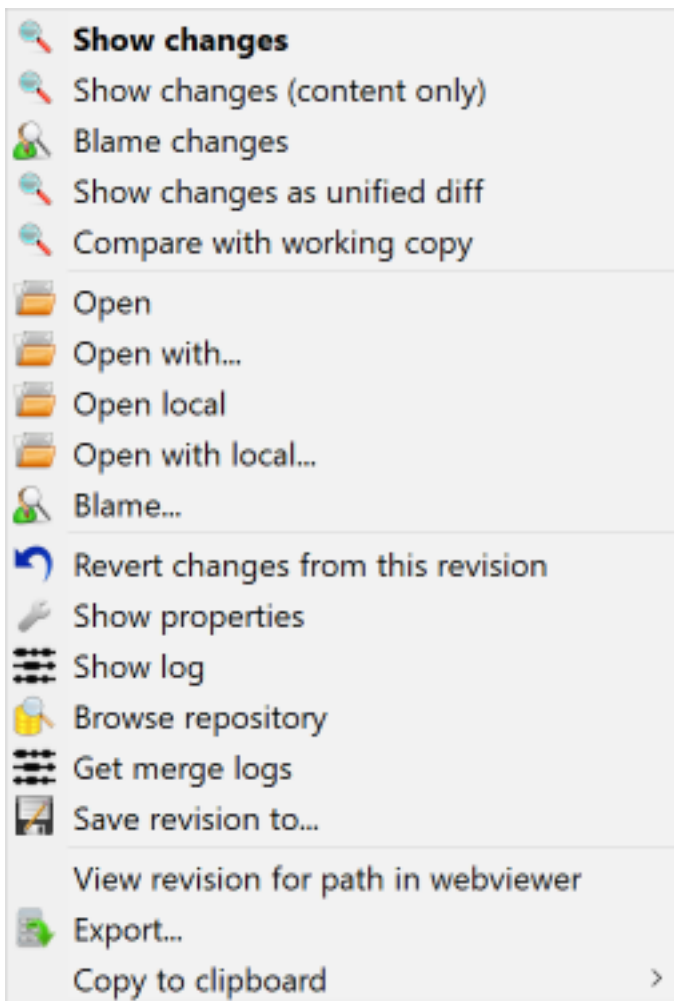
**Pesquisar mensagens de registo...**

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 4.22. O Painel de fundo da Caixa de Diálogo Registo com o Menu de Contexto**

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

**Mostrar alterações**

Mostrar alterações feitas na revisão seleccionada para o ficheiro seleccionado.

**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 4.24.2, “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 4.10.6, “Funcionalidades de Rastreamento de Integração”](#).

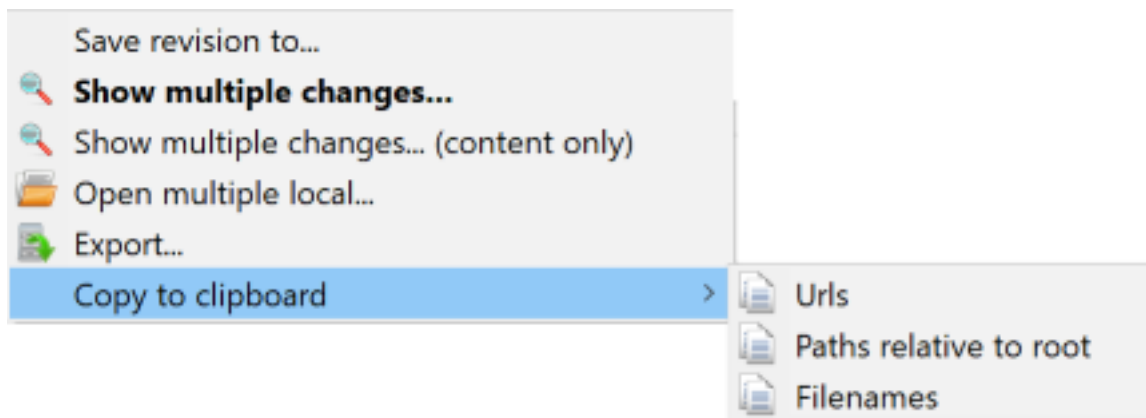
Guardar revisão em...

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

Exportar...

Exporta os itens seleccionados nesta revisão para uma pasta, preservando a hierarquia de ficheiros.

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



**Figura 4.23. The Log Dialog Bottom Pane with Context Menu When Multiple Files Selected.**

Guardar revisão em...

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

Mostrar múltiplas alterações ...

Show changes made in the selected revision for the selected files. Note that the show changes functionality is invoked multiple times, which may bring up multiple copies of your selected diff tool, or just add a new comparison tab in your diff tool. If you have selected more than 15 files, you will be prompted to confirm the action.

Abrir múltiplos locais ...

This will open local working copy files that correspond to your selected files using the application that is registered for the extension. [The behavior is the one you would get double-clicking the working-copy file(s) in Windows explorer]. Depending on how your file extension is associated to an application and the capabilities



of the application, this may be a slow operation. In the worst case, new instances of the application may be launched by Windows for each file that was selected.

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

Exportar...

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



### Dica

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

O Subversion usa números de revisão para representar 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 essa revisão, e.g. “Feito em r1234” significa que as alterações submetidas na r1234 implementam 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*.

#### 4.10.4. 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 4.31.1.2, “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.

Para refrescar a última versão, no caso de ter ocorrido outras submissões enquanto a caixa de diálogo de registo estava aberta, prime a tecla **F5**.

Para refrescar a cache de registo, prime as teclas **Ctrl-F5**.

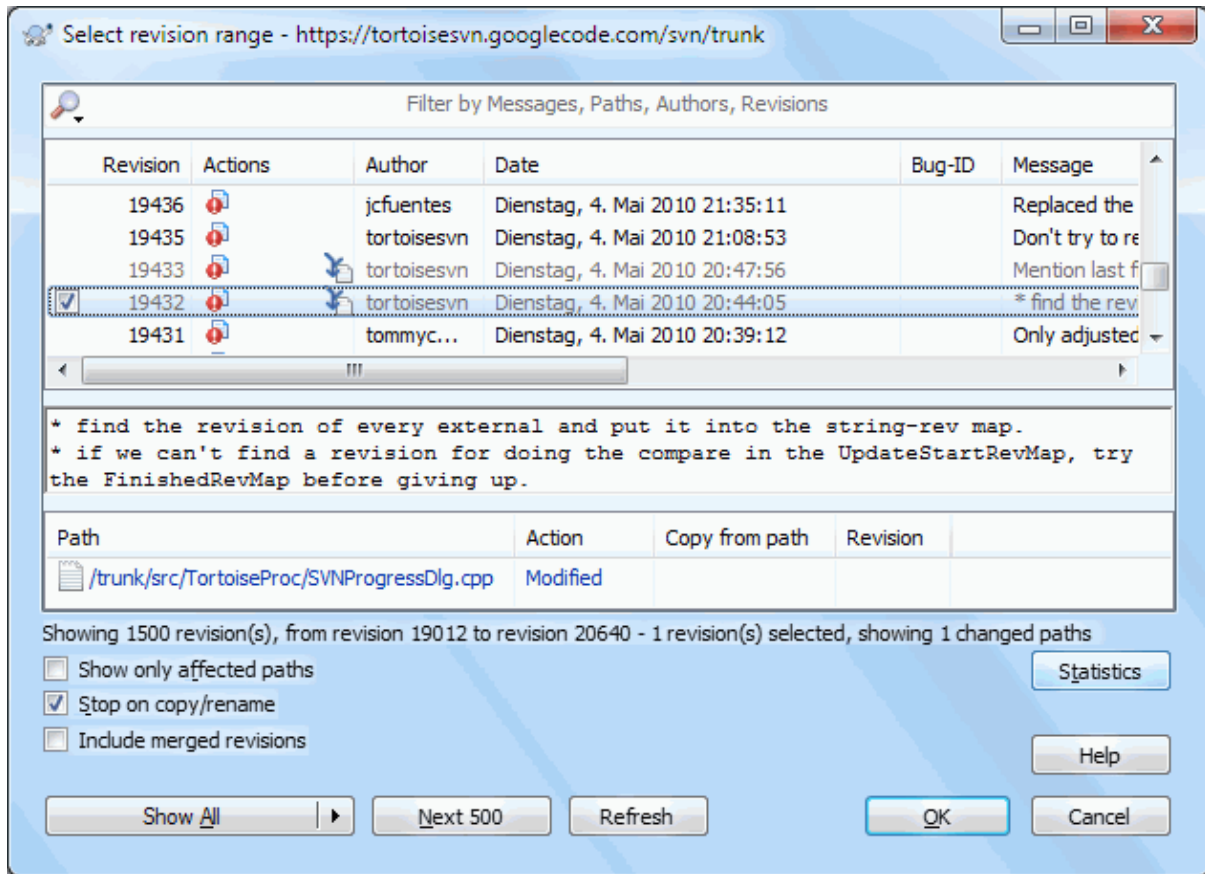
#### 4.10.5. Revisão Actual da Cópia de Trabalho

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.

Quando mostras o registo para uma pasta, a revisão realçada é a mais alta revisão encontrada em qualquer ponto dentro dessa pasta, o que requiere o rastreamento da cópia de trabalho. O rastreamento é feito numa thread separada, para que não atrase a visualização do registo, mas como resultado o realce para pastas poderá não aparecer imediatamente.

#### 4.10.6. 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 4.24. 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.

#### 4.10.7. 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...

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

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*.



## Importante

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

### 4.10.8. 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.

Clica no ícone de pesquisa para seleccionar qual a informação que queres procurar, e para escolher o modo *regex*. Normalmente só precisarás uma simples procura de uma sub-cadeia de caracteres, mas se precisas de termos de pesquisa mais flexíveis podes usar expressões regulares. Se mantiveres o rato sobre a caixa, uma etiqueta dar-te-á palpites em como usar as funções regex, ou as funções de sub-cadeias de caracteres. O filtro funciona verificando se a tua frase de pesquisa condiz com as entradas do registo, então só as entradas que *condizem* com a frase do filtro são mostradas.

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

```
Alice Bob -Eve
```

searches for strings containing both Alice and Bob but not Eve

```
Alice -Bob +Eve
```

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

```
-Case +SpecialCase
```

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

```
!Alice Bob
```

searches for strings which do not contain both Alice and Bob

```
!-Alice -Bob
```

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

```
"Alice and Bob"
```

searches for the literal expression "Alice and Bob"

```
" "
```

searches for a double-quote anywhere in the text

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

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

Descrever o uso de expressões regulares em buscas está para além do âmbito deste manual, mas podes encontrar documentação em linha e um tutorial em <http://www.regular-expressions.info/>.

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

Podes também filtrar os nomes de caminhos, no painel de fundo, usando a caixa de verificação **Mostrar só caminhos afectados**. Caminhos afectados são aqueles que contêm o caminho usado para mostrar o registo. Se fores buscar o registo para uma pasta, isso significa qualquer coisa nessa pasta ou abaixo. Para um ficheiro isso significa só um ficheiro. Normalmente a lista de caminhos mostra alguns outros caminhos que foram afectados pela mesma submissão, mas a cinzento. Se a caixa estiver verificada esses caminhos são escondidos.

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 4.18.2, "Propriedades de Projecto TortoiseSVN"](#) para descobrir como usar esta propriedade.



### Registos Não Formatados do Navegador de Repositório

Because the formatting depends upon accessing Subversion properties, you will only see the results when using a checked out working copy. Fetching properties remotely is a slow operation, so you will not see this feature in action from the repo browser.

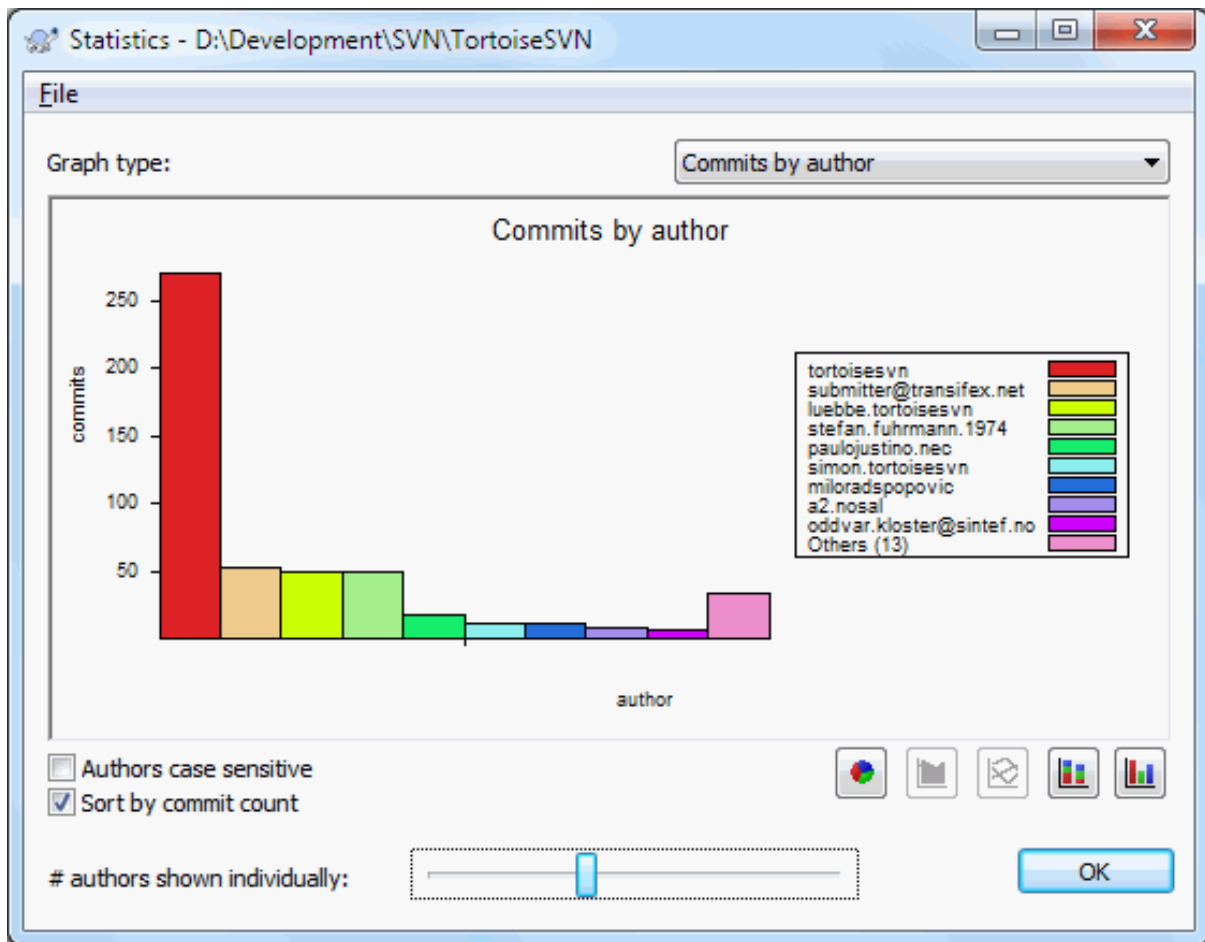
## 4.10.9. Informação Estatística

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 ;-)

### 4.10.9.1. Página de Estatísticas

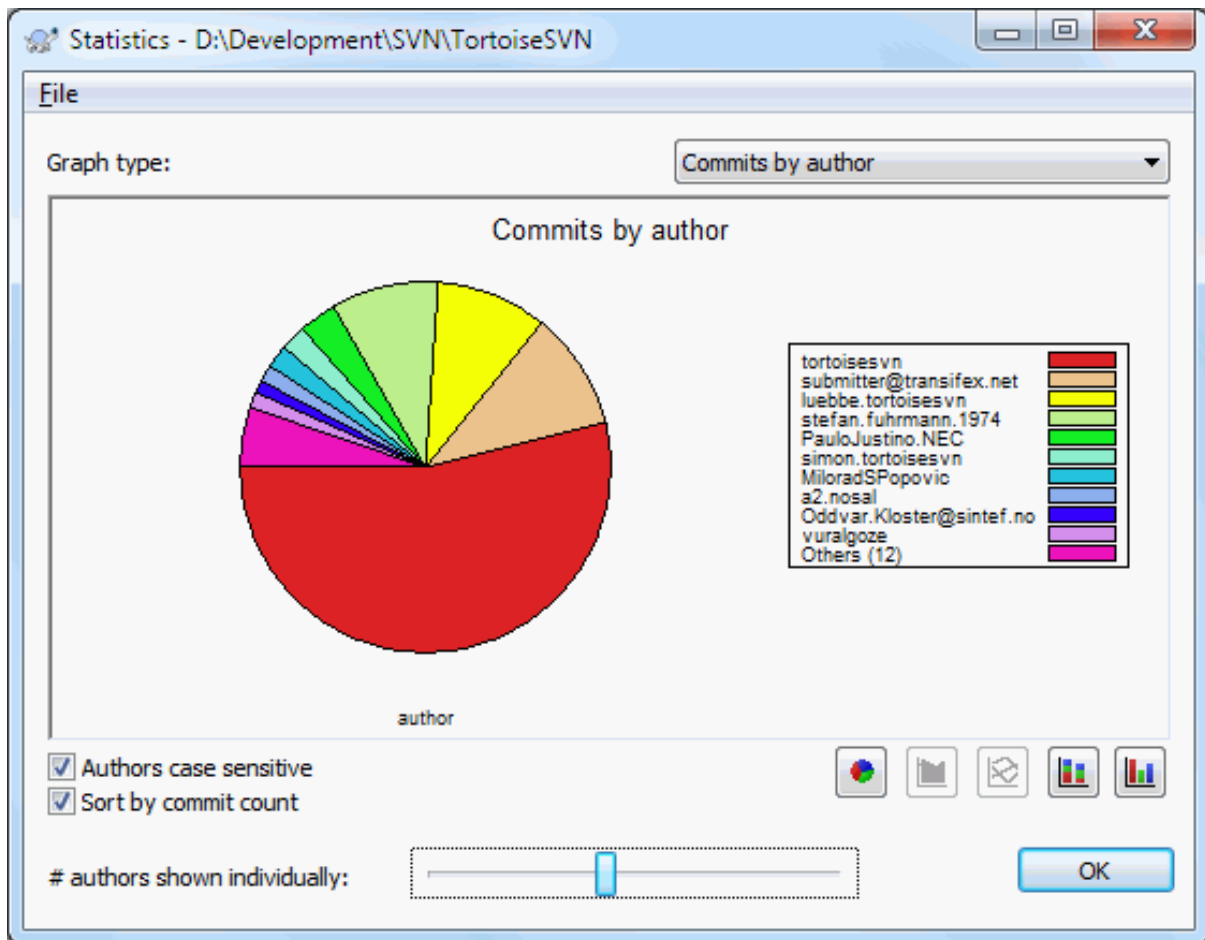
Esta página dá-te todos os números que tenhas em mente em particular, o período e o número de revisões cobertas e alguns valores min/max/média.

#### 4.10.9.2. Página de Submissões por Autor



**Figura 4.25. Histograma de Submissões-por-Autor**

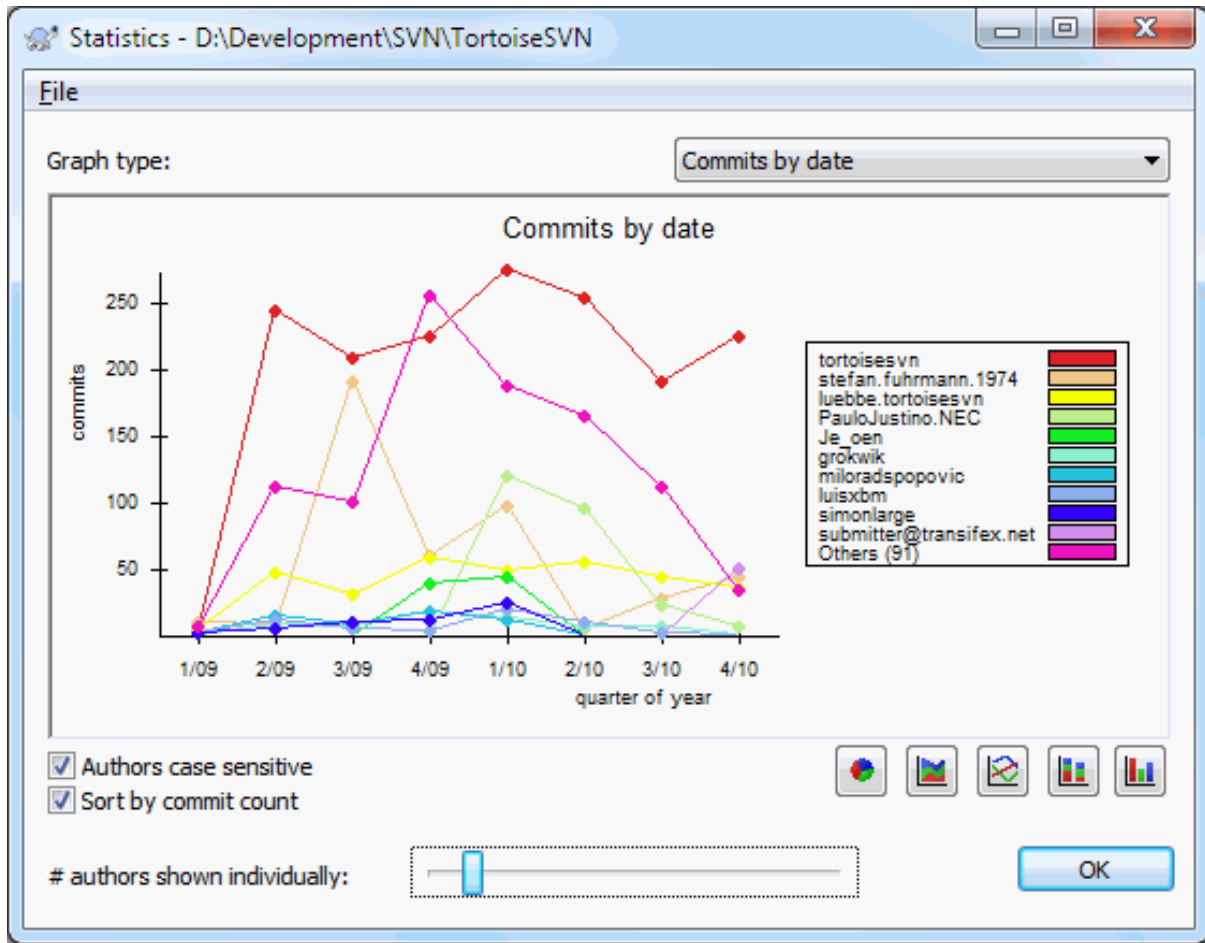
Este grafo mostra quais os autores que têm estado activos no projecto como um simples histograma, histograma empilhado ou gráfico de queijo.



**Figura 4.26. 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*.

### 4.10.9.3. Página de Submissões por data



**Figura 4.27. 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.

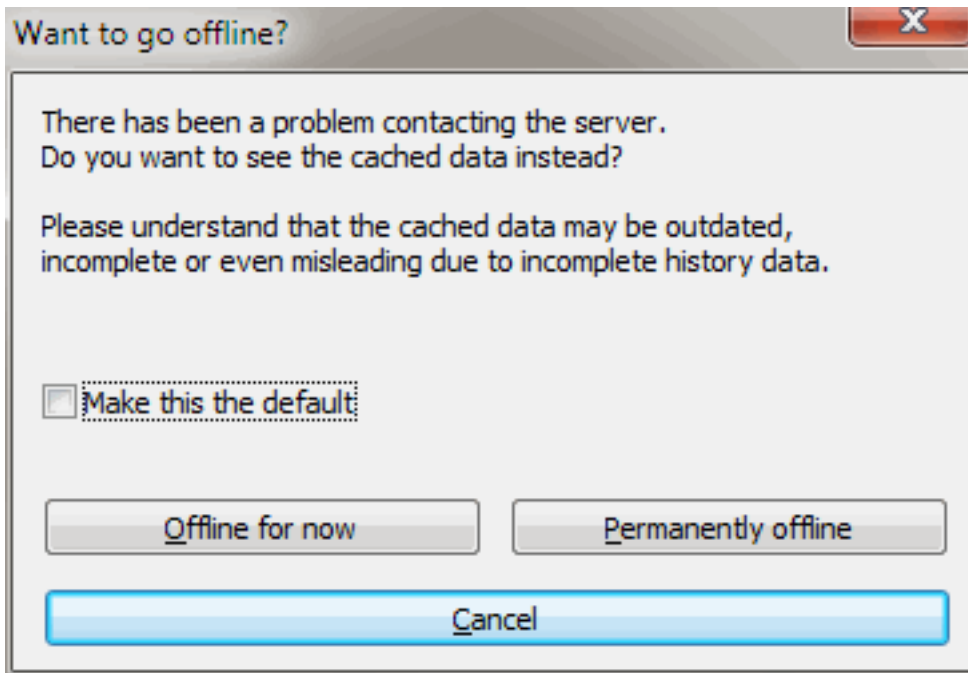
Quando existem vários autores irás obter muitas linhas no gráfico. Existem duas vistas disponíveis aqui: *normal*, onde a actividade de cada autor é relativa à linha base e, *empilhada*, onde a actividade de cada autor é relativa à linha de baixo. A última opção evita o cruzamento de linhas que pode 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.



#### 4.10.10. Modo Fora de Linha



**Figura 4.28. 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 4.10.11, “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.

#### 4.10.11. 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 estás 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.

## 4.11. 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 quiseses, podes usar o teu programa de comparação favorito.

### 4.11.1. Diferenças em ficheiros

#### Alterações locais

Se pretendes ver que alterações *tu* efectuastes 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 4.11.3, “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 comparar (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 4.11.3, “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.

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

#### 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 4.24.2, “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 4.11.6, “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 4.31.5, “Preferências de Programas Externos”](#) para descobrires como configurar outras ferramentas de comparação.

### 4.11.2. 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.

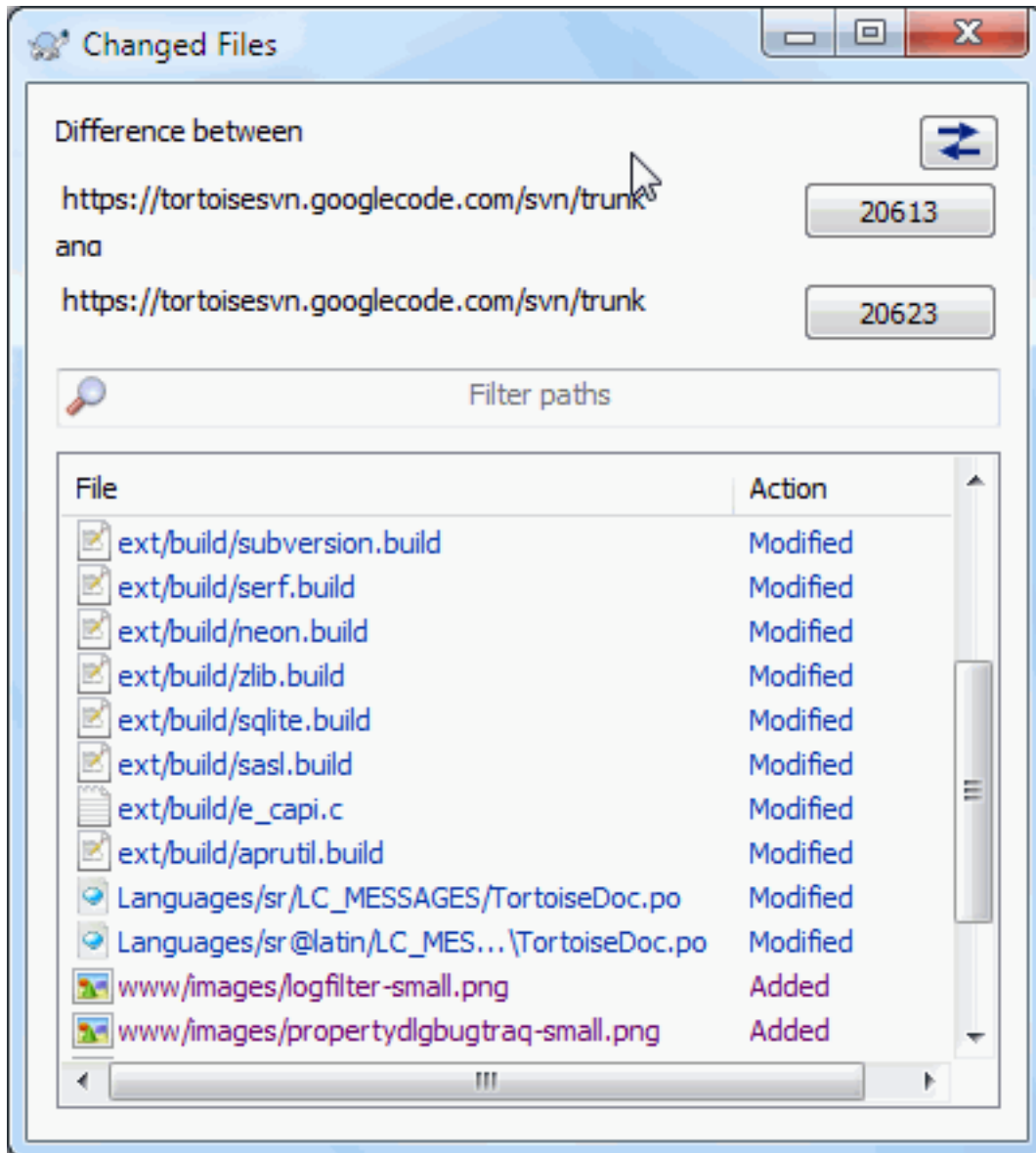
Comparar espaços em branco inclui todas as alterações na indentação e espaços-em-branco em linha, como linhas adicionadas/removidas.

Ignorar alterações de espaços em branco exclui alterações que são exclusivamente devidas à mudança na quantidade ou tipo de espaços em branco, e.g. alterando a indentação ou alterando tabulações para espaços. Adicionando espaços em branco onde antes 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.

### 4.11.3. Comparando Pastas



**Figura 4.29. 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.

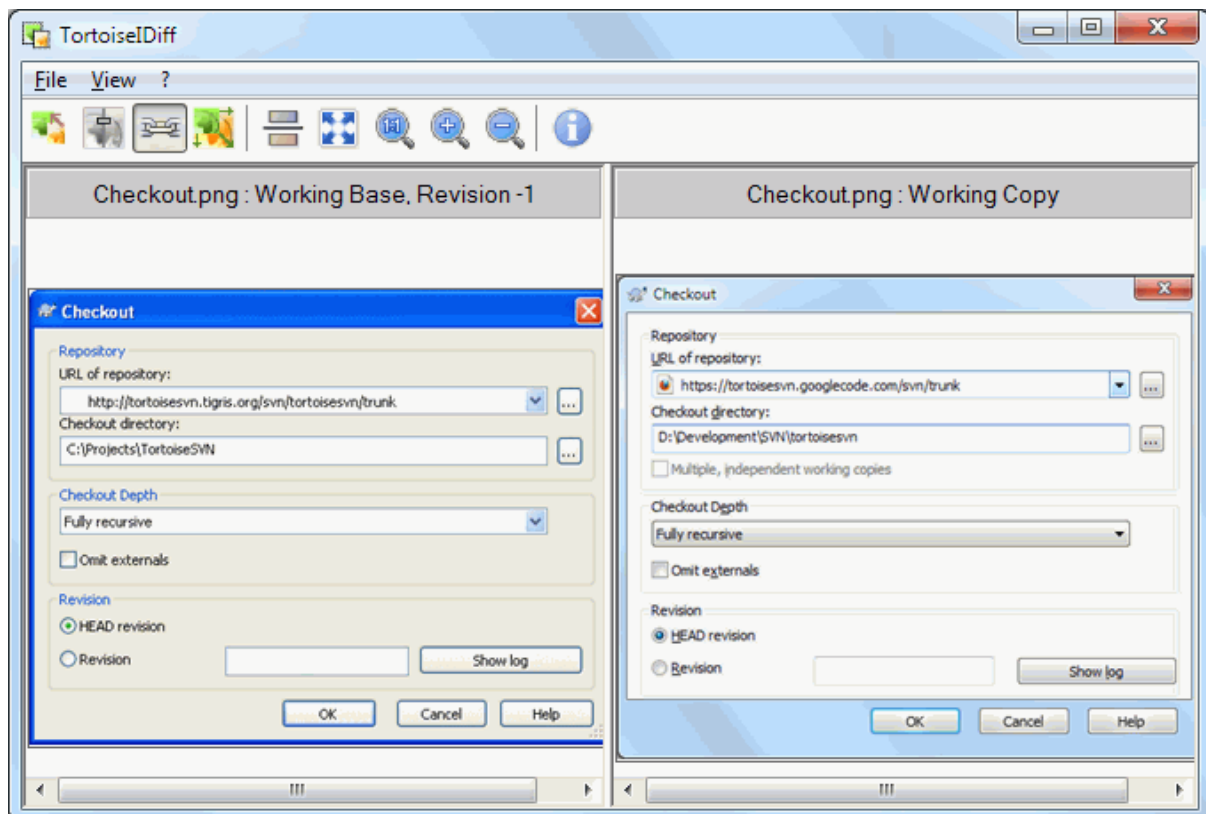
O botão no topo permite-te alterar a direcção da comparação. Podes mostrar as alterações necessárias para ir de A a B, ou se preferires, de B a A.

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`.

#### 4.11.4. 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 4.30. 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 nível 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-Rod** 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.

#### 4.11.5. Comparando Documentos Office

Quando queres comparar documentos não-textuais normalmente terás de utilizar o software usado para criar o documento, já que ele entende o formato do ficheiro. Para as suites do Microsoft Office e Open Office, normalmente usadas, existe realmente algum suporte para a visualização das diferenças e o TortoiseSVN inclui scripts para os invocar com as definições correctas ao comparar ficheiros com essas extensões bem conhecidas. Podes verificar que extensões de ficheiros são suportadas e adicionar a tua própria indo à secção TortoiseSVN → Definições e clicar Avançado em Programas Externos



#### Problemas com o Office 2010

Se instalaste a versão do Office 2010 *Click-to-Run* e tentares compara documentos, poderás obter uma mensagem de erro do Windows Script Host como esta: “ActiveX component can't create object: word.Application”. Parece que terás de usar a versão do Office baseada em MSI para obteres a funcionalidade de comparação.

#### 4.11.6. 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* [<https://winmerge.sourceforge.net/>] is a great open-source diff tool which can also handle directories.

##### Perforce Merge

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

##### KDiff3

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

##### SourceGear DiffMerge

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

##### 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

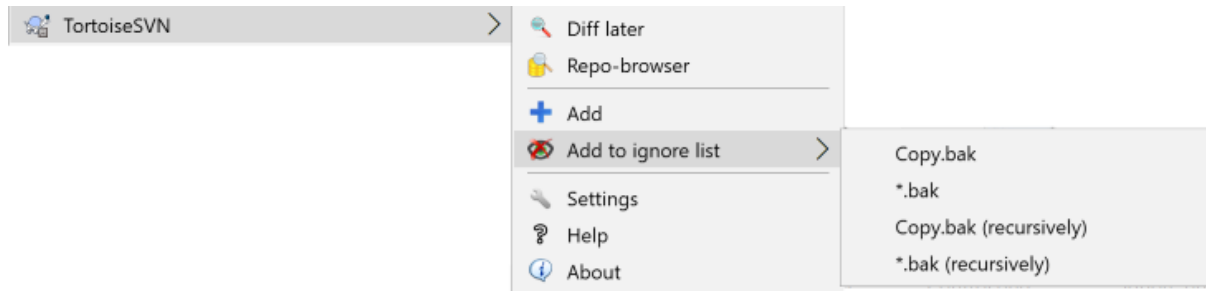
Similar to ExamDiff Pro, this is an excellent shareware diff tool which can handle directory diffs and unicode. Download it from *Scooter Software* [<https://www.scootersoftware.com/>].

### Araxis Merge

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

Consultar [Secção 4.31.5, “Preferências de Programas Externos”](#) para mais informações em como configurar o TortoiseSVN para usar essas ferramentas.

## 4.12. Adicionar Novos Ficheiros e Pastas



**Figura 4.31. 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 no interior 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.

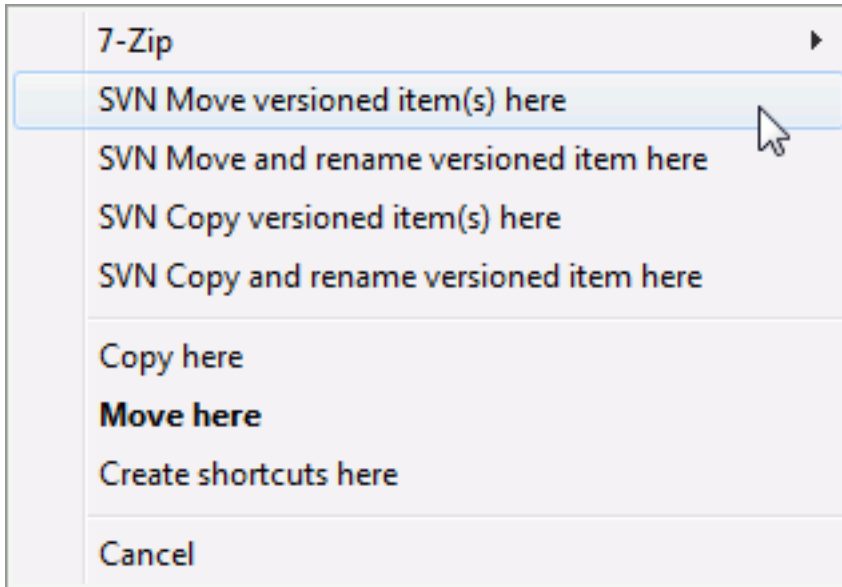
Podes também adicionar ficheiros de dentro da cópia de trabalho, simplesmente arrastando com o botão direito e largando-os na caixa de diálogo submeter.

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

## 4.13. Copiando/Movendo/Renomeando Ficheiros e Pastas

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

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 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 4.32. Menu arrastar com o botão direito para uma pasta sob controlo de versões.**

Agora, podes copiar o conteúdo versionado existente 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 cópias de trabalho, usando o familiar método cortar-e-colar. Usar o Copiar ou Colar padrão do Windows, para copiar um ou mais itens versionados para a área de transferência (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 4.20.1, “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 4.10.3, “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 4.25, “O Navegador de Repositório”](#) para mais detalhes.



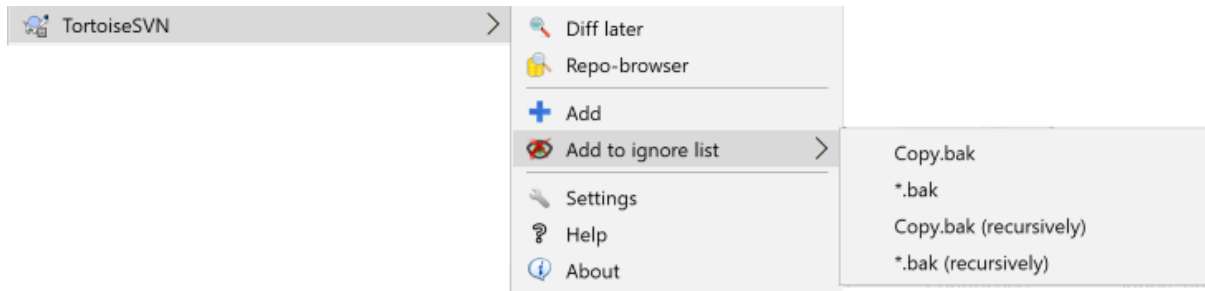
### **Não consigo copiar entre repositórios**

Embora possas copiar ou mover ficheiros e pastas *dentro* de um repositório, tu *não podes* copiar ou mover de um repositório para outro, enquanto preservas o histórico, usando o TortoiseSVN. Nem mesmos se os repositórios residem no mesmo servidor. Tudo o que podes fazer é copiar o conteúdo no seu estado corrente e adicioná-lo como novo conteúdo no segundo repositório.

Se tens dúvidas 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.



## 4.14. Ignorando Ficheiros E Pastas



**Figura 4.33. 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.

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

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

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 4.18, “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 4.31.1, “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”.

#### 4.14.1. 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 bruta ao emparelhar caracteres, e.g. para ignorar \*.tmp, independentemente do tamanho de 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](http://www.opengroup.org/onlinepubs/009695399/utilities/xcu_chap02.html#tag_02_13)].

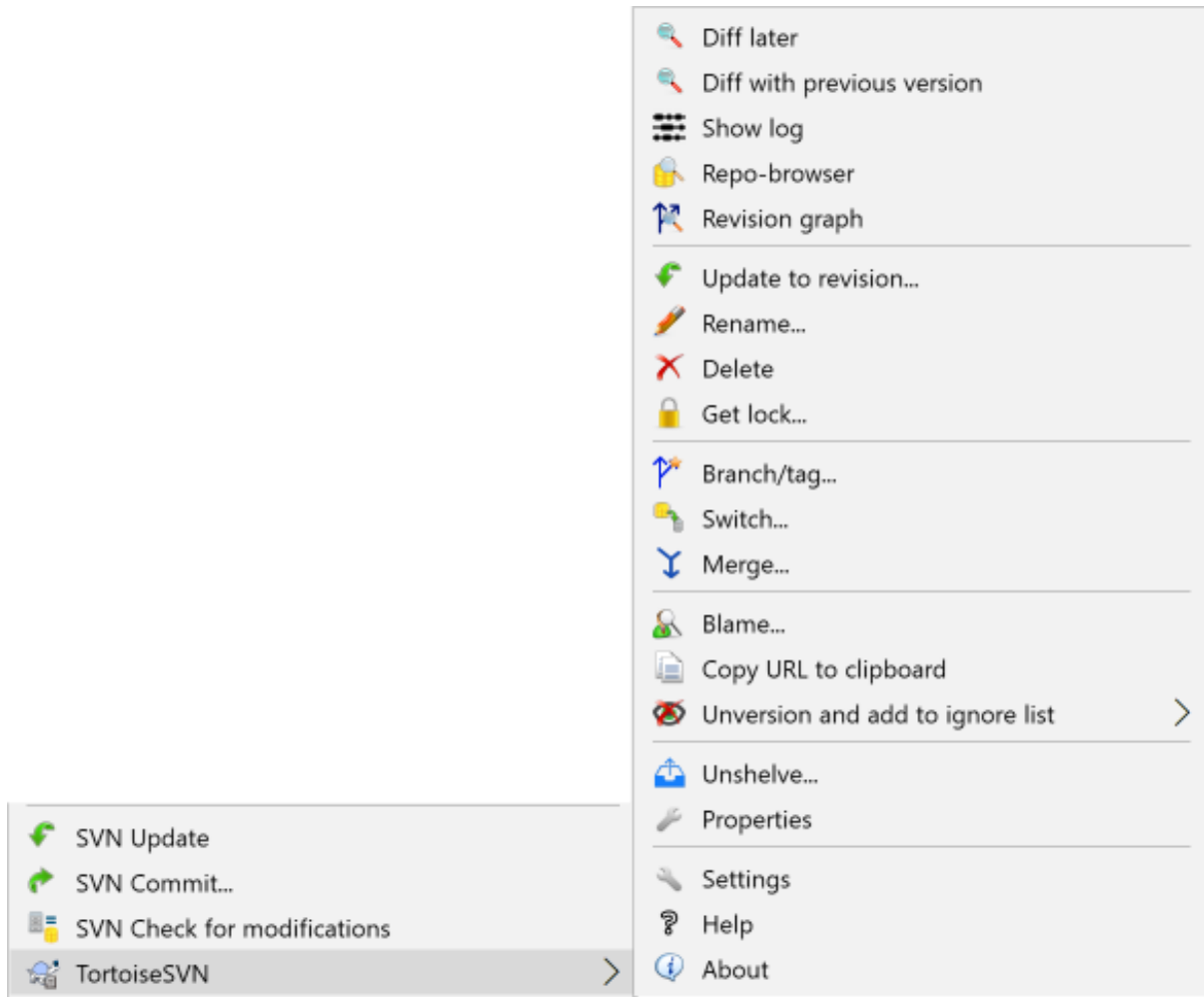


### **Não Usar Caminhos na Lista Global de Ignorados**

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.

## **4.15. Removendo, Movendo e Renomeando**

O Subversion permite renomear e mover ficheiro e pastas. Portanto, existem entradas para apagar e renomear no sub-menu do TortoiseSVN.



**Figura 4.34. Menu de contexto do Explorador para ficheiros versionados**

### 4.15.1. Removendo ficheiros e pastas

Para remover ficheiros ou pastas do Subversion usa TortoiseSVN → Remove.

Quando TortoiseSVNRemove

Se queres remover um tem do repositório, mas mantê-lo localmente como um ficheiro/pasta não versionado, usa Menu de Contexto Estendido → Remove (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.

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



#### **Recuperando um ficheiro ou pasta removido**

Se removeste um ficheiro ou pasta e já submeteste essa operação de remoção para o repositório, então o normal TortoiseSVN → Reverter , não o pode mais recuperar. Mas o ficheiro ou pasta não

está ainda totalmente perdido. Se conheceres 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 à direita e selecciona **Menu de Contexto** → **Copiar para...**, como destino dessa operação de cópia, selecciona o caminho para a tua cópia de trabalho.

#### 4.15.2. 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 pretendes 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 no interior 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.

Outra maneira de mover ou copiar ficheiros, é usar os comandos de copiar/cortar do Windows. Selecciona os ficheiros que queres copiar, clica com o botão direito e escolhe a partir do menu de contexto do explorador **Menu de Contexto** → **Copiar**. Então navega para a pasta destino, clica com o botão direito e escolhe **TortoiseSVN** → **Colar**. Para mover ficheiros, escolhe **Menu de Contexto** → **Cortar** em vez de **Menu de Contexto** → **Copiar**.

Podes também usar o navegador de repositório para mover itens no mesmo. Consultar [Secção 4.25, “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.

#### 4.15.3. 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 maiúsculas 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 maiúsculas 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

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

#### 4.15.4. Reparando Renomeações de Ficheiros

Por vezes o teu IDE amigável irá renomear ficheiros por ti como parte de um processo de refacturação, 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.

#### 4.15.5. Removendo Ficheiros Não Versionados

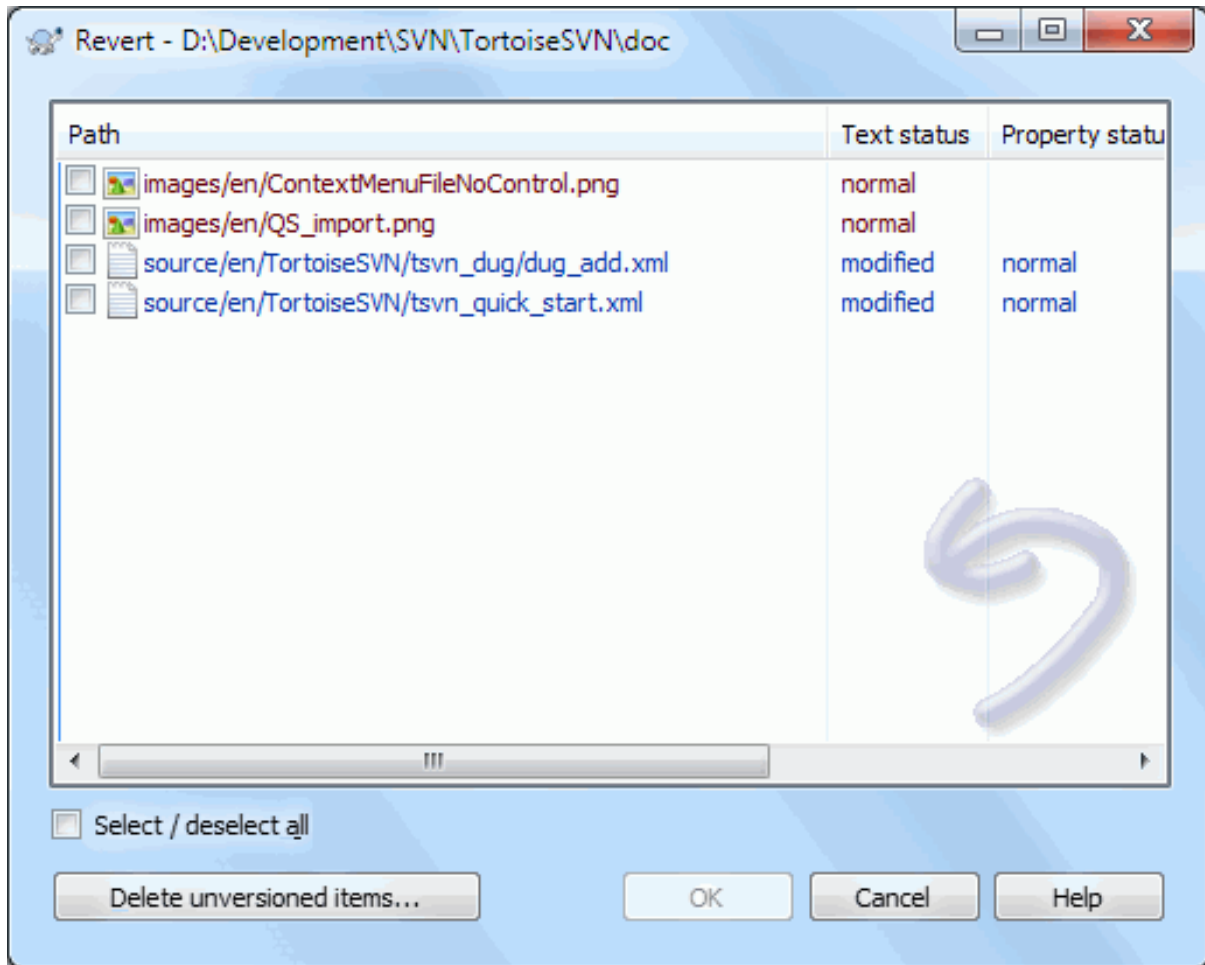
Normalmente tu configuras a tua lista de ignorados tal que todos os ficheiros gerados sejam ignorados pelo Subversion. Mas se queres limpar todos esse itens ignorados de modo a produzir uma versão limpa? Por norma tu inserias isso na tua makefile, mas se quiseres depurar a tua makefile ou alterar o sistema 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.

#### 4.16. Desfazer Alterações

Se pretenderes 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 4.35. Caixa de Diálogo Reverter**

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

Se queres desfazer uma remoção ou renomeação, necessitas de usar o Reverter na pasta pai, já que o item removido não existirá para poderes clicar à direita sobre ele.

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.

Visto que o reverter é por vezes usado para limpar uma cópia de trabalho, existe um botão extra que te permite remover também itens não versionados. Quando tu clicas neste botão, outra caixa de diálogo aparece com uma lista de todos os itens não versionados que tu podes seleccionar para remoção.



### 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 4.10, “Caixa de Diálogo Registo de Revisões”](#) para mais informação.

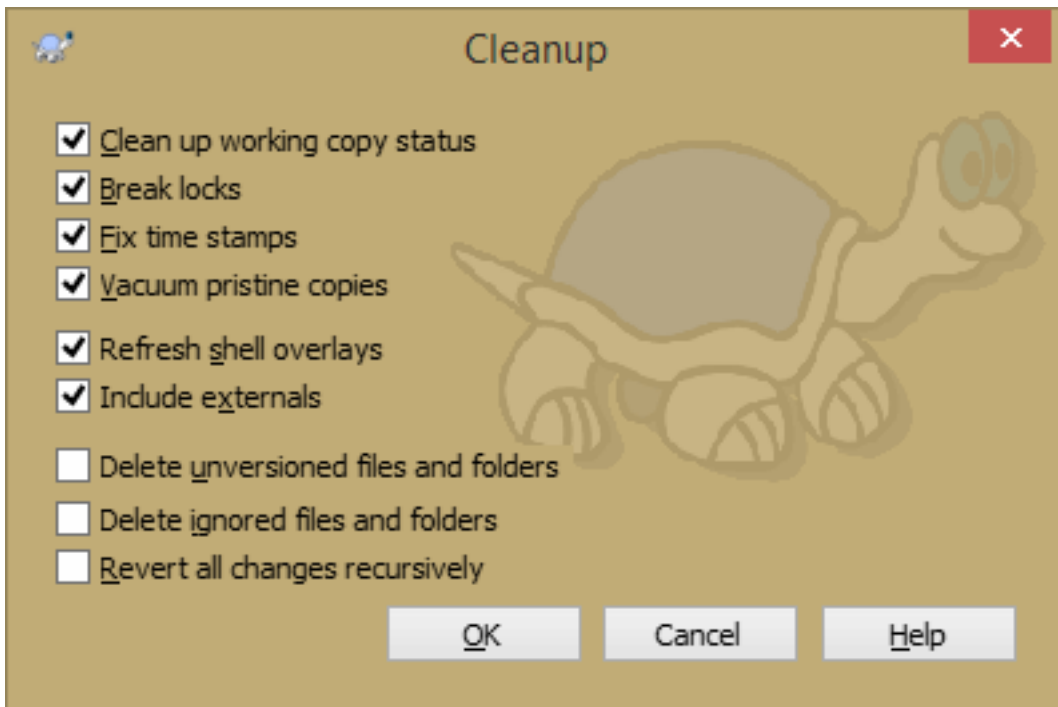


## Reverter é Lento

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.

## 4.17. Limpar

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



**Figura 4.36. The Cleanup dialog**

Na caixa de diálogo de limpeza, também existem outras opções úteis para obter uma cópia a funcionar no estado clean.

### Limpar o estado da cópia de trabalho

Como constatado acima, esta opção tenta colocar uma cópia de trabalho inconsistente num estado funcional e usável. Isto não afecta quaisquer dados que tenhas, mas apenas os estados internos da base de dados da cópia de trabalho. Este é actualmente o comando `Limpar` que conheces de antigos clientes TortoiseSVN ou de outros clientes SVN.

### Break write locks

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

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

**Fix time stamps**

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

**Vacuum pristine copies**

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

**Refrescar sobreposições de shell**

Às vezes, as sobreposições "shell", especialmente na visualização da árvore no lado esquerdo do explorador não mostram o estado actual, ou o estado da cache falhada para re-organizar as alterações. Nesta situação, podes usar este comando para forçar uma atualização.

**Incluir externos**

Se esta opção estiver assinalada, então todas as acções serão feitas para todos os arquivos e pastas incluídos com a propriedade `svn:externals` também.

**Apagar ficheiros e pastas não versionadas, Apagar ficheiros e pastas ignoradas**

Esta é uma maneira rápida e fácil de remover todos os ficheiros gerados na tua cópia em funcionamento. Todos os ficheiros e pastas que não são versionados, são movidos para a reciclagem.

Nota: tu também podes fazer o mesmo a partir do do diálogo no TortoiseSVN → Reverter. Lá também terá uma lista de todos os ficheiros e pastas não versionadas para seleccionar para a sua eliminação.

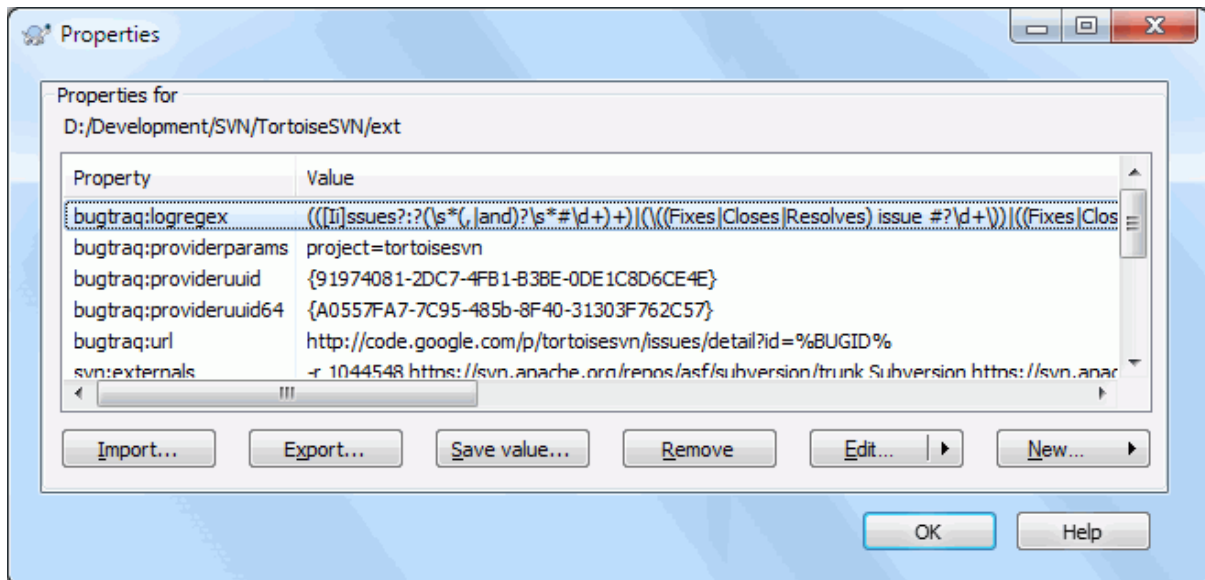
**Reverter recursivamente todas as alterações**

Este comando reverte todas as tuas modificações locais que ainda não estão submetidas.

Nota: é melhor usar o TortoiseSVN → Reverter comando em vez disto, porque lá podes primeiro ver e seleccionar os ficheiros que você deseja reverter.

## 4.18. Configurações de Projecto

### 4.18.1. Propriedades Subversion



**Figura 4.37. 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 4.19, “Itens Externos”](#).

#### 4.18.1.1. `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.

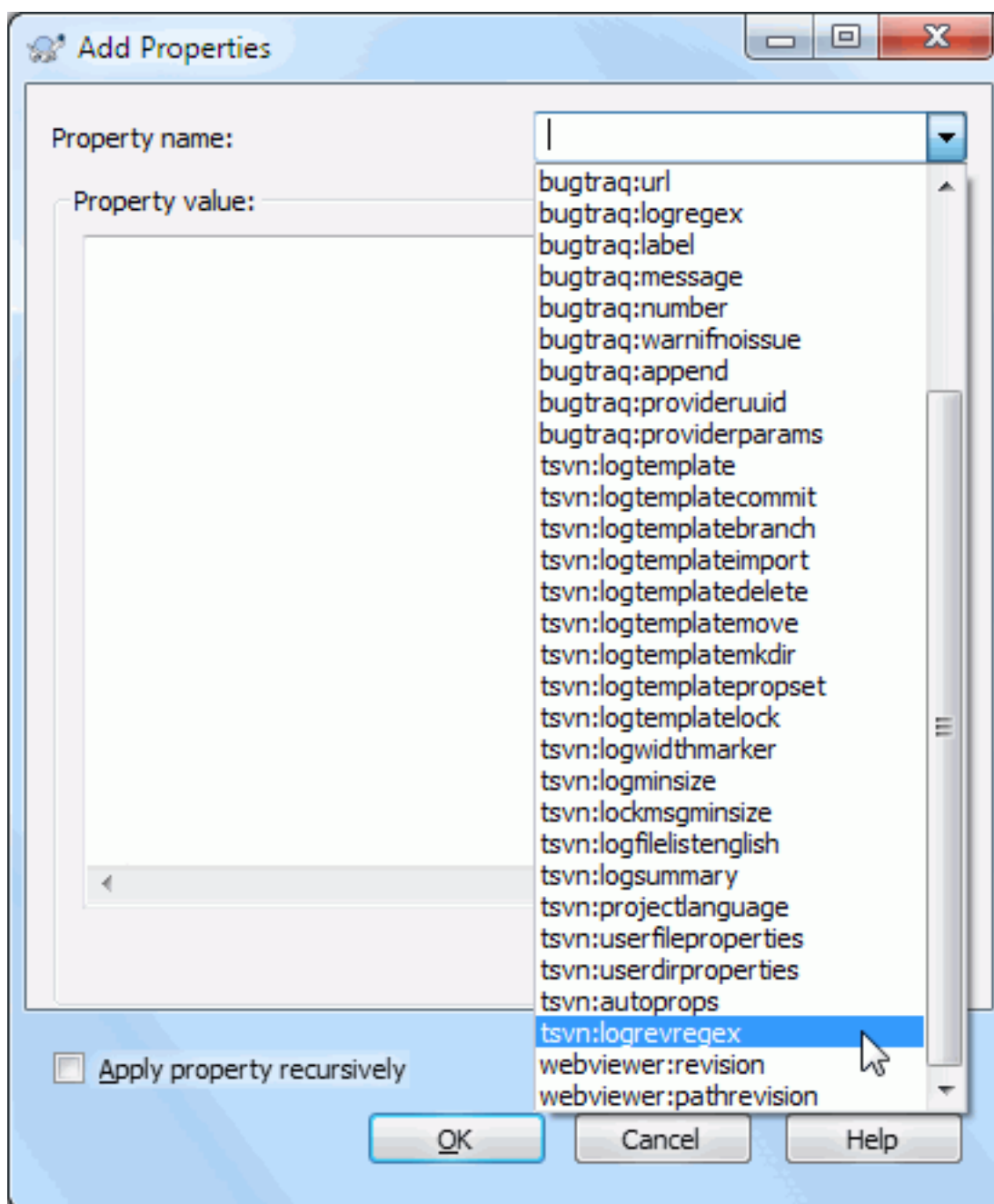
`$Id$`

A combinação comprimida das quatro palavras-chave prévias.

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

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

#### 4.18.1.2. Adicionando e Editando Propriedades



**Figura 4.38. Adicionando propriedades**

Para adicionar uma nova propriedade, clica primeiro em Nova.... Selecciona o nome da propriedade pretendida a partir do menu, e preenche então a informação pretendida na caixa de diálogo específica da propriedade. Essas caixas de diálogo específicas das propriedades são descritas em maior detalhe em [Secção 4.18.3, “Editores de Propriedades”](#).

Para adicionar uma propriedade que não tem a sua própria caixa de diálogo, selecciona Avançado a partir do menu Novo....

Se queres aplicar uma propriedade a vários itens de uma só vez, selecciona os ficheiros/pastas no explorador e então 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**.

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 4.19, “Itens Externos”](#).



### Edita propriedades na revisão HEAD

Porque as propriedades são versionadas, não poderás editar as propriedades das versões anteriores. Se olhares para as propriedades a partir da caixa de diálogo de registo, ou a partir de uma revisão não-HEAD no navegador de repositório, irás ver uma lista de propriedades e valores, mas sem controlos de edição.

#### 4.18.1.3. 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.

#### 4.18.1.4. 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.

#### 4.18.1.5. 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.

You can edit the Subversion configuration file to enable this feature on your client. The **General** page of TortoiseSVN's settings dialog has an edit button to take you there directly. The config file is a simple text file which controls some of Subversion's workings. You need to change two things: firstly in the section headed

`miscellany` uncomment the line `enable-auto-props = yes`. Secondly you need to edit the section below to define which properties you want added to which file types. This method is a standard Subversion feature and works with any Subversion client. However it has to be defined on each client individually - there is no way to propagate these settings from the repository.

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.

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

Qualquer que seja o método que escolhas, deverás ter em atenção que o auto-props só é aplicado a ficheiros, na altura em que forem 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.

## 4.18.2. 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.

Existem também templates específicos de acções, que podes usar em vez de `tsvn:logtemplate`. Os templates específicos de acções são usados se configurados, mas o `tsvn:logtemplate` será usado se não for configurado nenhum template específico de acção.

Os templates específicos de acção são:

- `tsvn:logtemplatecommit` é usado para todas as submissões a partir de uma cópia de trabalho.
  - `tsvn:logtemplatebranch` é usado quando crias um ramo/etiqueta, ou quando copias ficheiros ou pastas directamente do navegador de repositório.
  - `tsvn:logtemplateimport` é usado para importações.
  - `tsvn:logtemplatedelete` é usado ao remover itens directamente do navegador de repositório.
  - `tsvn:logtemplatemove` é usado quando renomeias ou moves itens no navegador de repositório.
  - `tsvn:logtemplatemkdir` é usado quando crias pastas no navegador de repositório.
  - `tsvn:logtemplatepropset` é usado quando modificas propriedades no navegador de repositório.
  - `tsvn:logtemplatelock` é usado quando obténs um bloqueio.
- Subversion allows you to set “autoprops” which will be applied to newly added or imported files, based on the file extension. This depends on every client having set appropriate autoprops in their Subversion configuration file. `tsvn:autoprops` can be set on folders and these will be merged with the user's local autoprops when importing or adding files. The format is the same as for Subversion autoprops, e.g. `*.sh = svn:eol-style=native;svn:executable` sets two properties on files with the `.sh` extension.

Se houver um conflito entre autoprops locais e `tsvn:autoprops`, as configurações de projecto tomam precedência porque são específicas a esse projecto.

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

- 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`.
- TortoiseSVN can use a spell checker. On Windows 10, the spell checker of the OS is used. On earlier Windows versions, it can use spell checker modules which are also used by OpenOffice and Mozilla. If you have those installed this property will determine which spell checker to use, i.e. in which language the log messages for your project should be written. `tsvn:projectlanguage` sets the language module the spell checking engine should use when you enter a log message. You can find the values for your language on this page: [MSDN: Language Identifiers](https://docs.microsoft.com/en-us/windows/desktop/intl/language-identifier-constants-and-strings) [https://docs.microsoft.com/en-us/windows/desktop/intl/language-identifier-constants-and-strings].

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.

- A propriedade `tsvn:logrevregex` define uma expressão regular que corresponde a referências para revisões, na mensagem de registo. Isto é, usado na caixa de diálogo registo para converter tais referências em hiperligações que quando clicadas, ou irão ou seleccionar essa mesma revisão (se a revisão já for mostrada na caixa de diálogo registo, ou se está disponível a partir da cache de registo) ou abrir uma nova caixa de diálogo de registo mostrando essa revisão.

A expressão regular tem de corresponder à referência completa, e não apenas ao número de revisão. O número de revisão é extraído automaticamente da string de referência que lhe é correspondida.

Se esta propriedade não for configurada, será usada uma expressão regular por defeito para ligar referências de revisão.

- Existe disponíveis várias propriedades para configurar scripts de gancho de cliente. Cada propriedade corresponde a um tipo específico de script de gancho.

As propriedades/scripts-gancho disponíveis são

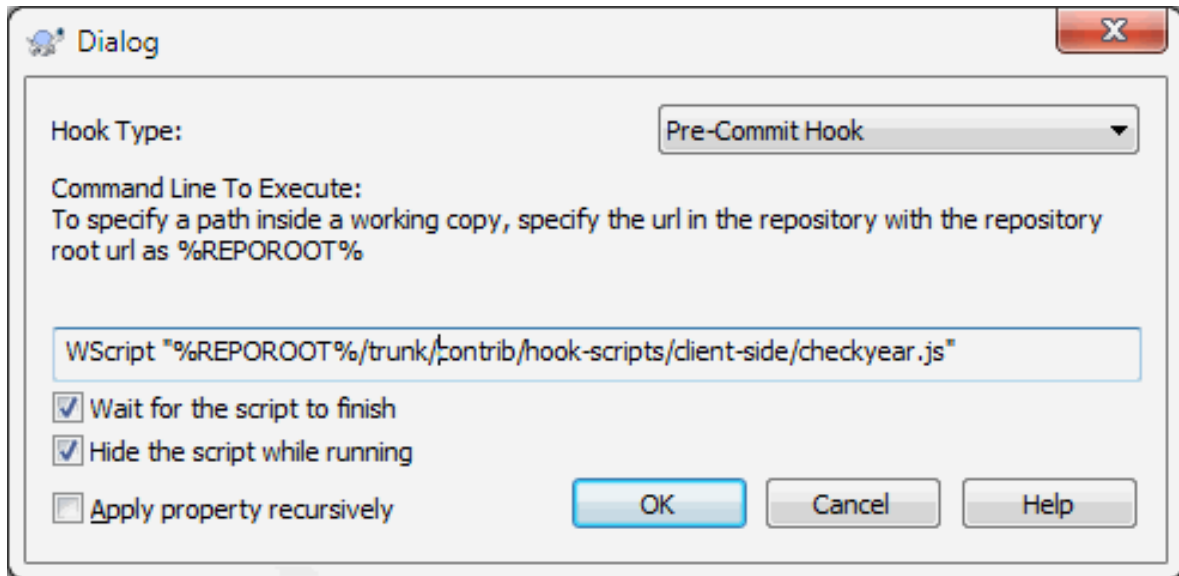
- `tsvn:startcommithook`
- `tsvn:precommithook`
- `tsvn:postcommithook`
- `tsvn:startupdatehook`
- `tsvn:preupdatehook`
- `tsvn:postupdatehook`
- `tsvn:prelockhook`
- `tsvn:postlockhook`

Os parâmetros são os mesmos tal com se tu configurasses os scripts-gancho na caixa de diálogo preferências. Ver [Secção 4.31.8, “Scripts de Gancho do Lado do Cliente”](#) para mais detalhes.

Visto que nem todos os utilizadores têm a sua cópia de trabalho colocada na mesma localização, nem com o mesmo nome, podes então configurar um script/ferramenta a executar que resida na tua cópia de trabalho especificando o URL do repositório em alternativa, usando `%REPOROOT%` como a parte do URL referente à raiz do repositório. Por exemplo, se o script de gancho está na tua cópia de trabalho sob `contrib/hook-scripts/client-side/checkyear.js`, irás especificar o caminho para o script como `%REPOROOT%/trunk/contrib/hook-scripts/client-side/checkyear.js`. Deste modo mesmo que movas o teu repositório para outro servidor, não terás de ajustar as propriedades dos scripts de gancho.

Instead of `%REPOROOT%` you can also specify `%REPOROOT+%`. The `+` is used to insert any number of folder paths necessary to find the script. This is useful if you want to specify your script so that if you create a branch the script is still found even though the url of the working copy is now different. Using the example above, you would specify the path to the script as `%REPOROOT+%/contrib/hook-scripts/client-side/checkyear.js`.

A próxima captura de ecrã mostra como é configurado o script para verificar os anos de copyright nos cabeçalhos dos ficheiros fonte no TortoiseSVN.



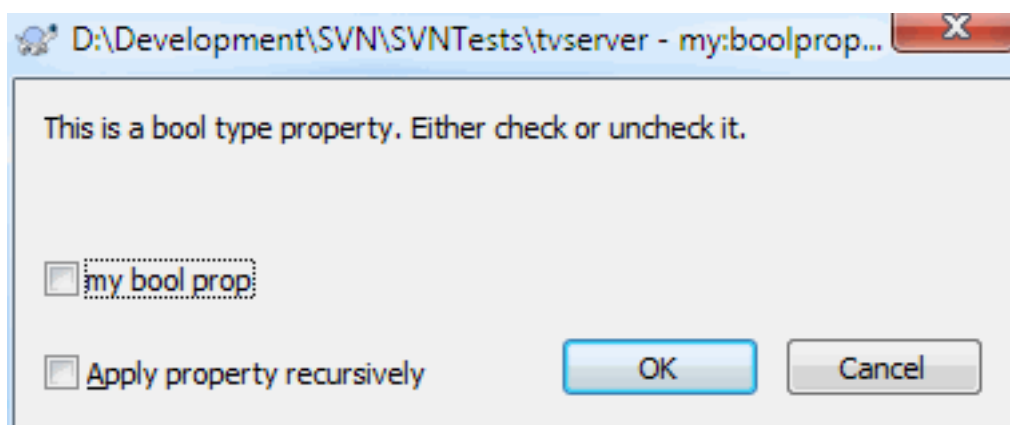
**Figura 4.39. Caixa de diálogo de propriedades para scripts gancho**

- 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 personalizadas, 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 personalizadas 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 personalizadas irão aparecer na lista de nomes de propriedades pré-definidas.

You can also specify whether a custom dialog is used to add/edit your property. TortoiseSVN offers four different dialog, depending on the type of your property.

bool

Se a tua propriedade só pode ter dois estados, e.g., true ou false, podes então configurar a tua propriedade como do tipo `bool`.



**Figura 4.40. Caixa de diálogo da propriedade tipos de utilizador booleano**

. Especifica a tua propriedade deste modo:

```
propertyname=bool ; labeltext ( YESVALUE ; NOVALUE ; Checkboxtext )
```

Na

```
<oliteral>labeltext eststateSe a tua propriedade representar um de muitos estados, e.g., sim,n, podes entstate Caixa de di do seguinte modo:propertyname=state;labeltext(DEFVAL;VAL1;TEXT1;VAL2;TEXT2;VAL3;TEXT3;...)Os parool, com DEFVAL como parPara atsinglelinePara as propriedades que consistem numa linha: Caixa dipropertyname=singleline;labeltext(regex)O regex indica a expressmultiline propriedades que consistem em mmulti-linha: Caixa dipropertyname=multiline;labeltext(regex)O regex indica a expressThe screenshots above were made with the following tsvn:userdirproperties: my:boolprop=bool;This is a bool type property. Either check or uncheck it.(true;false;my bool prop) my:stateprop1=state;This is a state property. Select one of the two states.(true;true;true value;false;false value) my:stateprop2=state;This is a state property. Select one of the three states.(maybe;true;answer is correct;false;answer is wrong;maybe;not answered) my:stateprop3=state;Specify the day to set this property.(1;1;Monday;2;Tuesday;3;Wednesday;4;Thursday;5;Friday;6;Saturday;7;Sunday) my:singlelineprop=singleline;enter a small comment.(*) my:multilineprop=multiline;copy and paste a full chapter here.(*) </oliteral>
```

O TortoiseSVN pode-se integrar com algumas ferramentas de controlo de problemas. Isto usa as propriedades de projecto que começam com `bugtraq:`. Consulta [Secção 4.29, “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 4.30, “Integração com visualizadores de repositório Web-based”](#) para mais informação.



## Fixar propriedades de projecto em pastas

Essas propriedades especiais de projecto devem ser fixadas em *pastas*, para o sistema funcionar. Quando executas um comando do TortoiseSVN que usa essas propriedades, elas são lidas a partir da pasta em que clicaste. Se as propriedades não são encontradas aí, o TortoiseSVN irá procurar em 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 efectua checkout só a partir do, 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. Se colocares a mesma propriedade, mas usares diferentes valores em diferentes niveis da hierarquia do projecto, irás então obter resultados diferentes dependendo do sítio na estrutura de pastas em que clicas.

Só para propriedades de projecto, i.e. `tsvn:`, `bugtraq:` e `webviewer:` podes usar a caixa de verificação **Recursiva** para fixar a propriedade em todas as subpastas na hierarquia, sem também fixa-la em todos os ficheiros.

Quando adicionas novas subpastas a uma cópia de trabalho, usando o TortoiseSVN, quaisquer propriedades de projecto presentes na pasta pai serão automaticamente adicionadas à nova pasta filho também.



## Limitações no Uso do Navegador de Repositório

Obter propriedades remotamente é uma operação lenta, pelo que algumas das funcionalidades descritas acima não funcionarão no navegador de repositório, como funcionam na cópia de trabalho.

- Quando adicionas uma propriedade usando o navegador de repositório, só as propriedades padrão `svn:` são disponibilizadas na lista pré-definida. Qualquer outro nome de propriedade deverá ser introduzido manualmente.
- As propriedades não podem ser configuradas ou removidas recursivamente, usando o navegador de repositório.
- As propriedades de projecto *não* serão automaticamente propagadas quando uma pasta filho for adicionada com uso do navegador de repositório.



- O `tsvn:autoprops` *não* irá configurar as propriedades nos ficheiros que são adicionados usando o navegador de repositório.



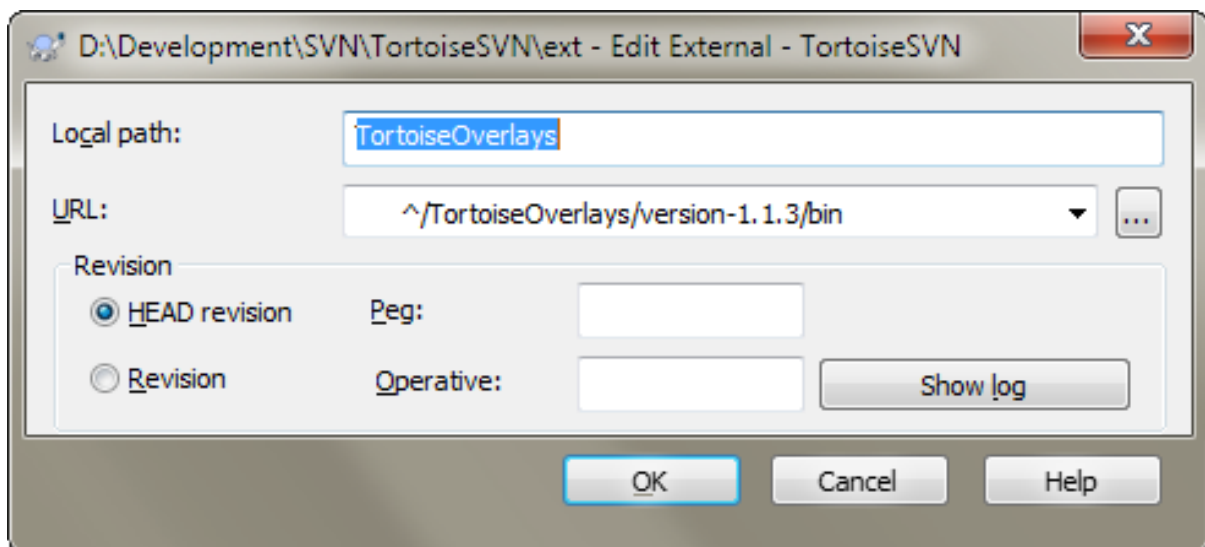
## Cuidado

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.

### 4.18.3. Editores de Propriedades

Algumas propriedades têm de usar valores específicos, ou estarem formatadas de uma determinada maneira de modo a serem usadas para automação. Para ajudar a obter a formatação correcta, o TortoiseSVN apresenta caixas de edição para certas propriedades em particular que mostram os valores possíveis, ou quebram a propriedade em componentes individuais.

#### 4.18.3.1. Conteúdo Externo



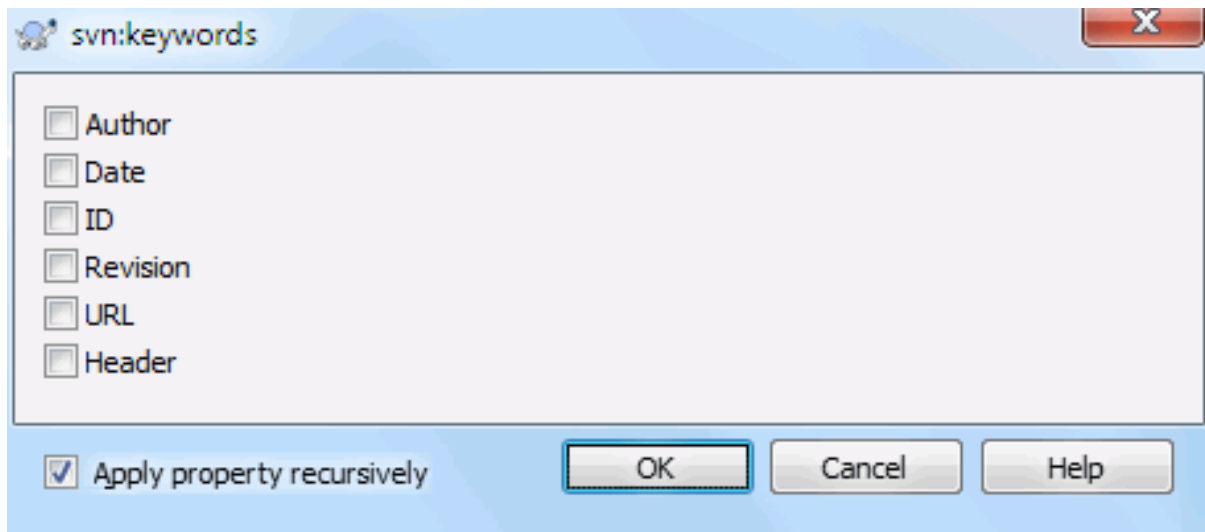
**Figura 4.41. Página de propriedade `svn:externals`**

A propriedade `svn:externals` pode ser usada para introduzir outros projectos do mesmo repositório ou de um repositório totalmente diferente, como descrito em [Secção 4.19, “Itens Externos”](#).

You need to define the name of the sub-folder that the external folder is checked out as, and the Subversion URL of the external item. You can check out an external at its HEAD revision, so when the external item changes in the repository, your working copy will receive those changes on update. However, if you want the external to reference a particular stable point then you can specify the specific revision to use. IN this case you may also want to specify the same revision as a peg revision. If the external item is renamed at some point in the future then Subversion will not be able to update this item in your working copy. By specifying a peg revision you tell Subversion to look for an item that had that name at the peg revision rather than at HEAD.

The button Find HEAD-Revision fetches the HEAD revision of every external URL and shows that HEAD revision in the rightmost column. After the HEAD revision is known, a simple right click on an external gives you the command to peg the selected externals to their explicit HEAD revision. In case the HEAD revision is not known yet, the right click command will fetch the HEAD revision first.

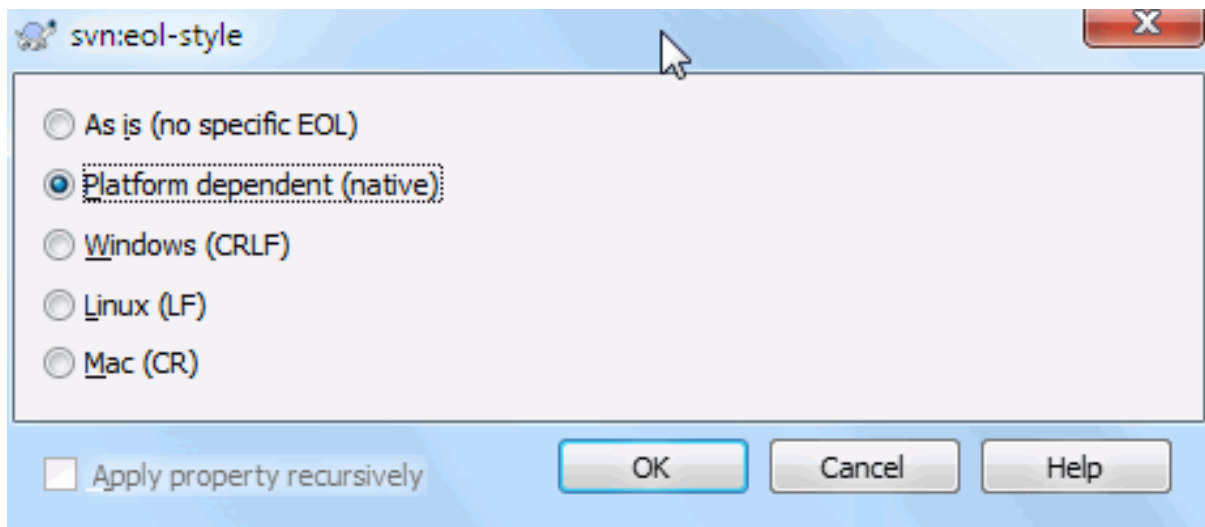
#### 4.18.3.2. Palavras-chave SVN



**Figura 4.42. Página de propriedade svn:keywords**

Selecciona as palavras-chave que gostarias de ver expandidas no teu ficheiro.

#### 4.18.3.3. Estilo EOL



**Figura 4.43. Página de propriedade svn:eol-style**

Selecciona o estilo de fim-de-linha que desejas usar e o TortoiseSVN irá utilizar o valor correcto da propriedade.

#### 4.18.3.4. Integração do Controlador de Problemas

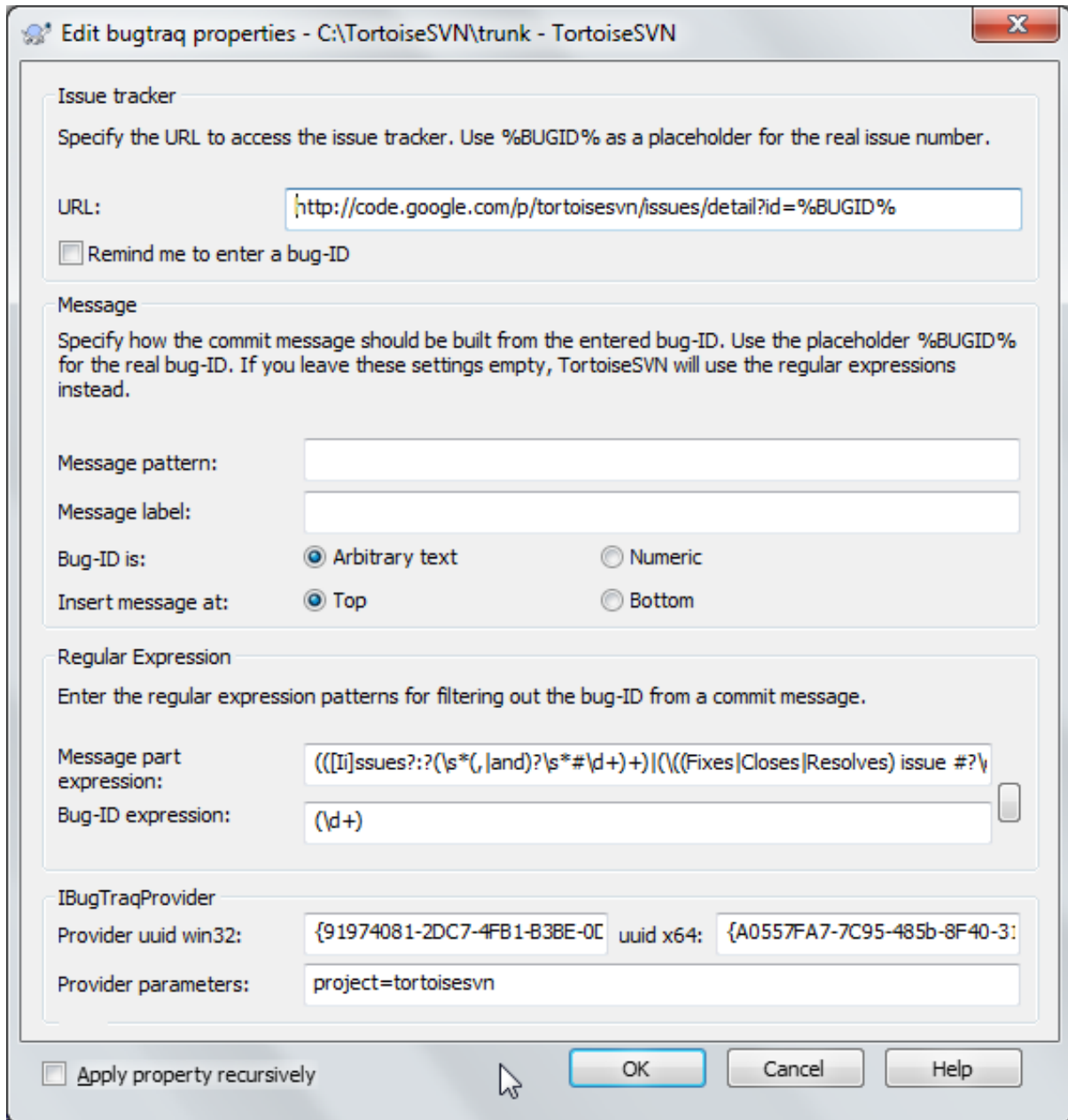
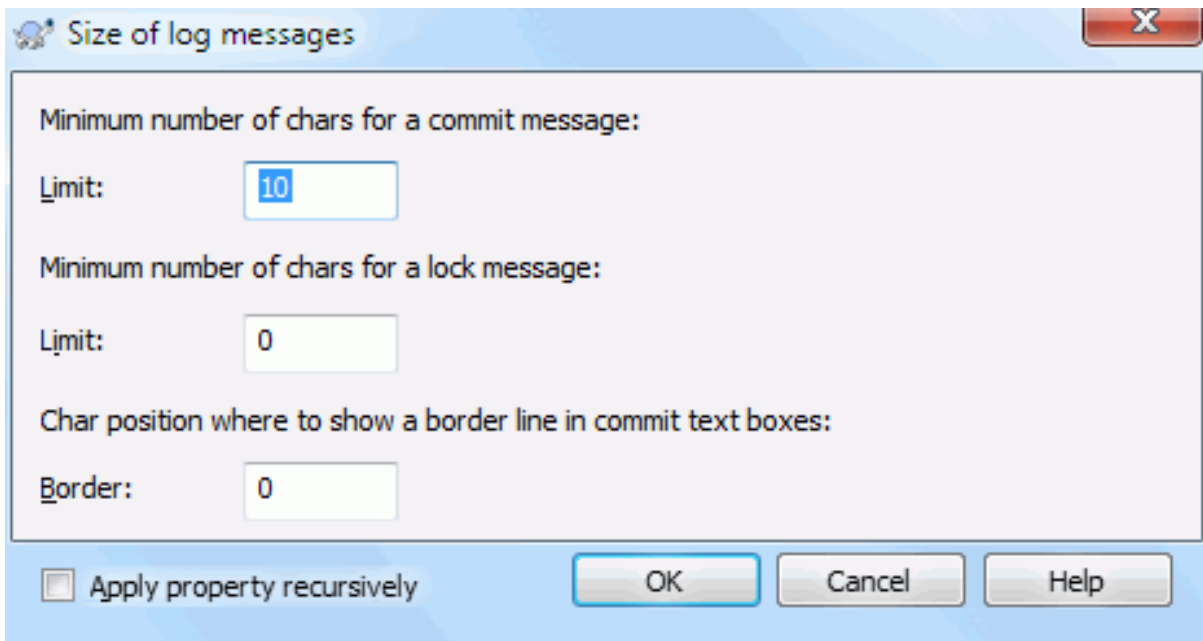


Figura 4.44. Página de propriedade tsvn:bugtraq

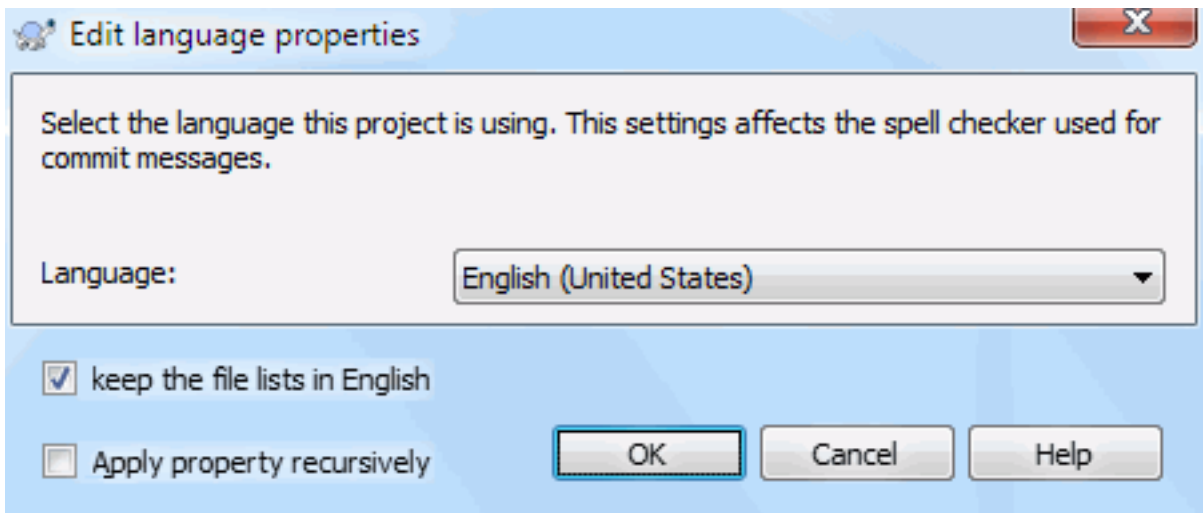
#### 4.18.3.5. Tamanhos das mensagens de registo



**Figura 4.45. Página de propriedade do tamanho das mensagens de registo**

These 3 properties control the formatting of log messages. The first 2 disable the OK in the commit or lock dialogs until the message meets the minimum length. The border position shows a marker at the given column width as a guide for projects which have width limits on their log messages. Setting a value to zero will delete the property.

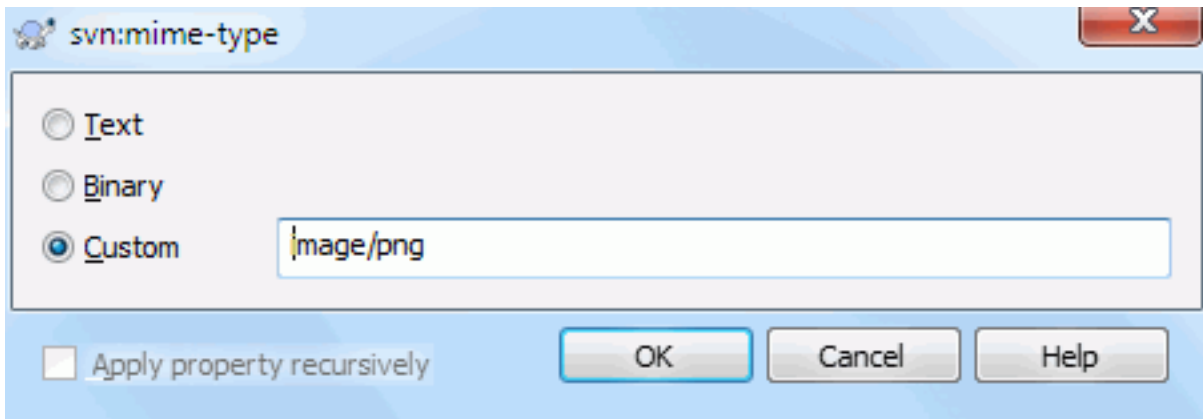
#### 4.18.3.6. Língua do Projecto



**Figura 4.46. Página de propriedade da Língua**

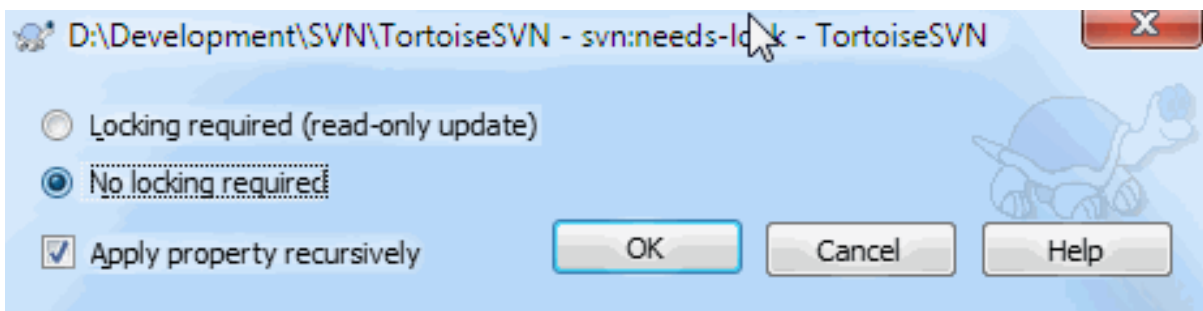
Escolhe a linguagem a usar para verificar ortografia nas mensagens de registona caixa de diálogo submeter. A caixa de verificação de lista de ficheiros têm utilidade quando clicas à direita no painel de mensagem e seleccionas Colar lista de ficheiros. Por defeito o estado do Subversion será mostrado na lingua local. Quando esta caixa está sinalizada o estado é sempre dado em Inglês, para projectos que requerem mensagens de registo apenas em Inglês.

#### 4.18.3.7. MIME-type



**Figura 4.47. Página de propriedade svn:mime-type**

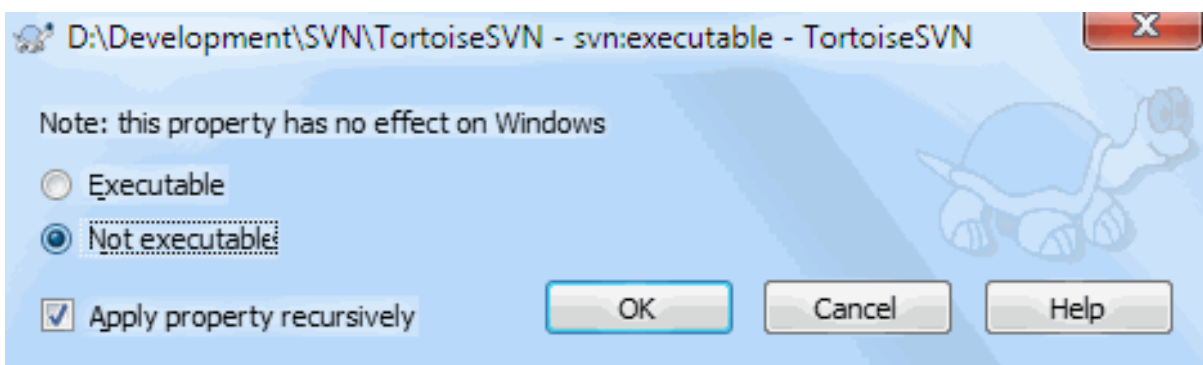
#### 4.18.3.8. svn:needs-lock



**Figura 4.48. Página de propriedade svn:needs-lock**

Esta propriedade simplesmente controla quando um ficheiro será obtido (checkout) como apenas de leitura, se não existir nenhum bloqueio detido para ele na cópia de trabalho.

#### 4.18.3.9. svn:executable



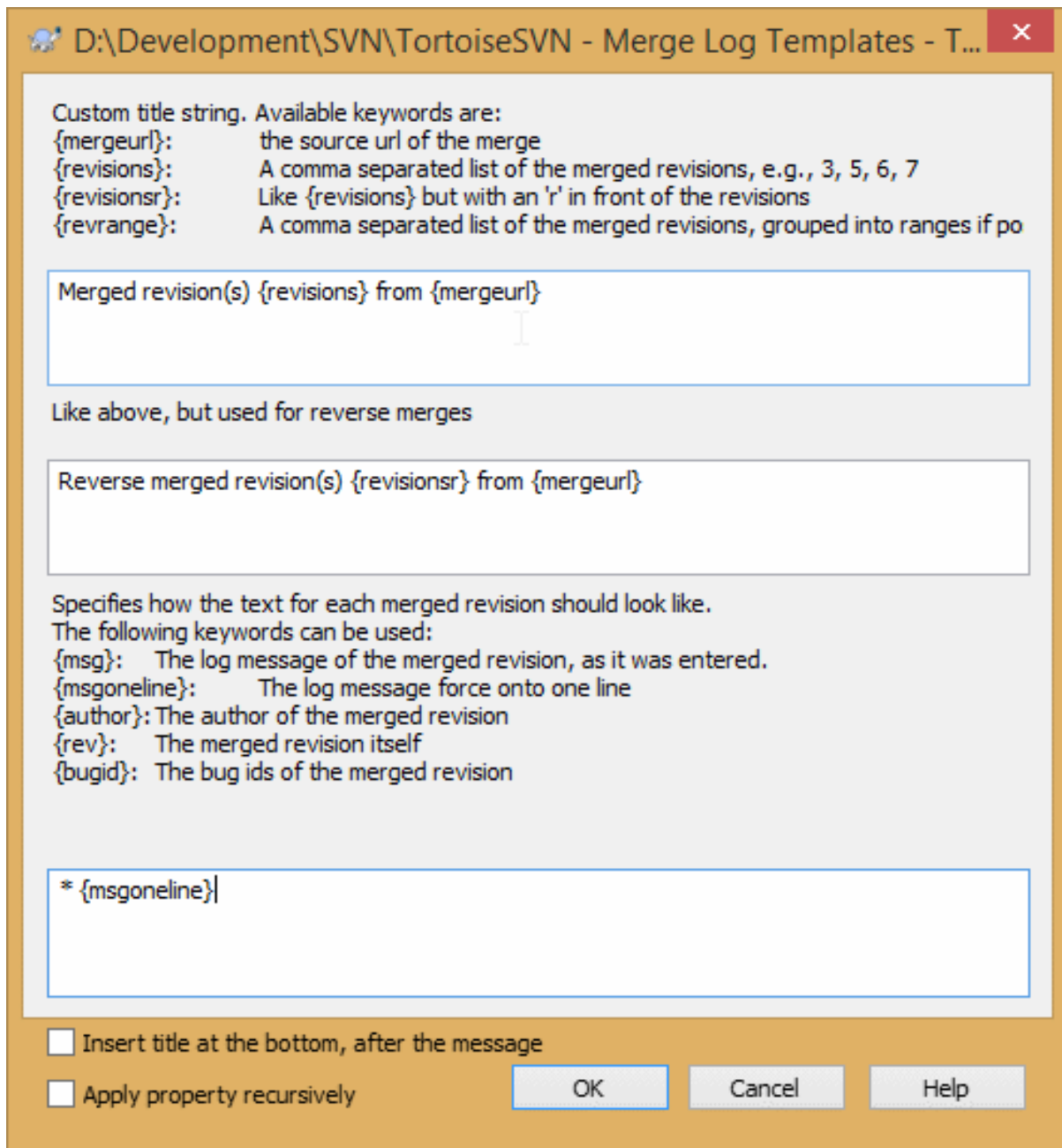
**Figura 4.49. Página de propriedade svn:executable**

Esta propriedade controla se será dado o estado de executável a um ficheiro, quando for obtido (checkout) num sistema Unix/Linux. Não tem efeito num checkout em Windows.

#### 4.18.3.10. Templates de mensagens de registo

Quando são integradas as revisões na cópia de trabalho, o TortoiseSVN gera uma mensagem de registo para todas as revisões integradas. Essas ficam então disponíveis a partir do botão **Mensagens recentes** na janela de diálogo submeter.

Podes personalizar essa mensagem gerada com as seguintes propriedades:



**Figura 4.50. Caixa de propriedades integra templates de mensagens de registo**

tsvn:mergelogtemplatetitle, tsvn:mergelogtemplaterersetitle

Esta propriedade especifica a primeira parte da mensagem de registo gerada. As próximas palavras-chave podem ser usadas:

{revisions}

Lista de revisões integradas, separadas por virgulas, e.g., 3, 5, 6, 7

{revisionsr}

Como {revisions}, mas com cada revisão precedida com um r, e.g., r3, r5, r6, r7

{revrange}

Lista de revisões integradas separadas por virgulas, se possível agrupadas em intervalos, e.g., 3, 5-7

{mergeurl}

O URL fonte da integração, i.e., de onde são integradas as revisões.

O valor por defeito para esta string é `Merged revision(s) {revrange} from {mergeurl}`: com uma linha nova no fim.

tsvn:mergelogtemplatemsg

Esta propriedade especifica como o texto em cada revisão integrada deve ficar formatado. Podem ser usadas as seguintes palavras chave:

{msg}

A mensagem de registo da revisão integrada, como foi introduzida.

{msgonline}

Como o {msg}, mas todas as linhas novas são substituídas por um espaço, para que toda a mensagem de registo apareça numa única linha.

{author}

O autor da revisão integrada.

{rev}

A própria revisão integrada.

{bugids}

Os IDs dos problemas da revisão integrada, se existirem.

tsvn:mergelogtemplatemsgtitlebottom

This property specifies the position of the title string specified with the `tsvn:mergelogtemplatetitle` or `tsvn:mergelogtemplatereversetitle`. If the property is set to `yes` or `true`, then the title string is appended at the bottom instead of the top.



### Importante

Isto só funciona se as revisões integradas já estão na cache de registo. Se desactivaste a cache de registo ou não indicaste o registo antes do merge, a mensagem gerada não irá então conter qualquer informação sobre as revisões integradas.

## 4.19. 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.

### 4.19.1. Pastas Externas

Digamos que efectuas o checkout da cópia de trabalho do `/project1` para `D:\dev\project1`. Selecciona a pasta `D:\dev\project1`, clica à direita e a partir do menu de contexto selecciona **Menu Windows** → **Propriedades**. Aparecerá a caixa de diálogo **Propriedades**. Vai então à aba **Subversion**. Aqui podes colocar propriedades. Clica **Properties...** Na caixa de diálogo propriedades, ou efectuas duplo clique na `svn:externals` se já existir, ou clicas no botão **NOVO...** e seleccionas **externos** a partir do menu. Para adicionar um novo externo, clica **NOVO...** e então preenche a informação requerida na caixa de diálogo mostrada.



### Cuidado

Os URLs têm de ser convenientemente escapados, ou não irão funcionar, e.g. deverás substituir cada espaço com `%20`.

Se queres que o caminho local inclua espaços ou outro caracter especial, tens rodeá-lo em aspas, ou usar o caracter \ (barra invertida), como um caracter de escape, ao estilo da linha de comandos do linux, precedendo cada caracter 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

You should strongly consider using explicit revision numbers in all of your externals definitions, as described above. Doing so means that you get to decide when to pull down a different snapshot of external information, and exactly which snapshot to pull. Besides the common sense aspect of not being surprised by changes to third-party repositories that you might not have any control over, using explicit revision numbers also means that as you backdate your working copy to a previous revision, your externals definitions will also revert to the way they looked in that previous revision, which in turn means that the external working copies will be updated to match the way *they* looked back when your repository was at that previous revision. For software projects, this could be the difference between a successful and a failed build of an older snapshot of your complex code base.

A caixa de diálogo de edição para propriedades `svn:externals` permite-te seleccionar as externas e coloca-las explicitamente na revisão HEAD.

Se o projecto externo está no mesmo repositório, qualquer alteração que lá fizeres, será incluída na lista de submissões, quando submeteres o teu projecto principal.

If the external project is in a different repository, any changes you make to the external project will be shown or indicated when you commit the main project, but you have to commit those external changes separately.

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

These URLs always begin with the string `../` for example:

```
../../widgets/foo common/foo-widget
```

This will extract `http://example.com/svn/repos-1/widgets/foo` into `C:\Working\common\foo-widget`.

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

These URLs always begin with the string `^/` for example:

```
^/widgets/foo common/foo-widget
```

This will extract `http://example.com/svn/repos-1/widgets/foo` into `C:\Working\common\foo-widget`.

You can easily refer to other repositories with the same `SVNParentPath` (a common directory holding several repositories). For example:



```
^/../../repos-2/hammers/claw common/claw-hammer
```

This will extract `http://example.com/svn/repos-2/hammers/claw` into `C:\Working\common\claw-hammer`.

#### Relativo ao esquema

URLs beginning with the string `//` copy only the scheme part of the URL. This is useful when the same hostname must be accessed with different schemes depending upon network location; e.g. clients in the intranet use `http://` while external clients use `svn+ssh://`. For example:

```
//example.com/svn/repos-1/widgets/foo common/foo-widget
```

This will extract `http://example.com/svn/repos-1/widgets/foo` or `svn+ssh://example.com/svn/repos-1/widgets/foo` depending on which method was used to checkout `C:\Working`.

#### Relativo ao nome do servidor

URLs beginning with the string `/` copy the scheme and the hostname part of the URL, for example:

```
/svn/repos-1/widgets/foo common/foo-widget
```

This will extract `http://example.com/svn/repos-1/widgets/foo` into `C:\Working\common\foo-widget`. But if you checkout your working copy from another server at `svn+ssh://another.mirror.net/svn/repos-1/project1/trunk` then the external reference will extract `svn+ssh://another.mirror.net/svn/repos-1/widgets/foo`.

You can also specify a peg and operative revision for the URL if required. To learn more about peg and operative revisions, please read the *corresponding chapter* [<http://svnbook.red-bean.com/en/1.8/svn.advanced.pegrevs.html>] in the Subversion book.



### Importante

If you specify the target folder for the external as a subfolder like in the examples above, make sure that *all* folders in between are versioned as well. So for the examples above, the folder `common` should be versioned!

While the external will work in most situations properly if folders in between are not versioned, there are some operations that won't work as you expect. And the status overlay icons in explorer will also not show the correct status.

Se necessitas de mais informação em como o TortoiseSVN lida com Propriedades, consulta [Secção 4.18, "Configurações de Projecto"](#).

Para saberes mais sobre os diferentes métodos de aceder a subprojectos comuns, consulta [Secção B.6, "Incluir um subprojecto comum"](#).

## 4.19.2. 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.

- The path to the file external must be a direct child of the folder where you set the `svn:externals` property.

- 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.

### 4.19.3. Creating externals via drag and drop

If you already have a working copy of the files or folders you want to include as externals in another working copy, you can simply add those via drag and drop from the windows explorer.

Simply right drag the file or folder from one working copy to where you want those to be included as externals. A context menu appears when you release the mouse button: **SVN Add as externals here** if you click on that context menu entry, the `svn:externals` property is automatically added. All you have to do after that is commit the property changes and update to get those externals properly included in your working copy.

## 4.20. Ramificando/Etiquetando

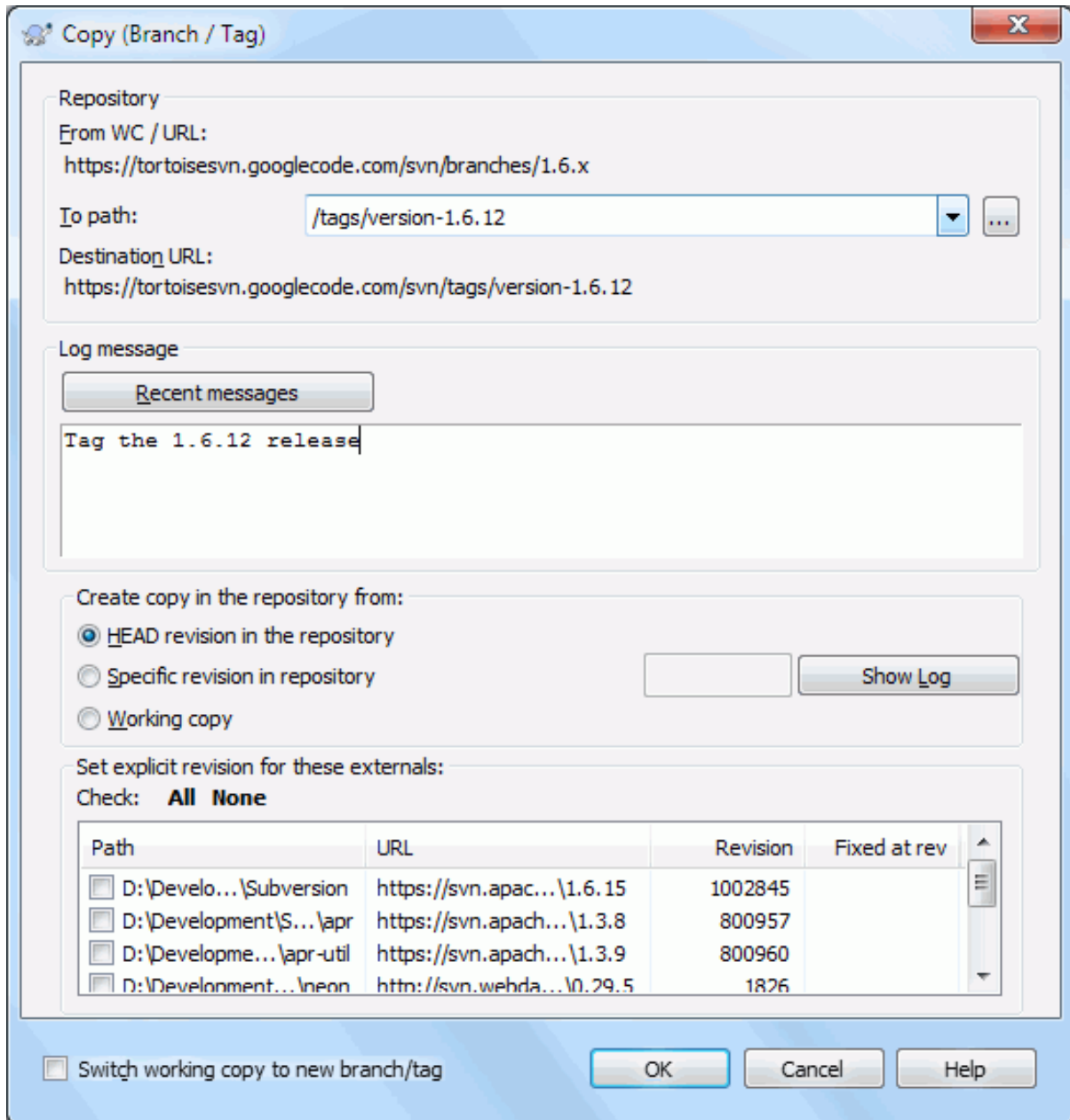
Uma das funcionalidades dos sistemas de controlo de versões, é a sua capacidade para isolar alterações em linhas de desenvolvimento separadas. Esta linha é conhecida como um *ramo*. Ramos são com frequência usados para experimentar novas funcionalidades, sem perturbar a linha principal de desenvolvimento 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.

### 4.20.1. 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 4.51. 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....

The default destination URL for the new branch will be the source URL on which your working copy is based. You will need to edit that URL to the new path for your branch/tag. So instead of

`http://svn.collab.net/repos/ProjectName/trunk`

you might now use something like

`http://svn.collab.net/repos/ProjectName/tags/Release_1.10`

If you can't remember the naming convention you used last time, click the button on the right to open the repository browser so you can view the existing repository structure.



### pastas intermédias

When you specify the target URL, all the folders up to the last one must already exist or you will get an error message. In the above example, the URL `http://svn.collab.net/repos/ProjectName/tags/` must exist to create the `Release_1.10` tag.

However if you want to create a branch/tag to an URL that has intermediate folders that don't exist yet you can check the option `Create intermediate folders` at the bottom of the dialog. If that option is activated, all intermediate folders are automatically created.

Note that this option is disabled by default to avoid typos. For example, if you typed the target URL as `http://svn.collab.net/repos/ProjectName/Tags/Release_1.10` instead of `http://svn.collab.net/repos/ProjectName/tags/Release_1.10`, you would get an error with the option disabled, but with the option enabled a folder `Tags` would be automatically created, and you would end up with a folder `Tags` and a folder `tags`.

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.

If your working copy has other projects included with `svn:externals` properties, those externals will be listed at the bottom of the branch/tag dialog. For each external, the target path and the source URL is shown.

If you want to make sure that the new tag always is in a consistent state, check all the externals to have their revisions pinned. If you don't check the externals and those externals point to a HEAD revision which might change in the future, checking out the new tag will check out that HEAD revision of the external and your tag might not compile anymore. So it's always a good idea to set the externals to an explicit revision when creating a tag.

The externals are automatically pinned to either the current HEAD revision or the working copy BASE revision, depending on the source of the branch/tag:

Copy Source	Pinned Revision
A revisão HEAD do repositório	external's repos HEAD revision
Revisão específica no repositório	external's repos HEAD revision

Copy Source	Pinned Revision
Cópia de trabalho	external's WC BASE revision

**Tabela 4.1. Pinned Revision**



### externals within externals

If a project that is included as an external has itself included externals, then those will not be tagged!  
Only externals that are direct children can be tagged.

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.

#### 4.20.2. Outras formas de criar um ramo ou etiqueta

Tu também podes criar um ramo ou etiqueta sem ter uma cópia de trabalho. Para o fazeres, abre o navegador de repositório. Aí podes arrastar pastas para uma nova localização. Terás de manter premida a tecla **Ctrl** enquanto arrastas de modo a criar uma cópia, de outro modo a pasta é movida, e não copiada.

Podes também arrastar uma pasta com o botão direito do rato. Uma vez que libertes o botão, poderás escolher, a partir do menu de contexto, o que queres fazer com a pasta a ser movida ou copiada. é claro que para criar um ramo ou etiqueta, tu tens de copiar a pasta e não movê-la.

No entanto, a outra maneira é a partir da caixa de diálogo registo. Podes visualizar a caixa de diálogo registo para o e.g. trunk, selecciona a revisão (a revisão HEAD no topo ou uma revisão anterior), clica com o botão direito e escolhe criar ramo/etiqueta a partir da revisão....

#### 4.20.3. 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. :-)

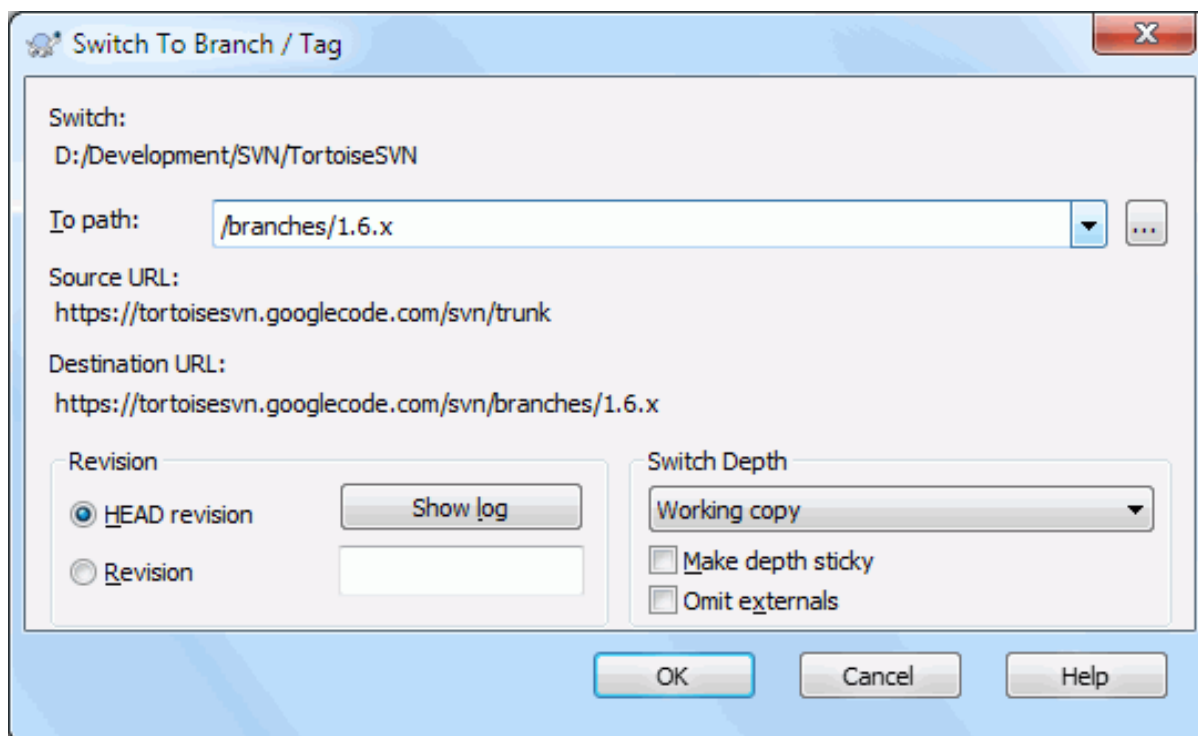
Para poderes trabalhar com o teu ramo ou etiqueta criado de fresco tens várias opções:

- 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 quiseses 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 4.52. 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.

- Etiquetas são usadas tipicamente para criar uma fotografia do projecto num patamar particular. Sendo que elas não são normalmente usadas em desenvolvimento - é para isso que existem os ramos, esta é a razão pela qual recomendamos, em primeiro lugar, a estrutura de repositório `/trunk /branches /tags`. Trabalhar numa revisão de etiqueta *não é uma boa ideia*, mas como os teus ficheiros não estão protegidos contra escrita, não existe nada que te impeça de o fazer por engano. No entanto, se tentares submeter para um caminho no repositório que contenha `/tags/`, o TortoiseSVN irá-te avisar.
- 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.

## 4.21. 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.

It is important to understand how branching and merging works in Subversion before you start using it, as it can become quite complex. It is highly recommended that you read the chapter *Branching and Merging* [<http://svnbook.red-bean.com/en/1.8/svn.branchmerge.html>] in the Subversion book, which gives a full description and many examples of how it is used.

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.

What you are asking Subversion to do is this: “ Calculate the changes necessary to get [FROM] revision 1 of branch A [TO] revision 7 of branch A, and apply those changes to my working copy (of trunk or branch B). ”

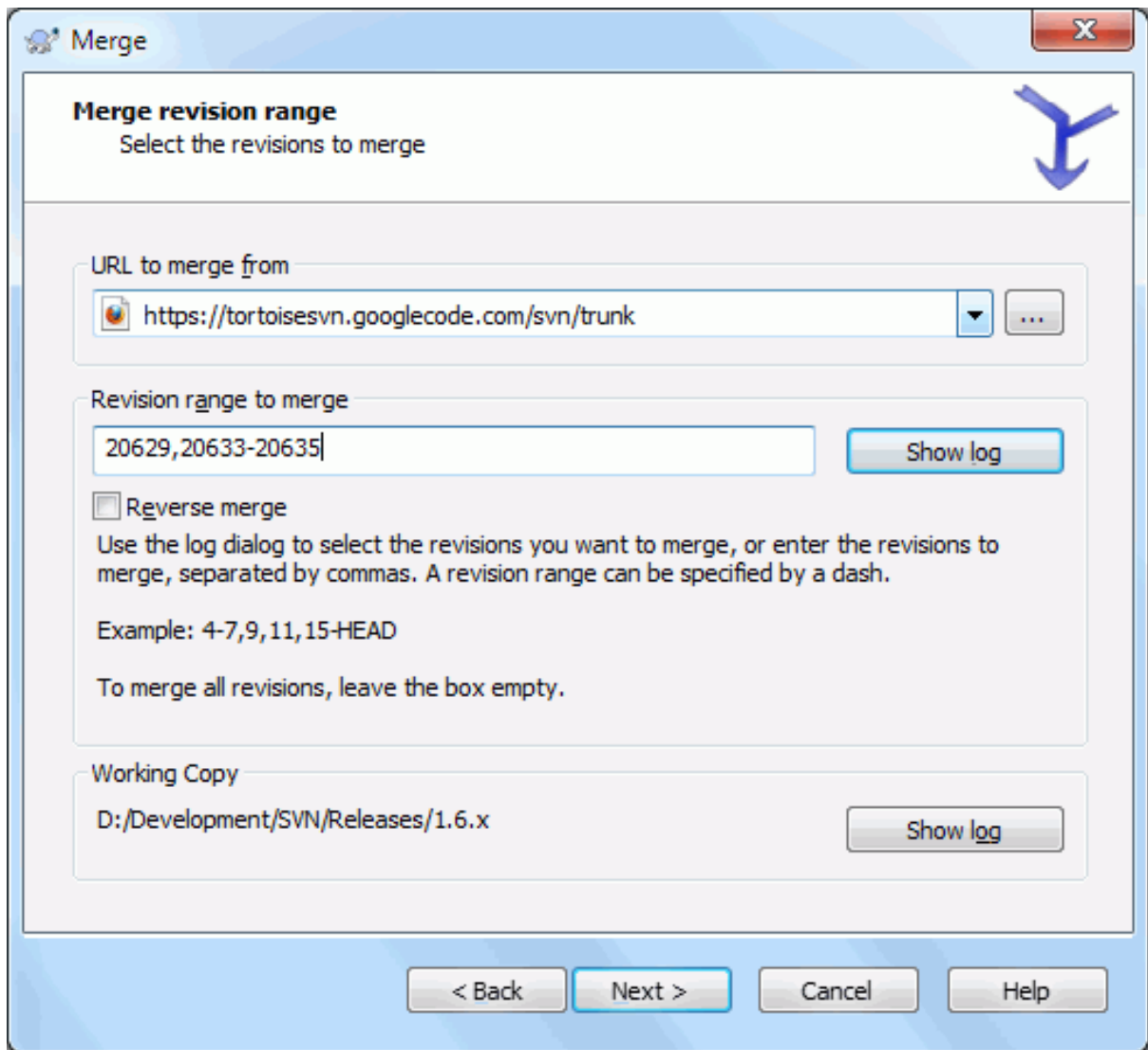
If you leave the revision range empty, Subversion uses the merge-tracking features to calculate the correct revision range to use. This is known as a reintegrate or automatic merge.

#### Integrar duas árvores diferentes

This is a more general case of the reintegrate method. What you are asking Subversion to do is: “ Calculate the changes necessary to get [FROM] the head revision of the trunk [TO] the head revision of the branch, and apply those changes to my working copy (of the trunk). ” The net result is that trunk now looks exactly like the branch.

If your server/repository does not support merge-tracking then this is the only way to merge a branch back to trunk. Another use case occurs when you are using vendor branches and you need to merge the changes following a new vendor drop into your trunk code. For more information read the chapter on [vendor branches](http://svnbook.red-bean.com/en/1.8/svn.advanced.vendorbr.html) [http://svnbook.red-bean.com/en/1.8/svn.advanced.vendorbr.html] in the Subversion Book.

### 4.21.1. A Integrar Um Intervalo de Revisões



**Figura 4.53. 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.

If you are merging from a renamed or deleted branch then you will have to go back to a revision where that branch still existed. In this case you will also need to specify that revision as a peg revision in the range of revisions being merged (see below), otherwise the merge will fail when it can't find that path at HEAD.

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.

Se precisas de especificar uma revisão cavilha para a integração, adiciona a revisão cavilha no fim das revisões, e.g. 5-7,10@3. No exemplo acima, as revisões 5,6,7 e 10 seriam integradas, sendo a 3 a revisão cavilha.





## 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 estiveres a integrar revisões em bloco, o método mostrado no livro do Subversion integra-te a 100-200 desta vez e 200-300 da próxima. Com o TortoiseSVN integra-te a 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 4.21.5, “Rastreamento de Integração”](#) para saberes mais.

Quando o seguimento de integração é usado, a caixa de diálogo de registo irá mostrar as revisões previamente integradas e as revisões antecedentes ao ponto do antecessor comum, i.e. antes do ramo ser copiado, a cinzento. A caixa de verificação **Ocultar revisões não.integráveis** permite-te filtrar completamente essas revisões, para que vejas apenas as revisões que *podem* ser integradas.

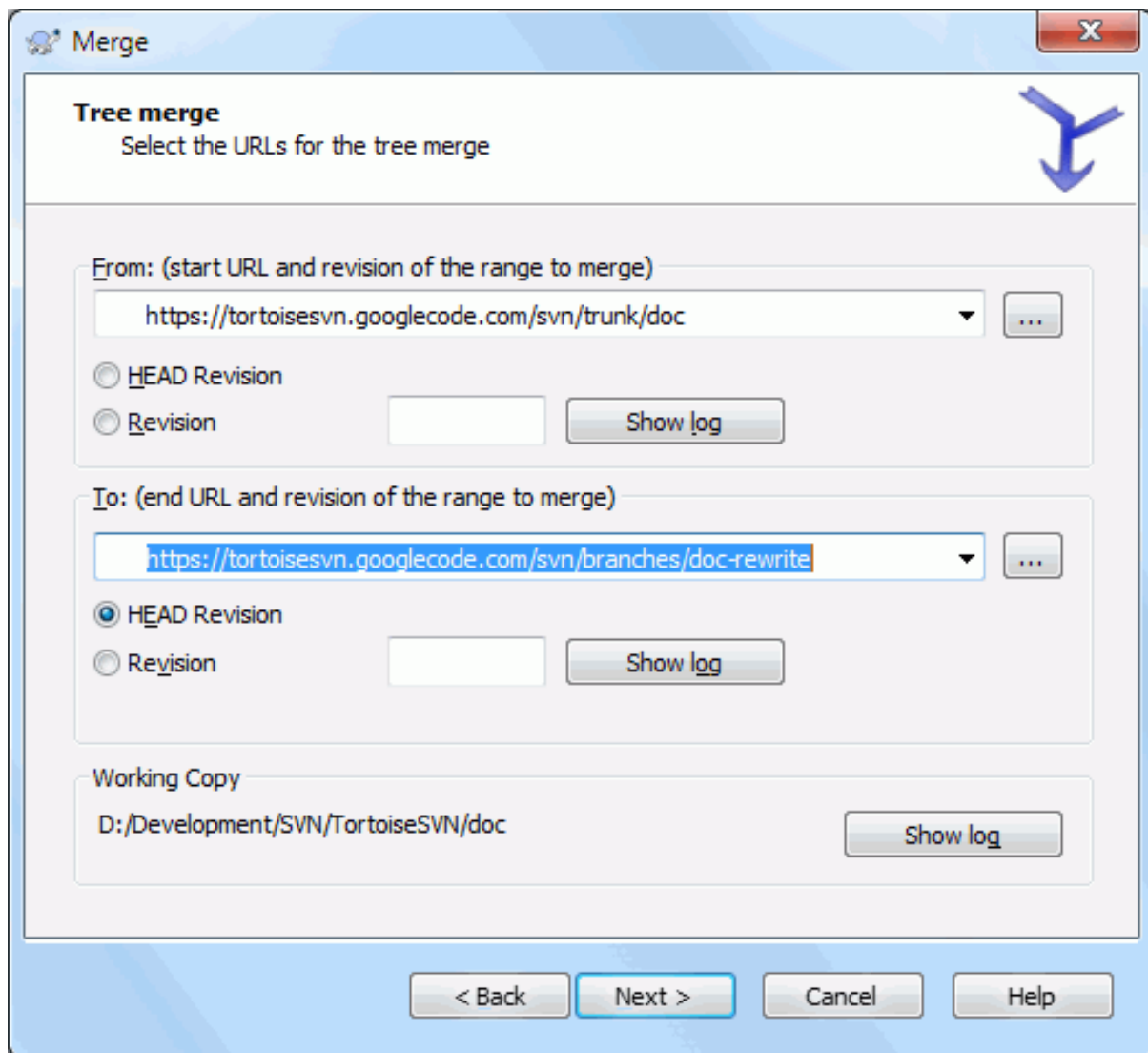
Se podem estar outras pessoas a submeter alterações então tem cuidado no uso da revisão HEAD. Poderá não se referir à revisão que estás a pensar se alguém efectuou uma submissão após a tua última actualização.

If you leave the range of revisions empty or have the radio button **all revisions** checked, then Subversion merges all not-yet merged revisions. This is known as a reintegrate or automatic merge.

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 nível infinito (sem checkouts dispersos), 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.

Clica Seguinte e segue para [Secção 4.21.3, “Opções de Integração”](#).

#### 4.21.2. A Integrar Duas Árvores Diferentes



**Figura 4.54. 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, deves usar números de revisões específicas para evitar perder submissões mais recentes.

Podes também usar o Mostrar Registo para seleccionar a revisão.

### 4.21.3. 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 o nível a usar na integração, i.e. até que ponto deverá a integração penetrar na tua cópia de trabalho. Os termos do nível usados estão descritos em [Secção 4.3.1, “Profundidade do Checkout”](#). O nível por defeito é a Cópia de trabalho, e é quase sempre o que precisas.

Most of the time you want merge to take account of the file's history, so that changes relative to a common ancestor are merged. Sometimes you may need to merge files which are perhaps related, but not in your repository. For example you may have imported versions 1 and 2 of a third party library into two separate directories. Although they are logically related, Subversion has no knowledge of this because it only sees the tarballs you imported. If you attempt to merge the difference between these two trees you would see a complete removal followed by a complete add. To make Subversion use only path-based differences rather than history-based differences, check the **Ignore ancestry** box. Read more about this topic in the Subversion book, [Noticing or Ignoring Ancestry](#) [<http://svnbook.red-bean.com/en/1.8/svn.branchmerge.advanced.html#svn.branchmerge.advanced.ancestry>].

Tu podes especificar a forma como os fins-de-linha e espaços-em-branco são lidados. Essas opções são descritas em [Secção 4.11.2, “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.

A caixa de verificação **Force the merge** é usada para evitar um conflito de árvore quando um remover afectar um ficheiro que foi modificado localmente ou não está versionado. Se o ficheiro foi removido então não há maneira de o recuperar, sendo por isso que essa opção não está marcada por defeito.

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.

Agora tudo está configurado, tudo o que tens de fazer é clicar no botão **Integrar**. Se queres pré-visualizar os resultados o **Testar Integração** simula a operação de integração, mas *não* modifica de todo a cópia de trabalho. Mostra-te uma lista de ficheiros que serão alterados por uma integração real, e anota os ficheiros onde os conflitos *podem* ocorrer. Porque o rastreador de integração torna o processo de integração muito mais complexo, não existe maneira garantida de determinar em avanço se uma integração irá ser concluída sem conflitos, pelo que os ficheiros marcados como em conflito numa integração de teste, poderão de facto integrar-se sem qualquer problema.

The merge progress dialog shows each stage of the merge, with the revision ranges involved. This may indicate one more revision than you were expecting. For example if you asked to merge revision 123 the progress dialog will report “Merging revisions 122 through 123”. To understand this you need to remember that Merge is closely related to Diff. The merge process works by generating a list of differences between two points in the repository, and applying those differences to your working copy. The progress dialog is simply showing the start and end points for the diff.

### 4.21.4. 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.



#### Dica

Quando são integradas as revisões na cópia de trabalho, o TortoiseSVN gera uma mensagem de registo para todas as revisões integradas. Essas ficam então disponíveis a partir do botão **Mensagens recentes** na janela de diálogo submeter.

Para personalizar essa mensagem gerada, coloca as respectivas propriedades de projecto na tua cópia de trabalho. Consultar [Secção 4.18.3.10, “Templates de mensagens de registo”](#)

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) [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 remove-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.

## 4.21.5. 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.

Mais informações sobre o controlo de integração do Subversion em; [Merge tracking documentation](http://svn.apache.org/repos/asf/Subversion/trunk/notes/merge-tracking/index.html) [http://svn.apache.org/repos/asf/Subversion/trunk/notes/merge-tracking/index.html].

#### 4.21.6. Handling Conflicts after Merge



#### Importante

The text in the conflict resolver dialogs are provided by the SVN library and might therefore not (yet) be translated as the TortoiseSVN dialogs are. Sorry for that.

Merging does not always go smoothly. Sometimes there is a conflict. TortoiseSVN helps you through this process by showing the *merge conflict* dialog.

#### Figura 4.55. The Merge Conflict Dialog

It is likely that some of the changes will have merged smoothly, while other local changes conflict with changes already committed to the repository. All changes which can be merged are merged. The Merge Conflict dialog gives you different ways of handling the lines which are in conflict.

For normal conflicts that happen due to changes in the file content or its properties, the dialog shows buttons which allow you to chose which of the conflicting parts to keep or reject.

##### Postpone

Don't deal with the conflict now. Let the merge continue and resolve the conflicts after the merge is done.

##### Accept base

This leaves the file as it was, without neither the changes coming from the merge nor the changes you've made in your working copy.

##### Accept incoming

This discards all your local changes and uses the file as it arrives from the merge source.

##### Reject incoming

This discards all the changes from the merge source and leaves the file with your local edits.

##### Accept incoming for conflicts

This discards your local changes where they conflict with the changes from the merge source. But it leaves all your local changes which don't conflict.

##### Reject conflicts

This discards changes from the merge source which conflict with your local changes. But it keeps all changes that don't conflict with your local changes.

##### Mark as resolved

Marks the conflicts as resolved. This button is disabled until you use the button **Edit** to edit the conflict manually and save those changes back to the file. Once the changes are saved, the button becomes enabled.

##### Editar

Starts the merge editor so you can resolve the conflicts manually. Don't forget to save the file so the button **Mark as resolved** becomes enabled.

If there's a tree conflict, please first see [Secção 4.6.3, "Conflitos de Árvore"](#) about the various types of tree conflicts and how and why they can happen.

To resolve tree conflicts after a merge, a dialog is shown with various options on how to resolve the conflict:

### Figura 4.56. The Merge Tree Conflict Dialog

Since there are various possible tree conflict situations, the dialog will show buttons to resolve those depending on the specific conflict. The button texts and labels explain what the option to resolve the conflict does. If you're not sure, either cancel the dialog or use the Postpone button to resolve the conflict later.

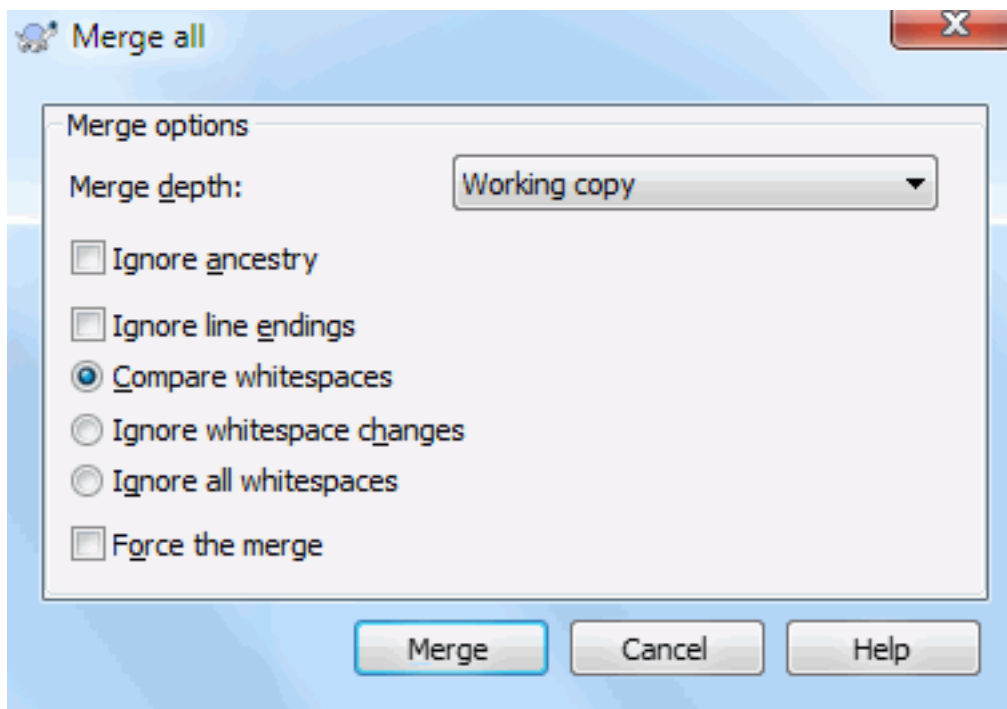
#### 4.21.7. Manutenção do Ramo de Funcionalidade

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**.

Another (fast) way to merge all changes from trunk to the feature branch is to use the TortoiseSVN → Merge all... from the extended context menu (hold down the **Shift** key while you right click on the file).



### Figura 4.57. The Merge-All Dialog

This dialog is very easy. All you have to do is set the options for the merge, as described in [Secção 4.21.3, “Opções de Integração”](#). The rest is done by TortoiseSVN automatically using merge tracking.

## 4.22. 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 estás a usar um acesso `file://`, então está claro que só o teu cliente necessita de ser actualizado.



### Os três significados de “Bloquear”

Nesta secção, e em particamente em qulaquer parte deste livro, as palavras “bloqueio” e “bloquear” descrevem um mecanismo de mutua exclusão entre utilizadores para evitar colisões ao submeter. Infelizmente, existem outros tipos de “bloqueio”s com que o Subversion, e por consequência este livro, deve ter em conta.

The second is *working copy locks*, used internally by Subversion to prevent clashes between multiple Subversion clients operating on the same working copy. Usually you get these locks whenever a command like `update/commit/...` is interrupted due to an error. These locks can be removed by running the `cleanup` command on the working copy, as described in [Secção 4.17, “Limpar”](#).

E por terceiro, ficheiros e pastas podem ficar bloqueadas se estiverem a ser usadas por outro processo, por exemplo, se tiveres um documento word aberto no Word, esse ficheiro está bloqueado e não pode ser acedido pelo TortoiseSVN.

Por norma podes descartar esses outros tipos de bloqueios, até acontecer algo errado que requeira o teu apoio

### 4.22.1. 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.



### Importante

O bloqueio não é atribuído apenas a um utilizador específico, mas a um utilizador específico e a uma cópia de trabalho. Tendo um bloqueio na cópia de trabalho previne também que o mesmo utilizador submeta um ficheiro bloqueado a partir de uma outra cópia de trabalho.

Como exemplo, imagina que o utilizador Jon tem uma cópia de trabalho no seu PC de escritório. Aí ele começa a trabalhar numa imagem, e para tal adquire um bloqueio para esse ficheiro. Quando abandona o escritório ele ainda não finalizou o trabalho nesse ficheiro, pelo que não libertou esse bloqueio. De volta a casa, o Jon também tem uma cópia de trabalho, e decide trabalhar mais um pouco no projecto. Mas ele não pode modificar ou submeter o mesmo ficheiro de imagem, porque o bloqueio para esse ficheiro reside na sua cópia de trabalho do escritório.

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 caso 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.

#### 4.22.2. 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....

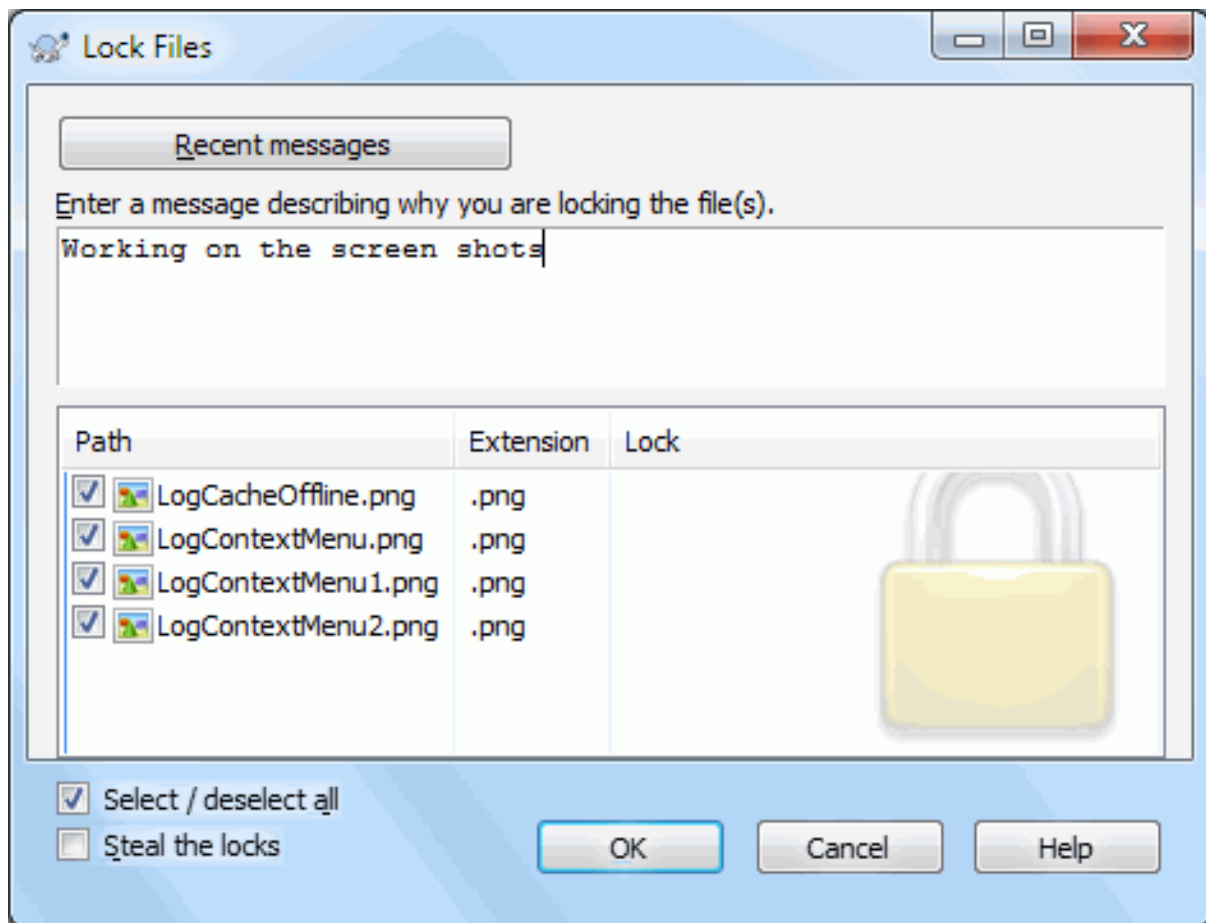


Figura 4.58. A Caixa de Diálogo Bloquear



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**.

Poderás configurar a propriedade de projecto `tsvn:logtemplatelock` para fornecer aos utilizadores uma mensagem de modelo (template), para que a preencham como mensagem de bloqueio. Consultar a secção [Secção 4.18, “Configurações de Projecto”](#) para instruções em como configurar propriedades.

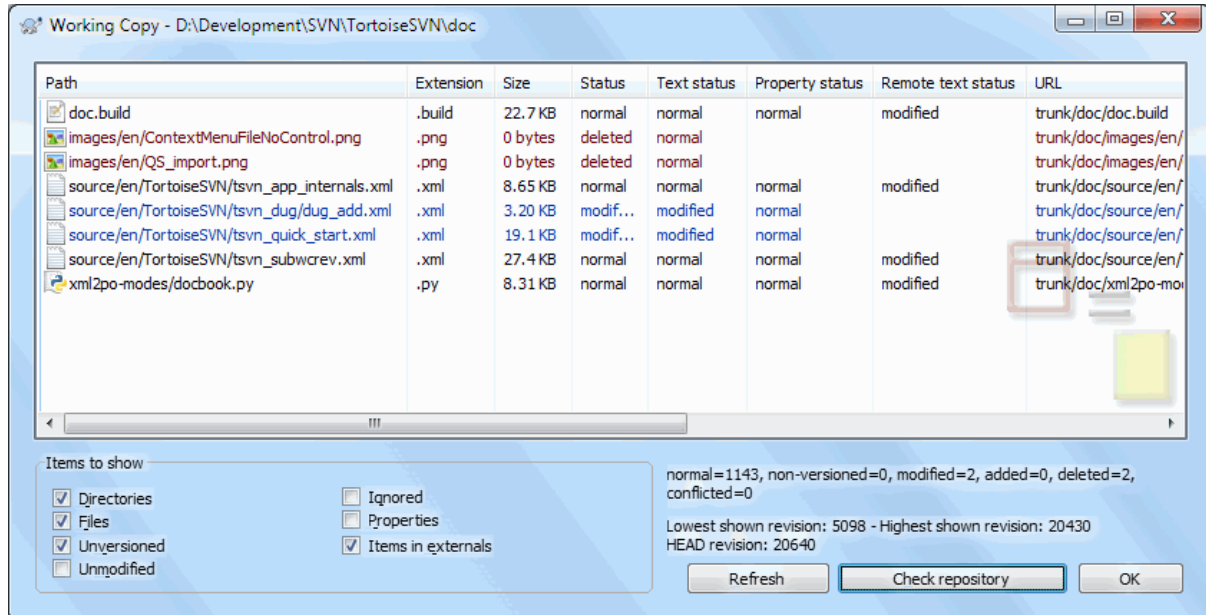
Se seleccionas um ficheiro e então usas **TortoiseSVN → Obter bloqueio...**, a caixa de diálogo bloquear irá abrir com *todos os* ficheiros em *todas* as subpastas seleccionadas para bloquear. Se realmente queres bloquear uma hierarquia inteira, esta é a maneira de o fazer, mas podes-te tornar muito impopular junto dos teus colegas de trabalho se bloqueias todo o projecto. Usa com moderação...

### 4.22.3. 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.

### 4.22.4. Verificar o Estado dos Bloqueio



**Figura 4.59. 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.

### 4.22.5. 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 4.18, “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 4.18.1.5, “Configuração automática de propriedades”](#) para mais informações.

### 4.22.6. 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.

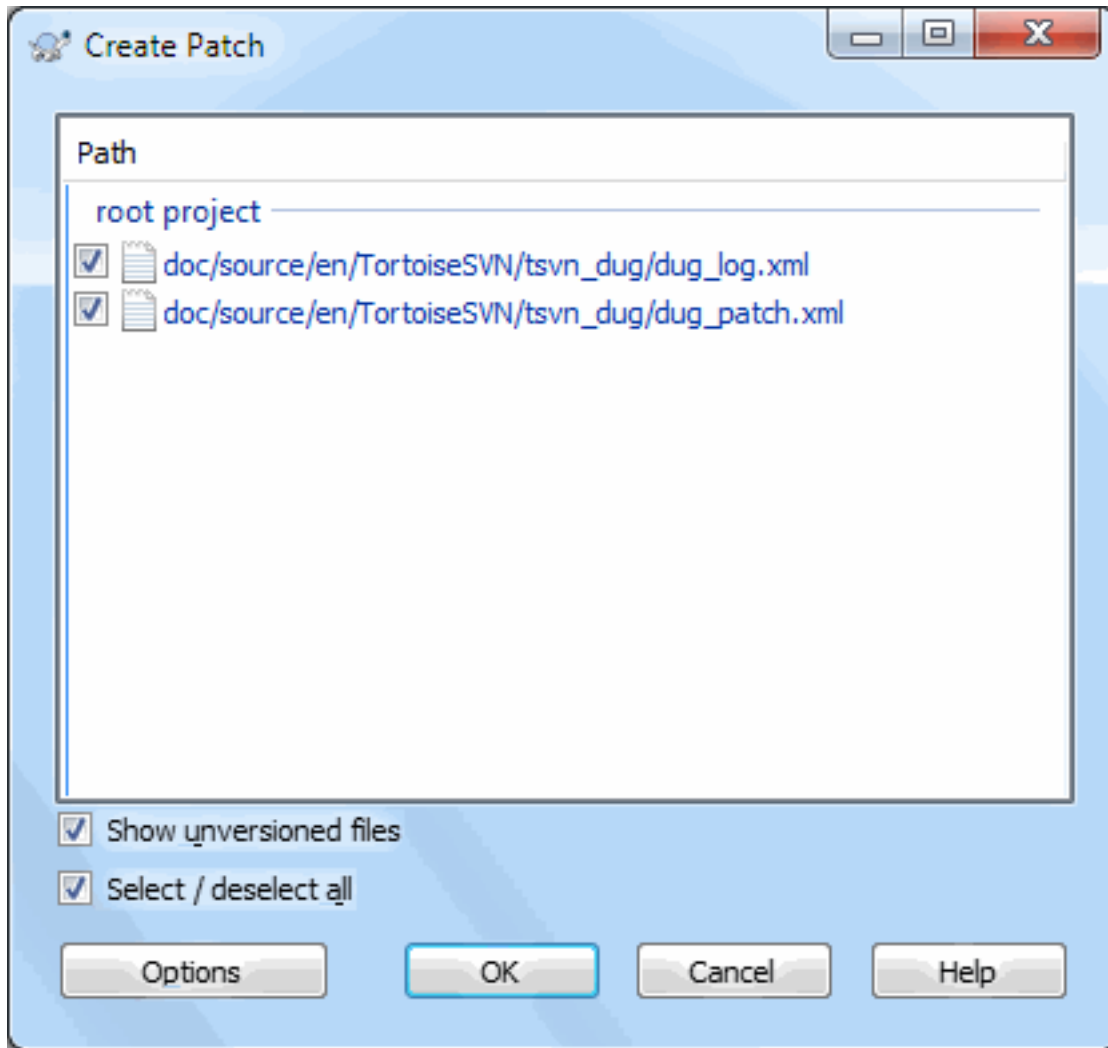
## 4.23. Criar e Aplicar Correções

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.

### 4.23.1. 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 4.60. 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.

Ao clicar no botão de Opções, podes especificar como o caminho é criado. Por exemplo, podes especificar que as mudanças no final da linha ou espaços em branco não são incluídas no arquivo do caminho final.

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.



### Dica

Do not save the patch file with a `.txt` extension if you intend to send it via email to someone else. Plain text files are often mangled with by the email software and it often happens that whitespaces

and newline chars are automatically converted and compressed. If that happens, the patch won't apply smoothly. So use `.patch` or `.diff` as the extension when you save the patch file.

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 quiseres transferir alterações de uma para outra, a correcção na área de transferência é a maneira conveniente de o fazer.

Se preferires podes criar um ficheiro correcção a partir das caixas de diálogo **Submeter** ou **Verificar alterações**. Selecciona apenas os ficheiros e usa o item do menu de contexto para criar a correcção a partir desses ficheiros. Se quiseres ver a caixa de diálogo **Opções** terás de manter premida a tecla **shift** quando clicares à direita.

### 4.23.2. 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.

A partir do menu de contexto desse ficheiro, clica em **TortoiseSVN → Aplicar correcção...** Isto fará aparecer uma caixa de diálogo abrir ficheiro, que te permitirá seleccionar o ficheiro de correcção a aplicar. Por defeito, só são mostrados ficheiros `.patch` ou `.diff`, mas tu podes optar por “Todos ficheiros”. Se guardaste previamente uma correcção na área de transferência, podes usar o **Abrir da área de transferência...** a partir da caixa de diálogo, abrir ficheiro. Ter em atenção que esta opção só aparece se guardaste uma correcção na área de transferência usando **TortoiseSVN → Criar correcção....** Copiar uma correcção para a área de transferência, a partir de outra aplicação, não fará o botão aparecer.

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.

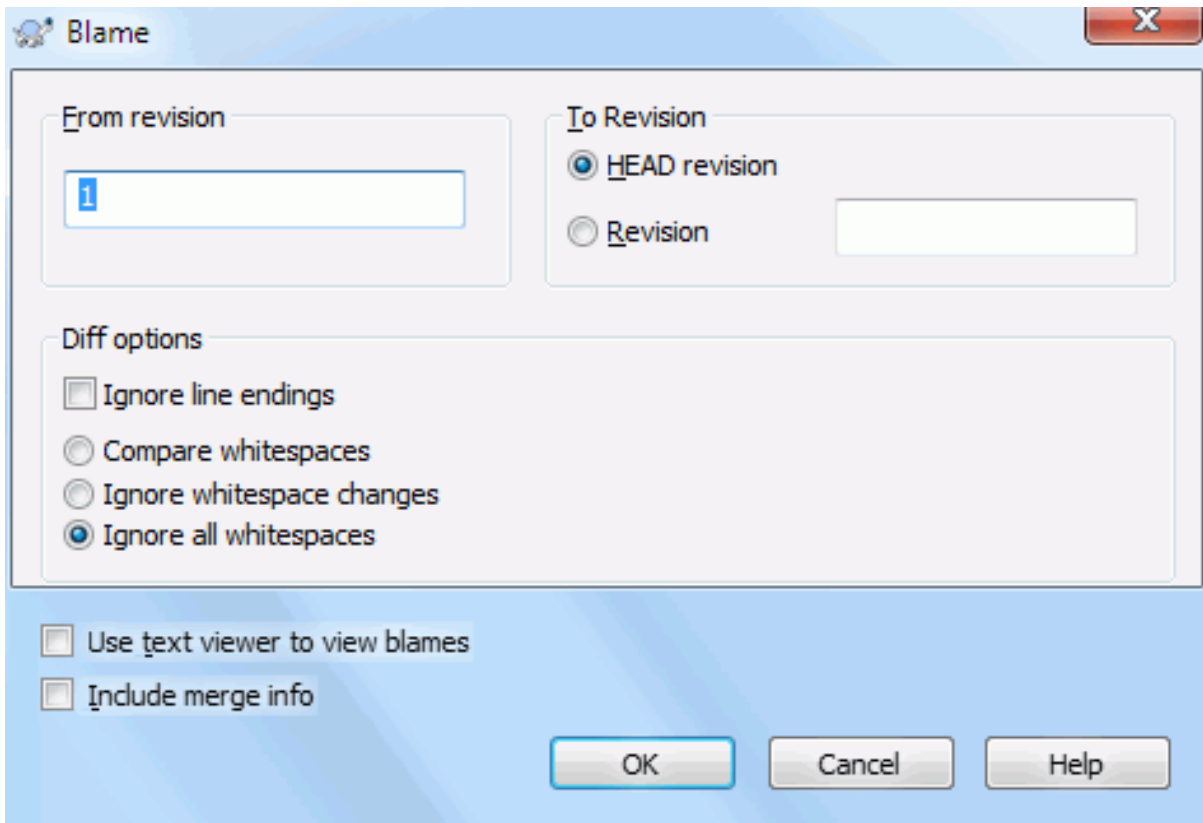
A correcção do programador remoto foi agora aplicada na tua cópia de trabalho, então precisas agora de submeter de modo a permitir que todos possam aceder às alterações a partir do repositório.

## 4.24. 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.

#### 4.24.1. Responsabilidade para Ficheiros



**Figura 4.61. A Caixa de Diálogo de Anotar/Responsabilizar**

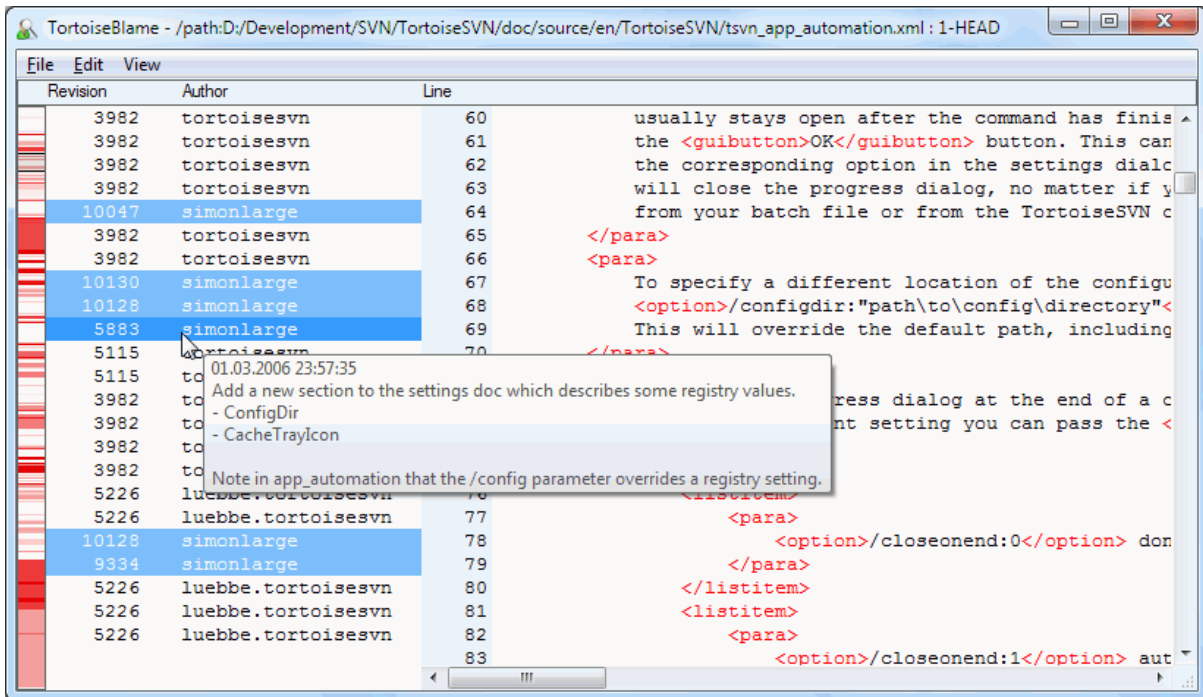
Se não estás interessado nas alterações em revisões anteriores, podes fixar a revisão a partir da qual a responsabilização deve começar. Fixa para 1, se queres a responsabilização para *cada* revisão.

Por defeito, o ficheiro responsabilizado é visualizado usando o *TortoiseBlame*, que salienta as diferentes revisões de forma a tornar mais fácil a sua leitura. Se desejares imprimir ou editar o ficheiro de responsabilidade, selecciona Usar visualizador 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 4.11.2, “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.

Ter em atenção que se uma etiqueta é usada como origem para uma cópia, talvez um novo ramo baseado numa etiqueta, então essa etiqueta será mostrada como um nó separado em vez de dobrada.

Once you press OK TortoiseSVN starts retrieving the data to create the blame file. Once the blame process has finished the result is written into a temporary file and you can view the results.



**Figura 4.62. 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 ecrã 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 e pediste info. de integração ao iniciar a responsabilização, então as linhas integradas são mostradas de um modo ligeiramente diferente. Onde foi alterada uma linha como resultado de uma integração de outro ramo, o TortoiseBlame irá mostrar a revisão e o autor da última alteração no ficheiro original em vez da revisão onde foi efectuada a integração. Essas linhas são identificadas ao mostrar a revisão e o autor em itálico. A revisão onde foi efectuada a integração é mostrada em separado na tooltip quando passas o rato sobre as colunas de info. de responsabilidade. Se não pretenderes que as linhas integradas sejam mostradas desta maneira, desselecciona a caixa **Incluir info de integração** ao iniciar a responsabilização.

Se quiseres ver os caminhos envolvidos na mesclagem, selecciona **Ver → Mesclar caminhos**. Isso irá mostrar o caminho onde a linha foi modificada pela última vez, excluindo as variações resultantes de uma fusão.

A revisão mostrada na informação de responsabilidade representa a última revisão onde o conteúdo dessa linha foi alterado. Se o ficheiro foi criado ao copiar outro ficheiro, então até alterares a linha, a sua revisão de responsabilidade irá mostrar a última alteração no ficheiro fonte original, e não a revisão onde a cópia foi efectuada. Isto também se aplica aos caminhos mostrados com a informação de integração. O caminho mostra a localização do repositório onde a última alteração foi efectuada a essa linha.

As preferências para o TortoiseBlame podem ser acedidas usando **TortoiseSVN → Preferências...** na aba TortoiseBlame. Consultar [Secção 4.31.9, “Preferências do TortoiseBlame”](#).

#### 4.24.2. Diferenças de responsabilidade

Limpa o estado da cópia de trabalho

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

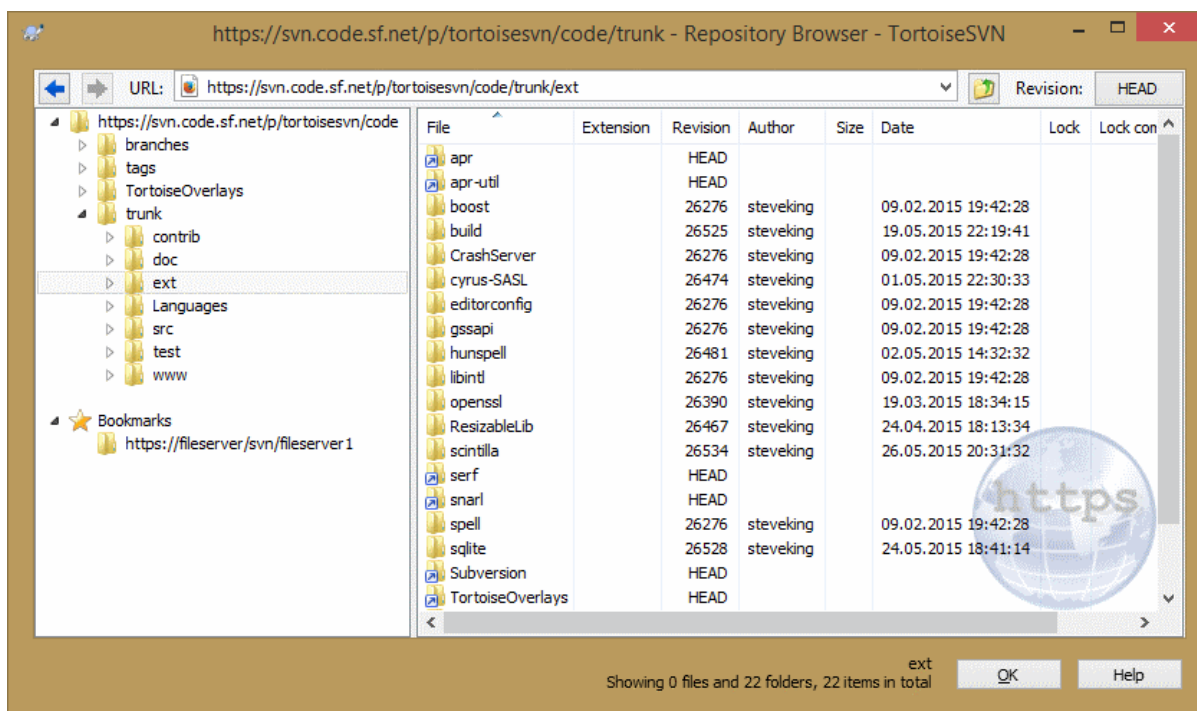
Selecciona uma revisão no painel de topo e então pega num ficheiro no painel de fundo e selecciona **Menu de contexto → Responsabilizar alterações**. Esta operação irá buscar os dados de responsabilidade para a revisão seleccionada e para a revisão anterior, e usar o visualizador de comparação para comparar 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.

#### 4.25. 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 4.63. 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 direita 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.

Pastas incluídas com a propriedade `svn:externals` são também mostradas no navegador de repositório. Essas pastas são visualizadas com uma pequena seta, para indicar que não fazem parte da estrutura do repositório, como hiperligações.

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.
- Editar o ficheiro seleccionado. Este irá efectuar checkout numa cópia de trabalho temporário e arrancar o editor por defeito para esse tipo de ficheiro. Quando fechores o programa de edição, se foram guardadas alterações irá aparecer uma caixa de diálogo submeter, permitindo que introduzas um comentário e submetas a alteração.
- Mostrar o registo de revisão para esse ficheiro, ou mostra o grafo de todas as revisões, para que possas ver de onde o ficheiro veio.
- Responsabiliza o ficheiro, para veres quem alterou que linha e quando.
- Checkout de um único ficheiro. Isso cria uma cópia de trabalho “dispersa” que contém apenas este ficheiro.
- Remove ou renomea o ficheiro.
- Guarda uma cópia não versionada do ficheiro para o teu disco rígido.
- Copia o URL da barra de endereços para o clipboard.



- 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.
- Cria um atalho para que possas rapidamente arrancar de novo o navegador de repositório, a abrir directamente nesta localização.

O menu de contexto para uma pasta, permite-te:

- Mostrar o registo de revisão para esse ficheiro, ou mostra o grafo 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 e pastas não versionadas directamente para o repositório. Isto é efectivamente a operação importar do Subversion.
- Remover ou renomear a pasta.
- Fazer uma cópia da pasta, numa parte diferente do repositório, ou para uma cópia de trabalho originária do mesmo repositório. Isto também podes ser usado para criar um ramo/etiqueta sem necessidade de ter uma cópia de trabalho exportada.
- 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 multiplas 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.

Reverter recursivamente todas as alterações

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

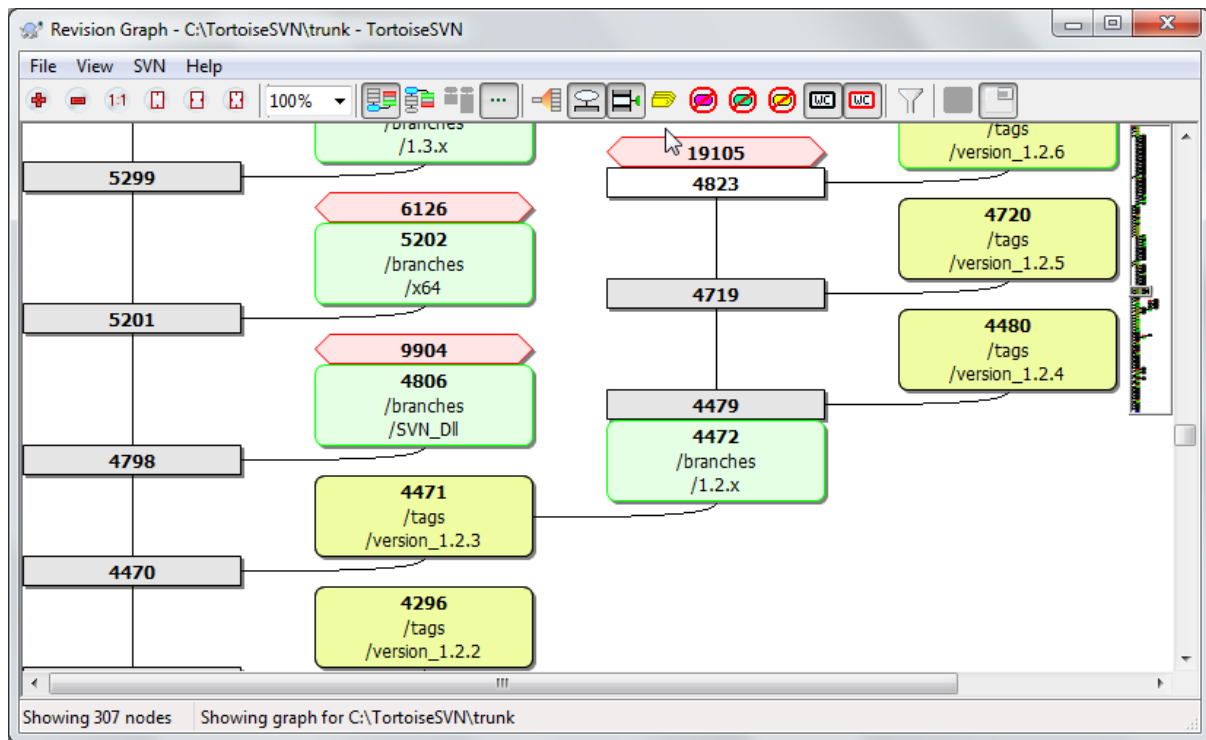
arrastar com o botão esquerdo. Nesse caso, a caixa de diálogo renomear é mostrada, e podes então introduzir um novo nome para a pasta ou ficheiro.

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**.

Bookmarked urls/repositories are shown below the current repository folders in the left tree view. You can add entries there by right clicking on any file or folder and select Context Menu → Add to Bookmarks. Clicking on a bookmark will browse to that repository and file/folder.

## 4.26. Grafo de Revisões



**Figura 4.64. O Grafo 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 grafo ou uma estrutura em árvore. É quando necessitas de usar TortoiseSVN → Grafo 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 grafo, 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.

#### 4.26.1. Nós do Grafo de Revisões

Cada nó do grafo 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 com um rectângulo de cantos arredondados. A sua cor por defeito é o verde. As etiquetas e trunks são tratados como um caso especial e usam uma sombra diferente dependendo das 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 a 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 grafo 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**, 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 poderás optar por mostrar a revisão BASE no grafo usando **Show WC revision**, que marca o nó BASE com uma moldura a negrito.

Cópia de trabalho modificada

Se invocaste o grafo 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 grafo só mostra os pontos onde itens foram adicionados, copiados ou removidos. Mostrando cada revisão do projecto irá gerar um grafo 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 grafos 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 grafo 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.

#### 4.26.2. 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 à 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 grafo muito amplo.

Este modo agrupa alterações por ramo, pelo que não existe uma ordenação global por revisão: Revisões consecutivas num ramo serão (usualmente) mostradas em linhas consecutivas. No entanto, sub-ramos serão ordenados de tal maneira que ramos posteriores serão mostrados na mesma coluna, acima de ramos anteriores, de modo a manter o grafo estreito. Como resultado, uma dada linha poderá conter alterações de diferentes revisões.

#### Mais antigo no topo

Normalmente o grafo 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 grafo 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

Esta opção está normalmente activada e evita o mostrar do grafo com muitas e confusas linhas cruzadas. No entanto isso poderá fazer com que as colunas do layout apareçam em sítios menos lógicos, por exemplo numa linha diagonal em vez de em coluna, e o grafo poderá requerer uma área maior para ser desenhado. Se isso for um problema poderás desactivar a opção a partir do menu **Vista**.

#### 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 grafo) foi alterado. Para históricos longos isto pode produzir um grafo realmente gigante.

#### Mostrar revisões HEAD

Isto assegura que a última revisão de cada ramo é sempre mostrada no grafo.

#### 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, o mostrar cada etiqueta como um nó separado no grafo ocupará muito espaço e obscurecerá, a mais interessante, estrutura de de ramos de desenvolvimento. Ao mesmo tempo deverás ter a possibilidade de aceder facilmente ao conteúdo da etiqueta, para que possas comparar revisões. Esta opção esconde os nós das etiquetas, e mostra-as em alternativa na tooltip do nó de onde foram copiadas. O ícone de etiqueta no lado direito do nó fonte indica que foram feitas etiquetas. Isto simplifica bastante a vista.

Nota que, se uma etiqueta for utilizada em si como uma fonte para obter uma cópia, talvez um novo ramo baseado numa etiqueta, então essa etiqueta será mostrada como um nó separado em vez de dobrado.

#### Esconder caminhos removidos

Esconde caminhos que já não estão presentes na revisão HEAD do repositório, e.g. ramos removidos.

Se seleccionaste a opção **Dobrar etiquetas**, então será mostrado um ramo removido, a partir do qual foram criadas as etiquetas, de outro modo as etiquetas também desapareceriam. A última revisão que foi etiquetada será mostrada na cor usada para os nós removidos, em vez de mostra uma revisão removida à parte.

Se seleccionares a opção **Ocultar etiquetas**, em seguida, esses ramos vão desaparecer, visto que não são necessários para mostrar as etiquetas.

#### Esconder ramos não usados

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 grafo a revisão que corresponde à revisão actualizada do item, para o qual obtiveste o grafo. 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 grafo usando **F5**.

#### Filtro

Por vezes o grafo 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.

Se esconderes um caminho em particular e esse nó tiver nós filho, os filhos serão mostrados como uma árvore separada. Se quiseres também esconder todos os filhos usa a caixa de verificação **Remover as sub-árvore(s) completas**.

#### Listas de árvore

Quando o grafo 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 grafo inteiro com a janela da vista corrente como um rectângulo que podes arrastar. Isto permite-te navegar pelo grafo de forma mais fácil. De notar que para grafos muito grandes a visão global pode-se tornar inutil, devido ao factor de ampliação (zoom) extremo, e por isso não ser mostrado nesses casos.

### 4.26.3. 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** serás 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.

### 4.26.4. 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.

### 4.26.5. 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 subtracção 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.

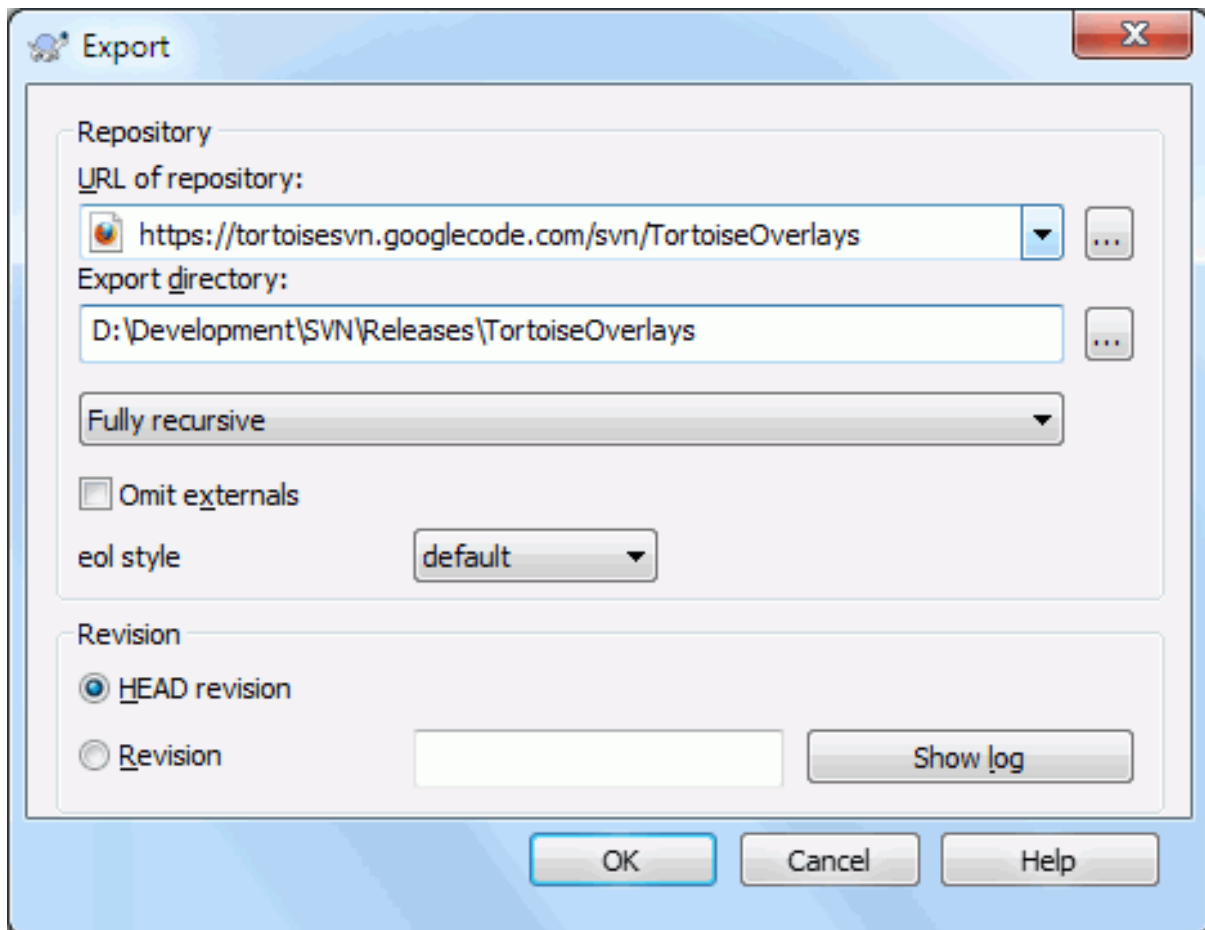


Clica no botão com o círculo para re-anexar a árvore dividida. Quando uma árvore foi dividida este botão permanece visível para indicar que existe uma sub-árvore separada.

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.

## 4.27. Exportar uma Cópia de Trabalho do Subversion

Sometimes you may want a clean copy of your working tree without the `.svn` directory, e.g. to create a zipped tarball of your source, or to export to a web server. Instead of making a copy and then deleting the `.svn` directory manually, TortoiseSVN offers the command TortoiseSVN → Export.... Exporting from a URL and exporting from a working copy are treated slightly differently.



**Figura 4.65. 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.

If you execute this command on your working copy you'll be asked for a place to save the *clean* working copy without the `.svn` folder. By default, only the versioned files are exported, but you can use the **Export unversioned files too** checkbox to include any other unversioned files which exist in your WC and not in the repository. External references using `svn:externals` can be omitted if required.

Another way to export from a working copy is to right drag the working copy folder to another location and choose Context Menu → SVN Export versioned items here or Context Menu → SVN Export all items here or

Context Menu → SVN Export changed items here. The second option includes the unversioned files as well. The third option exports only modified items, but maintains the folder structure.

Ao exportar uma cópia de trabalho, se a pasta de destino já contiver uma pasta com o mesmo nome daquela que estás a exportar, será te então dada a opção de 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 4.25, “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 4.11.3, “Comparando Pastas”.

If you want to export your working copy tree structure but containing only the files which are locally modified, refer to SVN Export changed items here above.

#### 4.27.1. Remover uma cópia de trabalho do controlo de versões

Por vezes terás uma cópia de trabalho que pretendes converter de volta para uma pasta normal, ou seja, sem a pasta .svn. Tudo o que precisarás de fazer é apagar a pasta .svn da raiz da tua cópia de trabalho.

Alternatively you can export the folder to itself. In Windows Explorer right drag the working copy root folder from the file pane onto itself in the folder pane. TortoiseSVN detects this special case and asks if you want to make the working copy unversioned. If you answer *yes* the control directory will be removed and you will have a plain, unversioned directory tree.

#### 4.28. Reposicionar uma cópia de trabalho

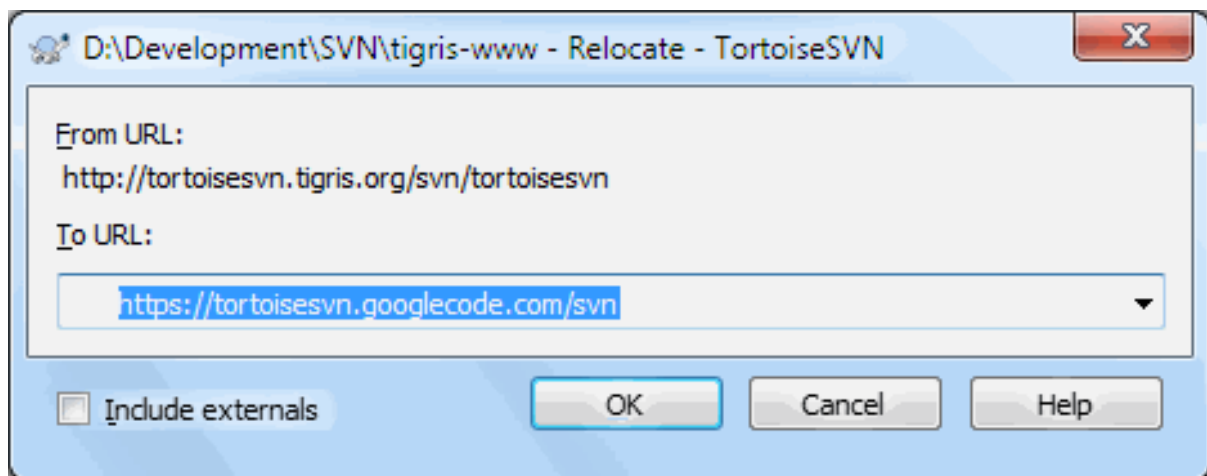


Figura 4.66. A Caixa de Diálogo Reposicionar

Se o teu repositório, por alguma razão, mudou a sua localização (IP/URL). Talvez estejas mesmo preso e não possas aprovar e não queres verificar a tua cópia funcional novamente a partir do novo local e para mover todos



os teus dados alterados de volta para a nova cópia funcional, TortoiseSVN → **Recolocar** é o comando que estás á procura. Ele basicamente faz muito pouco: ele reescreve todos os URLs que estão associados com cada arquivo e pasta com a nova URL.

### Nota

Esta operação só funciona nas *raízes* das cópias de trabalho . Pelo que a entrada do menu de contexto só aparecerá nas raízes das cópias de trabalho

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 4.20.3, “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.

## 4.29. 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:

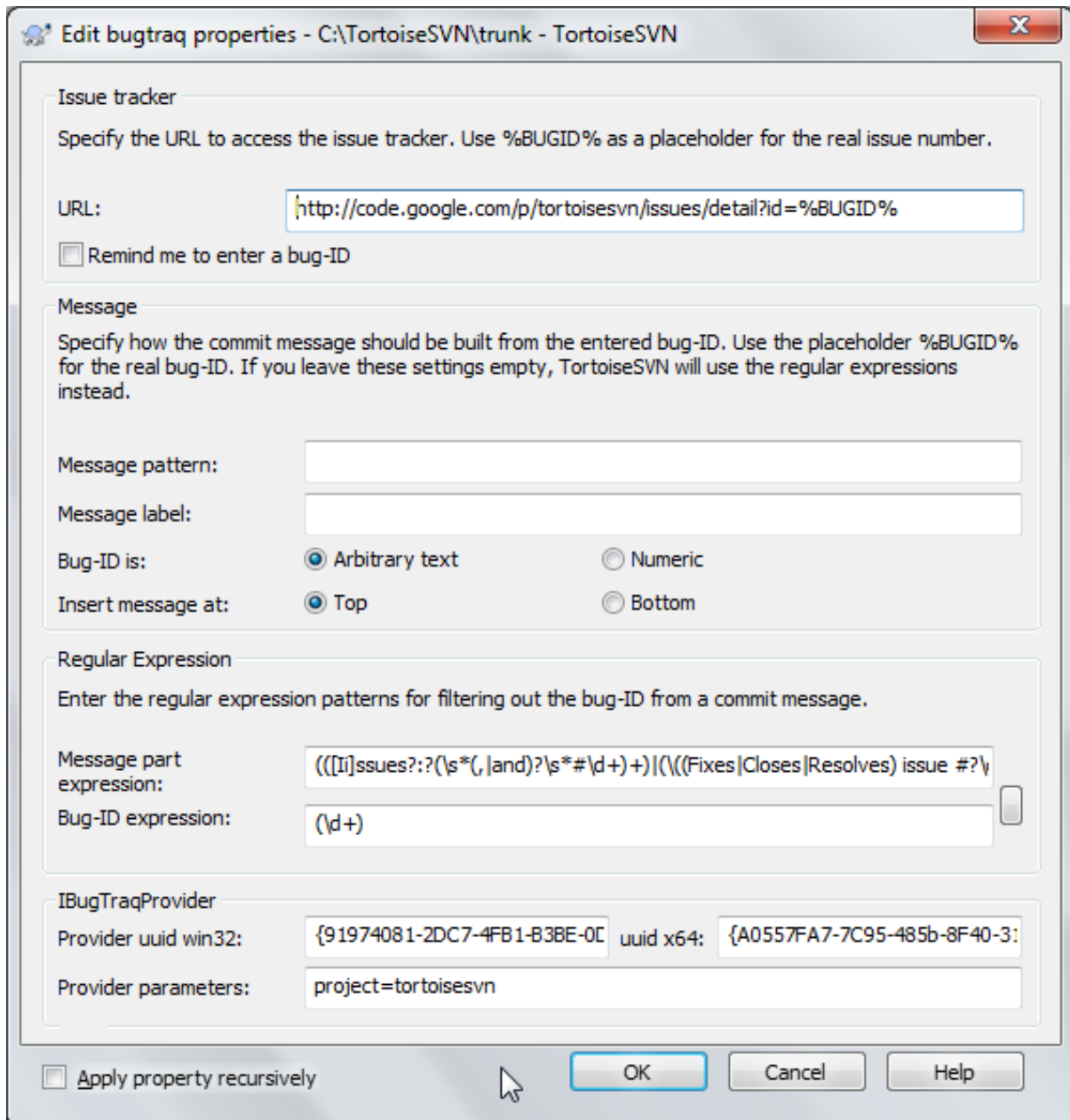
1. Quando o utilizador introduz a mensagem de registo, pode ser automaticamente adicionada uma linha bem definida, que incluí o número de problema associado com a submissão. Isto irá reduzir o risco de que o utilizador introduza um número de problema de uma maneira tal que, as ferramentas de identificação 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.

#### 4.29.1. 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 4.18, “Configurações de Projecto”)



**Figura 4.67. Caixa de diálogo de Propriedades do Bugtraq**

Quando editas qualquer uma das propriedades do bugtraq, um editor de propriedade especial será usado para tornar mais fácil definir os valores adequados.

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 para o URL da tua ferramenta de controlo de problemas. Deverá ser um URI devidamente 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 quando estiveres a olhar para o registo de revisões possas saltar directamente para a tua ferramenta de identificação de bugs. Não precisas de providenciar essa propriedade, mas então o TortoiseSVN apenas mostrará o número do problema e não a hiperligação para a tua ferramenta de controlo de problemas. E.g. o projecto TortoiseSVN está a usar o `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.*

#### 4.29.1.1. 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: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.*

**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.*

### 4.29.1.2. 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 colocadas 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 que desejes, em linguagem natural. 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 dentro da mensagem de registo, poderás 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.

If you are unfamiliar with regular expressions, take a look at the introduction at [https://en.wikipedia.org/wiki/Regular\\_expression](https://en.wikipedia.org/wiki/Regular_expression), and the online documentation and tutorial at <http://www.regular-expressions.info/>.

Não é sempre fácil obter a regex correcta, pelo que para dar ajudar, existe uma caixa de diálogo de teste embebida na caixa de diálogo de propriedades do bugtraq. Clica no botão à direita das caixas de edição para o mostrar. Aqui podes introduzir algum texto, e alterar cada regex para veres os resultados. Se a regex for inválida o fundo da caixa de edição muda para vermelho.

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 colocadas em pastas para o sistema funcionar. Quando submetes um ficheiro ou pasta, as propriedades são lidas dessa pasta. Se as propriedades não forem aí encontradas, o TortoiseSVN irá pesquisar ascendentemente na árvore de pastas de modo a encontrar até encontrar uma pasta não versionada, ou a raiz da árvore (eg. `C:\`). Se tens a certeza que cada utilizador só efectua checkout 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. Uma propriedade configurada mais profundamente na hierarquia do projecto, irá se sobrepor à configuração de níveis superiores (mais perto do `trunk/`).

As of version 1.8, TortoiseSVN and Subversion use so called `inherited properties`, which means a property that is set on a folder is automatically also implicitly set on all subfolders. So there's no need to set the properties on all folders anymore but only on the root folder.

Só para propriedades de projecto, i.e. `tsvn:`, `bugtraq:` e `webviewer:` podes usar a caixa de verificação **Recursiva** para fixar a propriedade em todas as subpastas na hierarquia, sem também fixa-la em todos os ficheiros.

Quando adicionas novas subpastas a uma cópia de trabalho, usando o TortoiseSVN, quaisquer propriedades de projecto presentes na pasta pai serão automaticamente adicionadas à nova pasta filho também.



## 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 unless you started the repo browser from your working copy. If you started the repo browser by entering the URL of the repository you won't see this feature.

Pela mesma razão, as propriedades de projecto não serão propagadas automaticamente quando uma pasta filho é adicionada usando o navegador de repositório.

This issue tracker integration is not restricted to TortoiseSVN; it can be used with any Subversion client. For more information, read the full *Issue Tracker Integration Specification* [<https://svn.code.sf.net/p/tortoisesvn/code/trunk/doc/notes/issuetrackers.txt>] in the TortoiseSVN source repository. (Secção 3, "Licença" explains how to access the repository.)

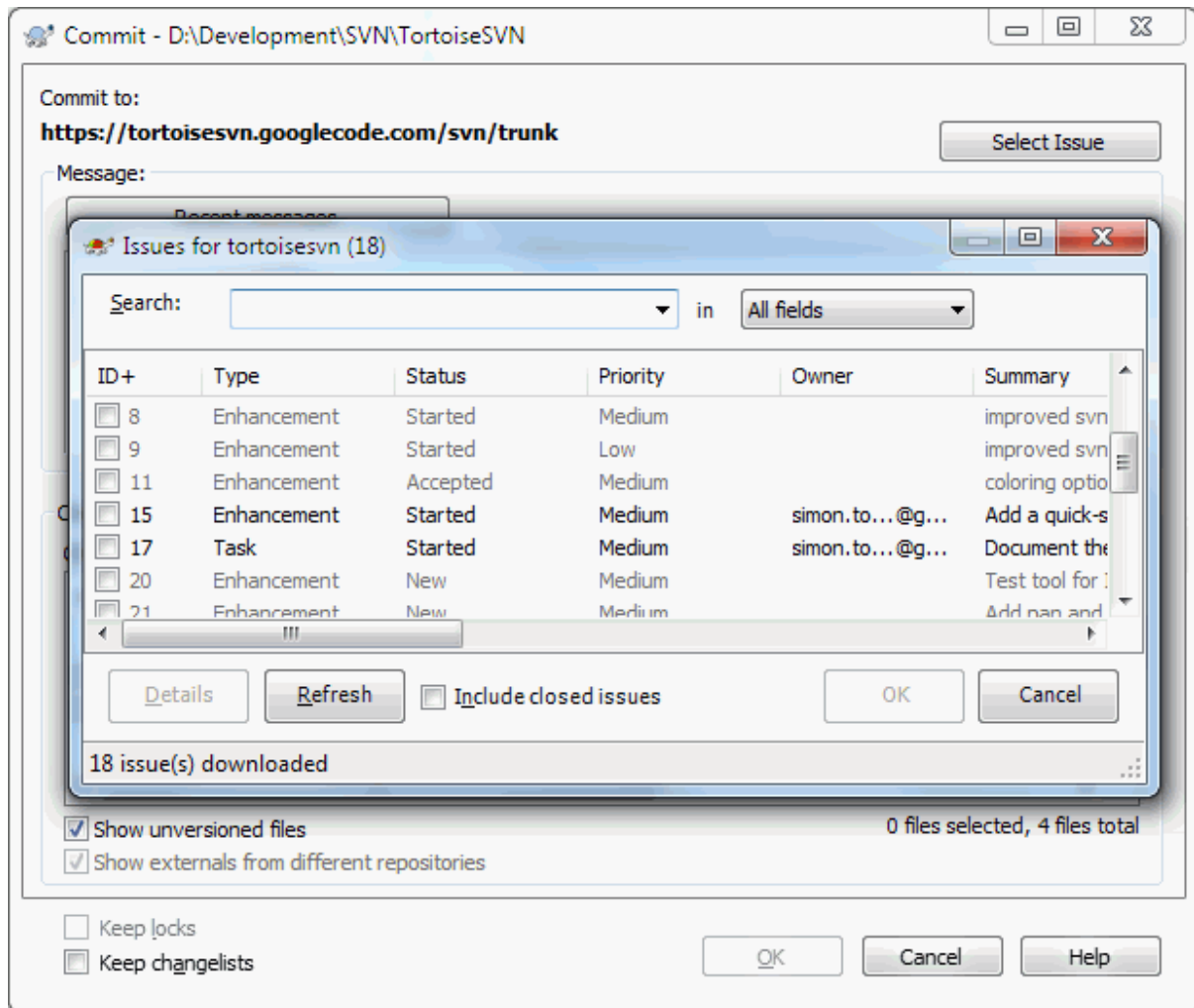
### 4.29.2. Obter Informações do Gestor de Problemas

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.

Any such interface is of course highly specific to your issue tracker system, so we cannot provide this part, and describing how to create such a program is beyond the scope of this manual. The interface definition and sample

plugins in C# and C++/ATL can be obtained from the contrib folder in the *TortoiseSVN repository* [https://svn.code.sf.net/p/tortoisesvn/code/trunk/contrib/issue-tracker-plugins]. (Secção 3, “Licença” explains how to access the repository.) A summary of the API is also given in *Capítulo 7, Interface IBugtraqProvider*. Another (working) example plugin in C# is *Gurtle* [http://code.google.com/p/gurtle/] which implements the required COM interface to interact with the *Google Code* [http://code.google.com/hosting/] issue tracker.

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 4.68. 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.

### 4.30. 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/] e o *WebSVN* [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 4.18, “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 da revisão em questão. Isto permite ao TortoiseSVN mostrar uma entrada no menu de contexto na caixa de diálogo registo **Menu de Contexto** → **Ver revisão no visualizador web**.

#### webviewer:pathrevision

Configura esta propriedade com o URL do teu visualizador de repositório, para veres as alterações num ficheiro específico, numa dada revisão. Este deverá ser um URI devidamente codificado e terá de conter %REVISION% e %PATH%. O %PATH% será substituído pelo caminho relativo para a raiz do repositório. Isto permite ao TortoiseSVN mostrar uma entrada no menu de contexto da caixa de diálogo de registo, **Menu de contexto** → **Ver revisão do caminho no visualizador web**. Por exemplo, se clicares com o botão direito no painel de fundo da caixa de diálogo de registo, numa entrada de ficheiro `/trunk/src/file o%PATH%` no URL será substituído por `/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 4.29, “Integração com Sistemas de controlo de problemas/Gestores de Problemas”](#).



### Configurar Propriedades em Pastas

Essas propriedades devem ser colocadas em pastas para o sistema funcionar. Quando submetes um ficheiro ou pasta, as propriedades são lidas dessa pasta. Se as propriedades não forem aí encontradas, o TortoiseSVN irá pesquisar ascendente na árvore de pastas de modo a encontrar até encontrar uma pasta não versionada, ou a raiz da árvore (eg. `C:\`). Se tens a certeza que cada utilizador só efectua checkout 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. Uma propriedade configurada mais profundamente na hierarquia do projecto, irá se sobrepor à configuração de níveis superiores (mais perto do `trunk/`).

Só para propriedades de projecto, i.e. `tsvn:`, `bugtraq:` e `webviewer:` podes usar a caixa de verificação **Recursiva** para fixar a propriedade em todas as subpastas na hierarquia, sem também fixa-la em todos os ficheiros.

Quando adicionas novas subpastas a uma cópia de trabalho, usando o TortoiseSVN, quaisquer propriedades de projecto presentes na pasta pai serão automaticamente adicionadas à nova pasta filho também.



### Limitações no Uso do Navegador de Repositório

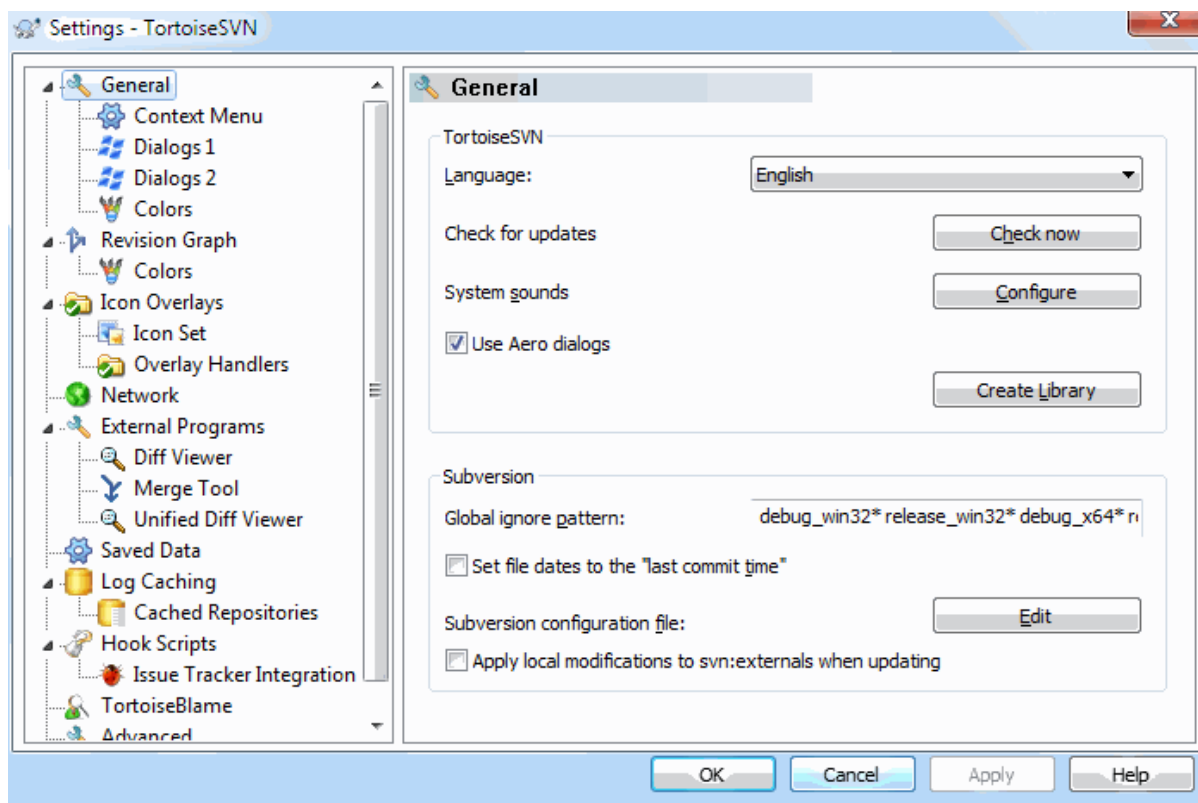
Because the repo viewer integration depends upon accessing Subversion properties, you will only see the results when using a checked out working copy. Fetching properties remotely is a slow operation, so you will not see this feature in action from the repo browser unless you started the repo browser from your working copy. If you started the repo browser by entering the URL of the repository you won't see this feature.

Pela mesma razão, as propriedades de projecto não serão propagadas automaticamente quando uma pasta filho é adicionada usando o navegador de repositório.

## 4.31. 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.

### 4.31.1. Preferências Gerais



**Figura 4.69. 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

Selects your user interface language. Of course, you have to install the corresponding language pack first to get another UI language than the default English one.

#### Verificar actualizações

O TortoiseSVN irá contactar o seu sítio de descarga periodicamente, para verificar se está disponível uma versão mais recente do programa. Se existir, irá mostrar uma hiperligação de notificação na caixa de diálogo submeter. Usa **Verificar agora** se queres uma resposta imediata. A nova versão não será descarregada de imediato; receberás simplesmente uma caixa de diálogo de informação a te dizer que está 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.

#### Usar caixas de diálogo Aero

No Windows Vista e sistemas posteriores, isto controla se as caixas de diálogo usam o estilo Aero.

#### Criar Biblioteca

No Windows 7 podes criar uma biblioteca na qual agrupas cópias de trabalho que estão espalhadas por vários locais do teu sistema.



## Padrão global para ignorar arquivos

Padrões globais para ignorar são usados para evitar que ficheiros não versionados apareçam, por exemplo, na caixa de diálogo `submit`. 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 4.14.1, “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 configures o padrão `global-ignores`, ele ir-se-á sobrepor à preferência que efectuaste aqui. O ficheiro de configuração do Subversion pode ser acedido usando o `Editar`, como aqui descrito.

This ignore pattern will affect all your projects. It is not versioned, so it will not affect other users. By contrast you can also use the versioned `svn:ignore` or `svn:global-ignores` property to exclude files or directories from version control. Read [Secção 4.14, “Ignorando Ficheiros E Pastas”](#) for more information.

### 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.

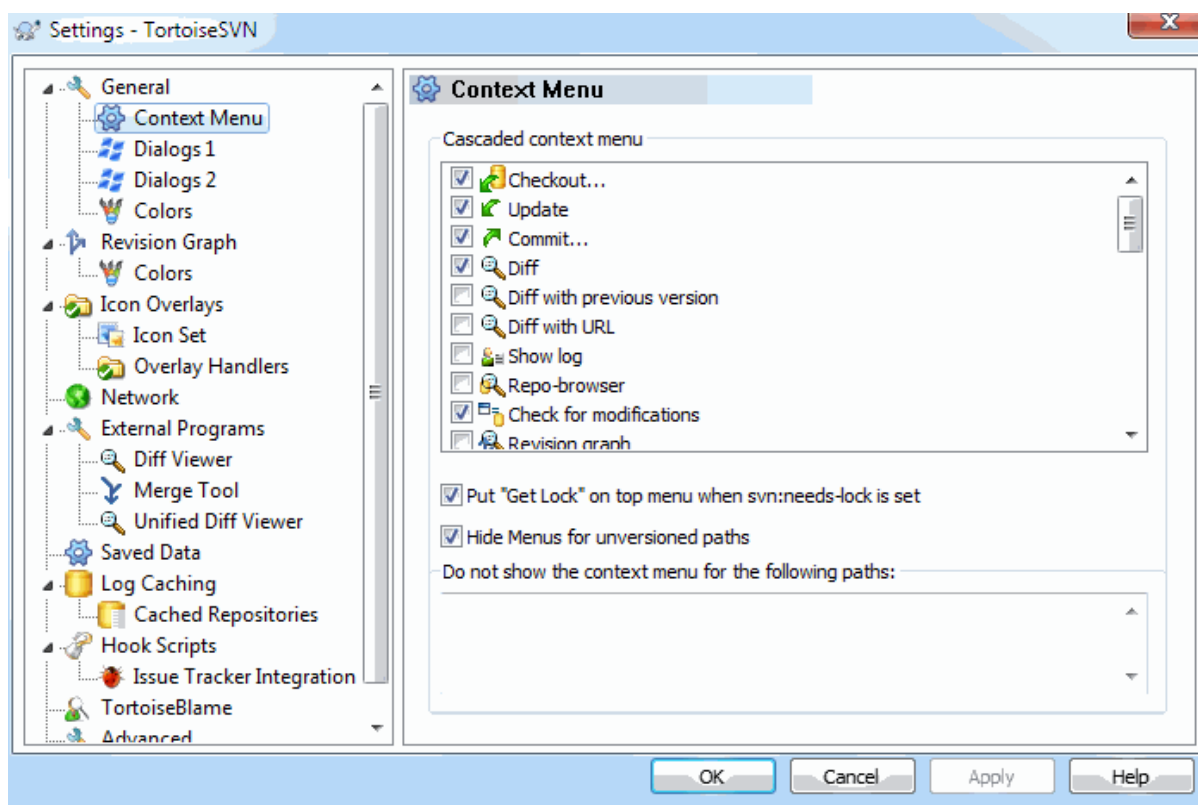
### Ficheiro de configuração do Subversion

Use `Edit` to edit the Subversion configuration file directly. Some settings cannot be modified directly by TortoiseSVN, and need to be set here instead. For more information about the Subversion config file see the [Runtime Configuration Area](http://svnbook.red-bean.com/en/1.8/svn.advanced.confarea.html) [http://svnbook.red-bean.com/en/1.8/svn.advanced.confarea.html]. The section on [Automatic Property Setting](http://svnbook.red-bean.com/en/1.8/svn.advanced.props.html#svn.advanced.props.auto) [http://svnbook.red-bean.com/en/1.8/svn.advanced.props.html#svn.advanced.props.auto] is of particular interest, and that is configured here. Note that Subversion can read configuration information from several places, and you need to know which one takes priority. Refer to [Configuration and the Windows Registry](http://svnbook.red-bean.com/en/1.8/svn.advanced.confarea.html#svn.advanced.confarea.windows-registry) [http://svnbook.red-bean.com/en/1.8/svn.advanced.confarea.html#svn.advanced.confarea.windows-registry] to find out more.

### Ao actualizar aplica alterações locais ao `svn:externals`

Esta opção diz ao TortoiseSVN para aplicar sempre alterações locais à propriedade `svn:externals` ao actualizar a cópia de trabalho.

### 4.31.1.1. Menu de Contexto preferências



**Figura 4.70. A Caixa de Diálogo Preferências, Página Menu de Contexto**

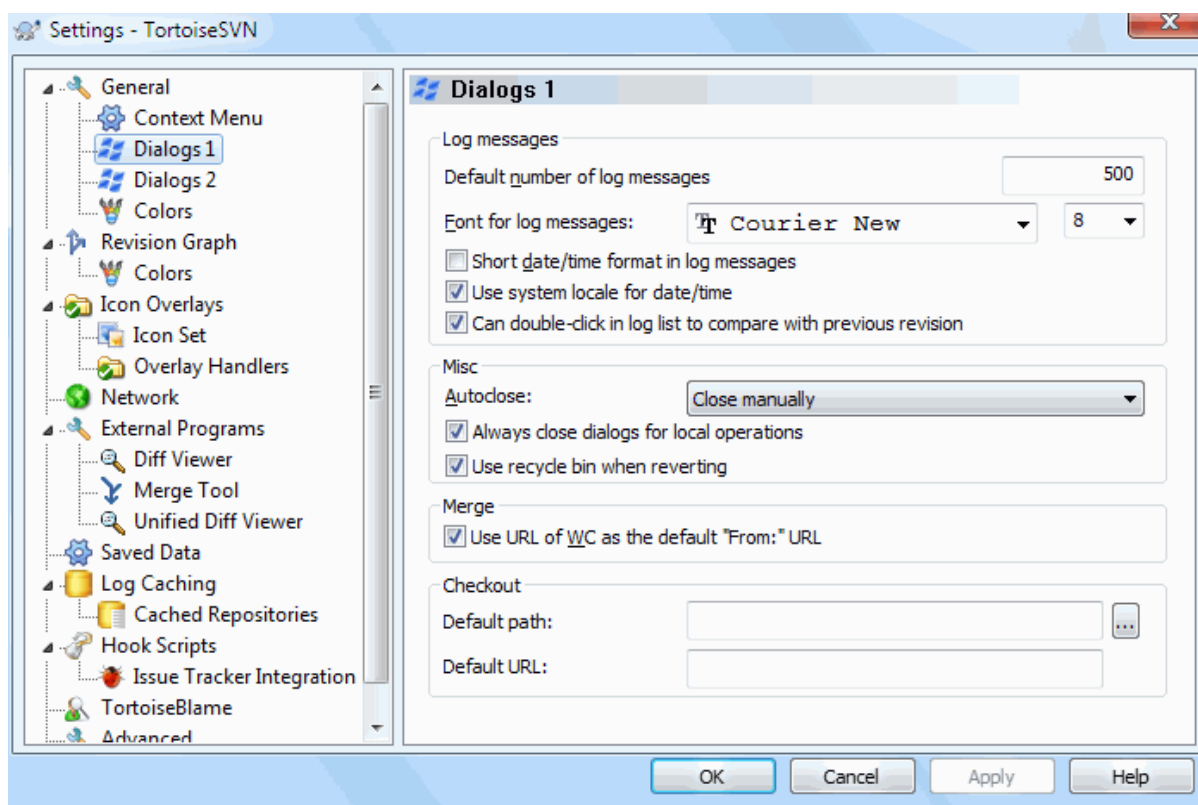
Esta página permite-te especificar, quais das entradas do menu de contexto do TortoiseSVN, que irão aparecer no menu de contexto principal, e quais aparecerão no submenu do TortoiseSVN. Por defeito a maioria dos itens estão desseleccionados e aparecem no submenu.

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.

Most of the time, you won't need the TortoiseSVN context menu, apart for folders that are under version control by Subversion. For non- versioned folders, you only really need the context menu when you want to do a checkout. If you check the option `Hide menus for unversioned paths`, TortoiseSVN will not add its entries to the context menu for unversioned folders. But the entries are added for all items and paths in a versioned folder. And you can get the entries back for unversioned folders by holding the **Shift** key down while showing the context menu.

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.

### 4.31.1.2. Preferências das Caixas de Diálogo TortoiseSVN 1



**Figura 4.71. 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.

Podes efectuar duplo clique na lista de mensagens de registo, para comparar com a 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 apanhar as diferenças é normalmente um processo moroso, e a maior parte das pessoas prefere evitar esperar após um duplo-clique accidental, pelo que é essa a razão por que esta opção não está activa por defeito.

Auto-fecho

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 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.

Fecha sempre as caixas de diálogo em operações locais

Operações locais como adicionar ficheiros ou reverter alterações não necessitam de contactar o repositório e são concluídas rapidamente, pelo que a caixa de diálogo de progresso é normalmente de pouco interesse. Selecciona esta opção se queres que a caixa de diálogo de progresso se feche automaticamente após essas operações, a não ser que existam erros.

Usar a reciclagem quando a reverter

Quando revertes 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.

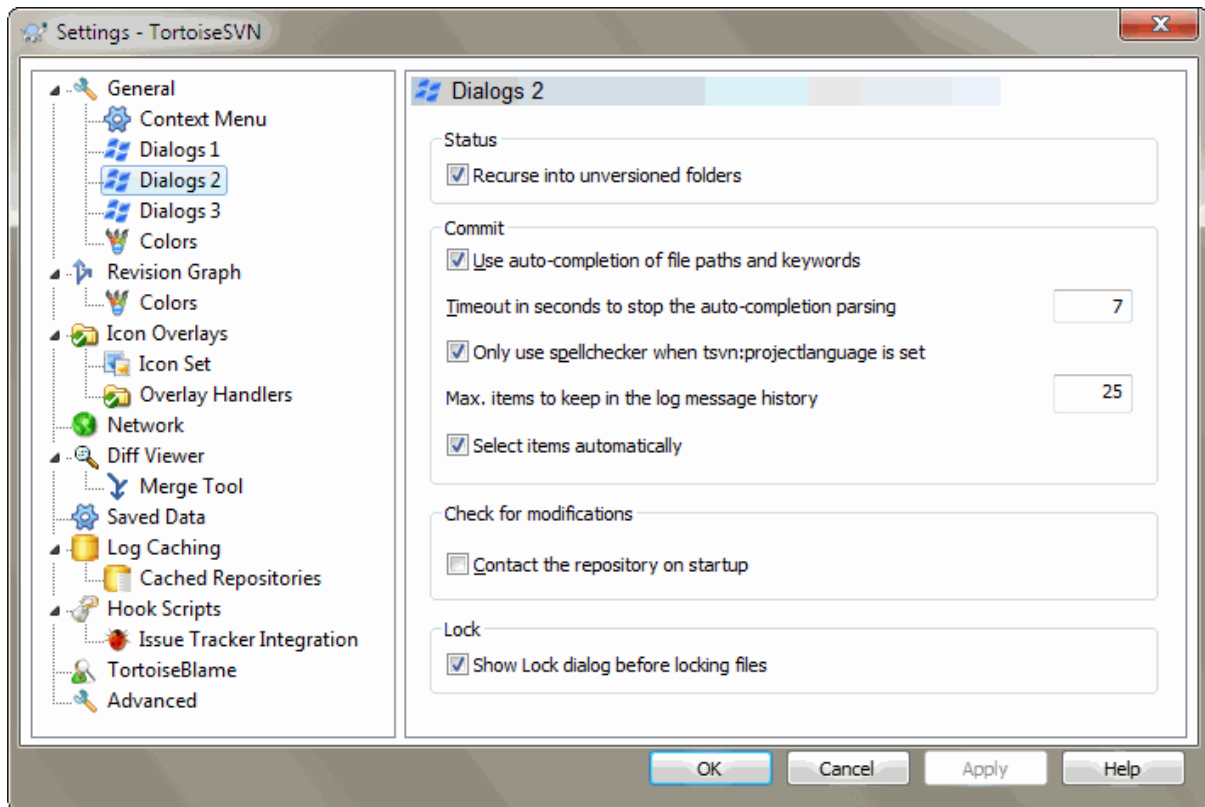
Caminho por defeito para colonizar

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.

### 4.31.1.3. Preferências das Caixas de Diálogo TortoiseSVN 2



**Figura 4.72. 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.

Na caixa de diálogo Verificar alterações tu podes optar por ver os itens ignorados. Se esta caixa estiver verificada quando uma pasta ignorada for encontrada, todos os itens filhos serão também mostrados.

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.

**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.

**Reabrir caixa diálogo após submeter, se foram deixados itens não submetidos**

This reopens the commit dialog automatically at the same directory after a successful commit. The dialog is reopened only if there still are items left to commit.

**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.

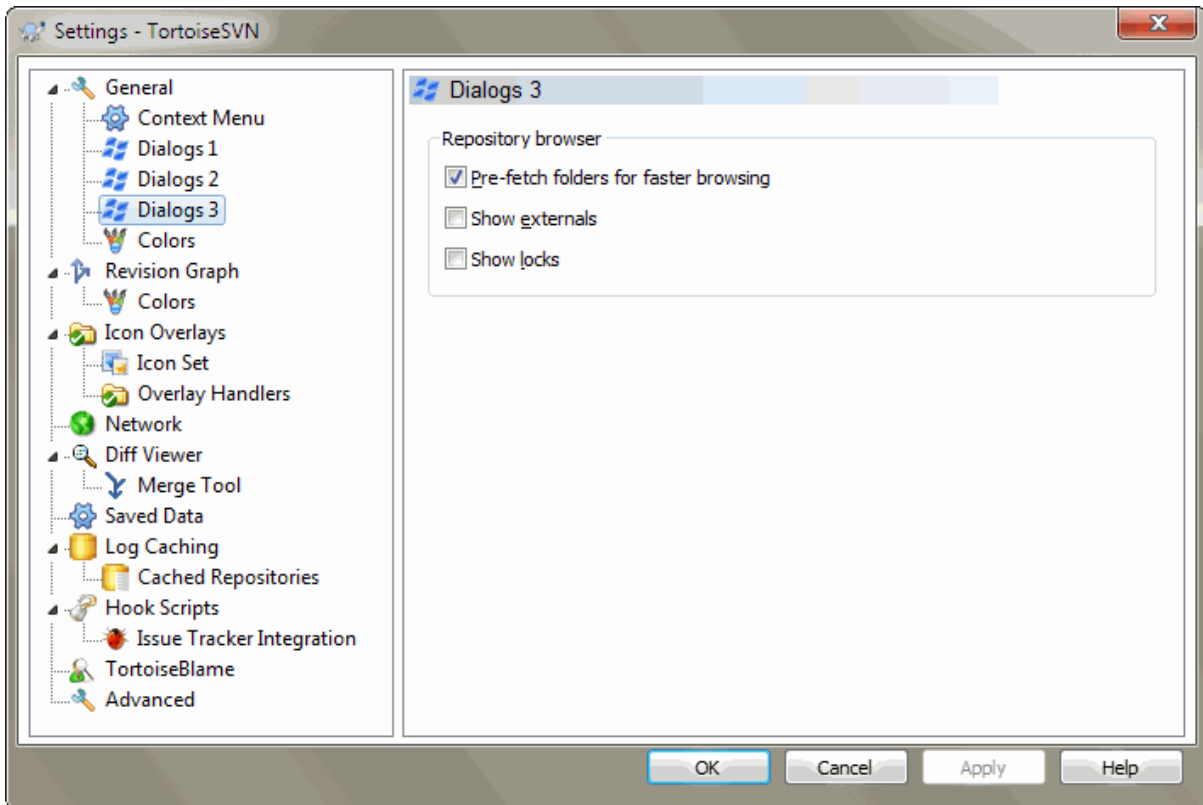
**Mostrar caixa de diálogo de Bloqueio antes de bloquear arquivos**

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.

**4.31.1.4. Caixa de diálogo de preferências 3 do TortoiseSVN**



**Figura 4.73. Caixa de diálogo de preferências, página de diálogos 3**

**Pré-colecta de pastas para navegação rápida**

Se esta caixa estiver verificada (estado por defeito), o navegador de repositório recolherá informação sobre as pastas visualizadas no plano de fundo. Deste modo, assim que navegares numa dessas pastas, a informação já estará assim disponível.

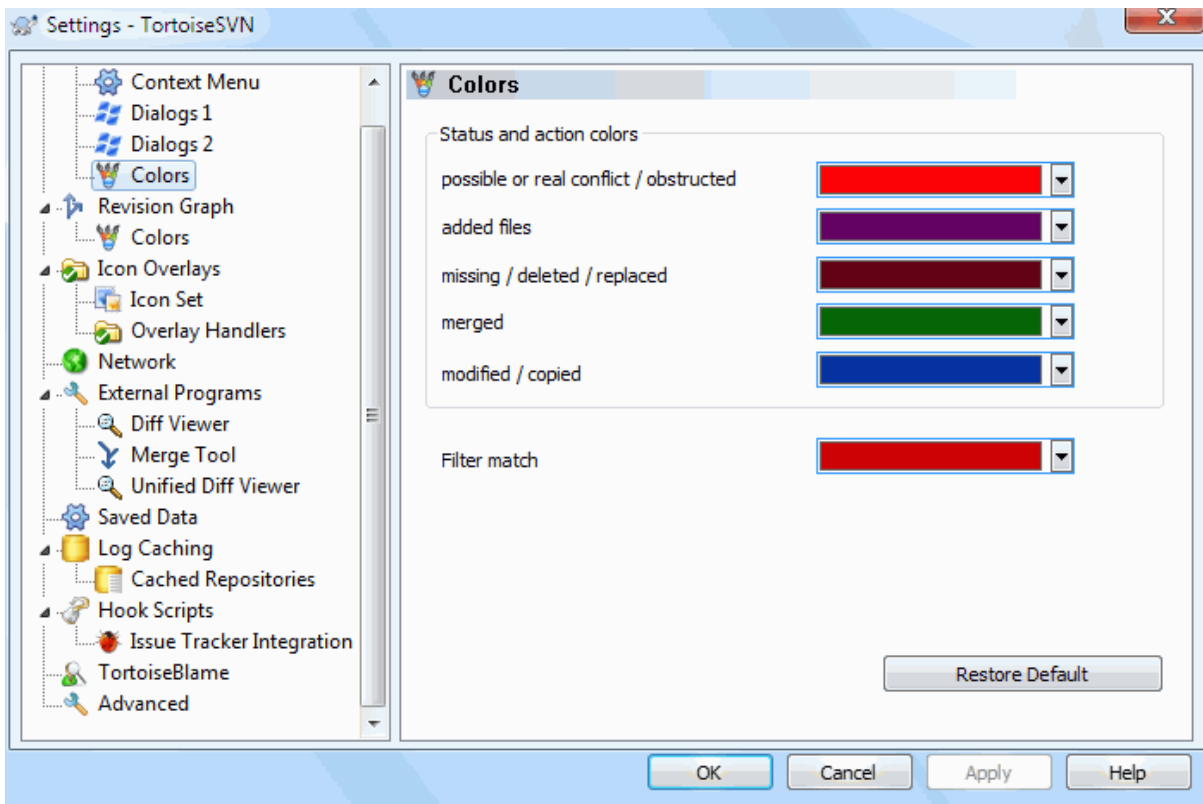
No entanto, alguns servidores não conseguem lidar com pedidos múltiplos que isto provoca, ou quando não configurado correctamente trata tantos pedidos como algo nefasto e começa a bloqueá-los. Neste caso poderás desactivar a pré-colecta aqui.

**Mostrar externos**

Se esta caixa estiver verificada (estado por defeito), o navegador de repositório mostra os ficheiros

Tal como a funcionalidade da pré-colecta exposta acima, esta também pode colocar muito esforço em servidores fracos. Neste caso pode desactivar esta funcionalidade aqui.

**4.31.1.5. Preferências de cor do TortoiseSVN**



**Figura 4.74. 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.

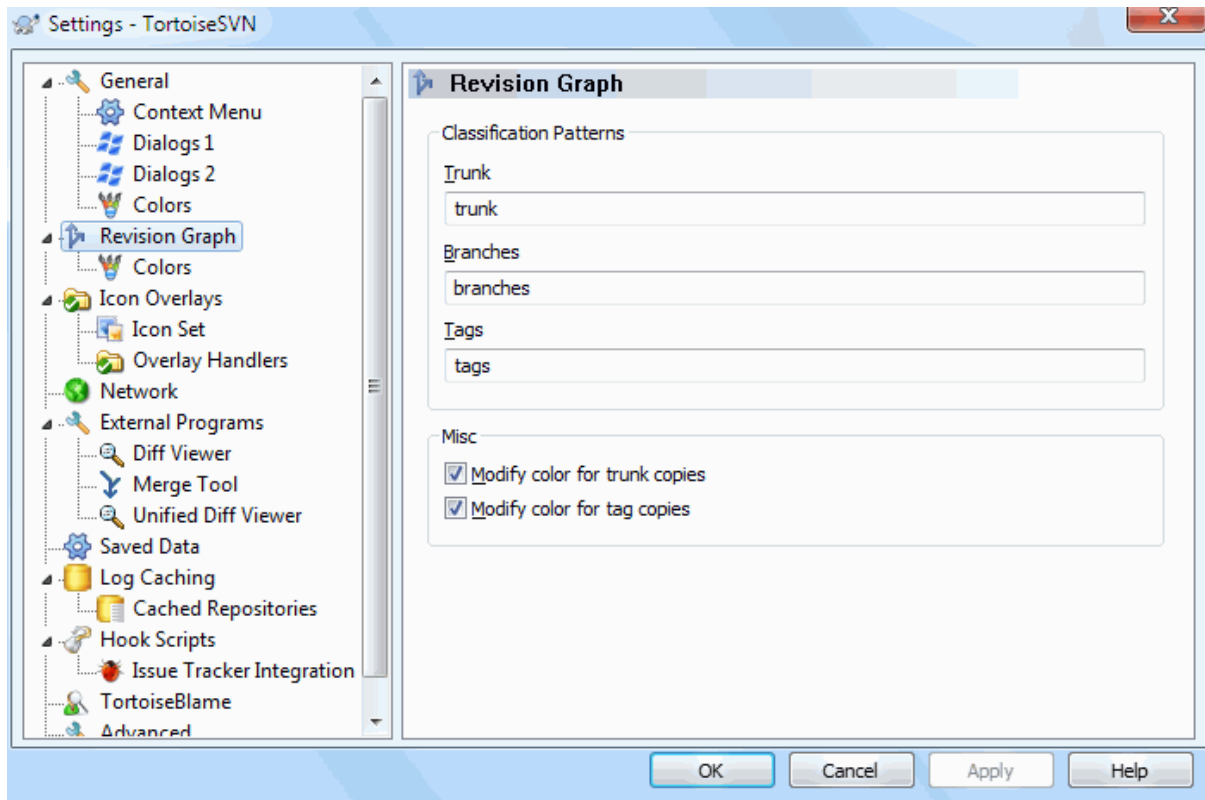
Nó substituído

O item original foi removido e um novo item com o mesmo nome substituído-o.

Correspondência de filtro

Ao usar filtragens na caixa de diálogo de registo, os termos de pesquisa são destacados nos resultados com recurso a esta cor.

### 4.31.2. Preferências do Gráfico de Revisões



**Figura 4.75. 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.



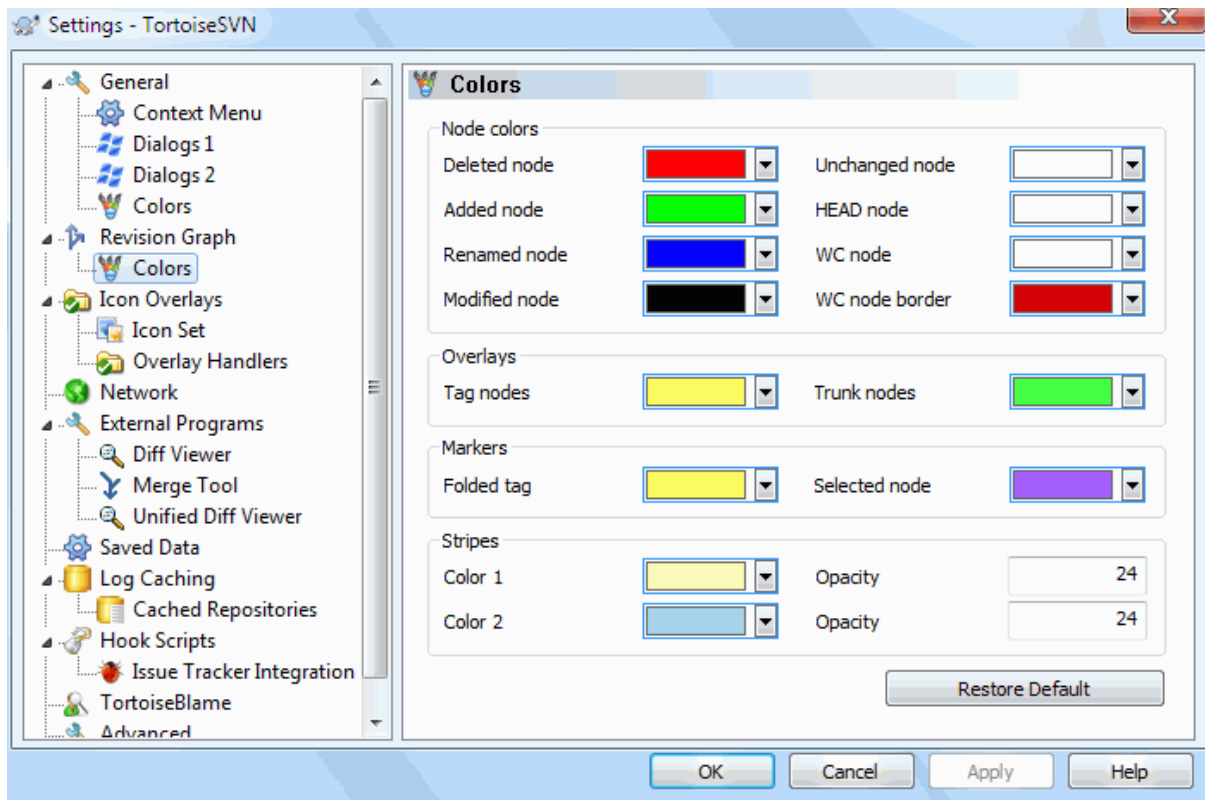
### Commit tag detection

Please note that these patterns are also used to detect commits to a tag, not just for the revision graph.

#### 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.

#### 4.31.2.1. Cores do Gráfico de Revisões



**Figura 4.76. 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.

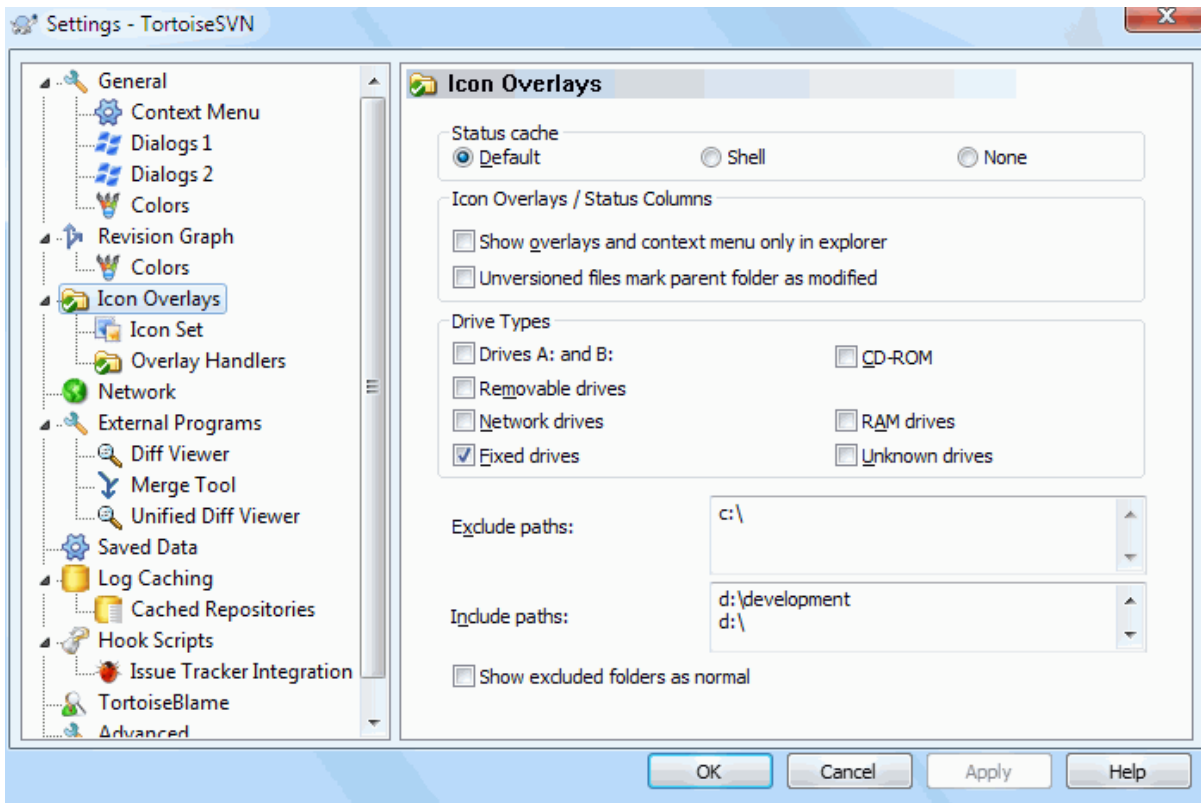
#### Marcadores de Nó Seleccionado

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.

### 4.31.3. Preferências de Sobreposição de Ícones



**Figura 4.77. 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.

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".

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**.

You can force the status cache to *None* for elevated processes by checking the **Disable status cache for elevated processes** box. This is useful if you want to prevent another TSVNCache.exe process getting created with elevated privileges.

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*.

If you have files in the `ignore-on-commit` changelist, you can chose to make those files not propagate their status to the parent folder. That way if only files in that changelist are modified, the parent folder still shows the unmodified overlay icon.

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 quais os caminhos para os quais *não* deverá ser mostrado sobreposições de ícones e colunas de estado. Isto é útil se tens algumas cópias de trabalho muito grandes, contendo apenas bibliotecas que não irás alterar e por isso não irão necessitar de sobreposições, ou se apenas queres que o TortoiseSVN olhe para pastas específicas.

Qualquer caminho que especifiques aqui é assumido que seja de aplicação recursiva, pelo que nenhuma das pastas filho irá também mostrar sobreposições. Se queres excluir *apenas* a pasta nomeada, adiciona ? após o 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 três preferências interagem. Para qualquer caminho dado verifica as listas de inclusão e exclusão, procurando ascendentemente através da estrutura de pastas até ser encontrada uma correspondência. Quando for encontrada a primeira correspondência, obedece a essa regra de inclusão ou exclusão. Se existir um conflito, a especificação da pasta simples toma precedência sobre a especificação recursiva, em seguida a inclusão toma precedência sobre a exclusão.

An example will help here:

```
Exclude:
C:
C:\develop\?
C:\develop\tsvn\obj
C:\develop\tsvn\bin
```

```
Include:
C:\develop
```

These settings disable icon overlays for the C: drive, except for `c:\develop`. All projects below that directory will show overlays, except the `c:\develop` folder itself, which is specifically ignored. The high-churn binary folders are also excluded.

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

It is often convenient to use a SUBST drive to access your working copies, e.g. using the command

```
subst T: C:\TortoiseSVN\trunk\doc
```

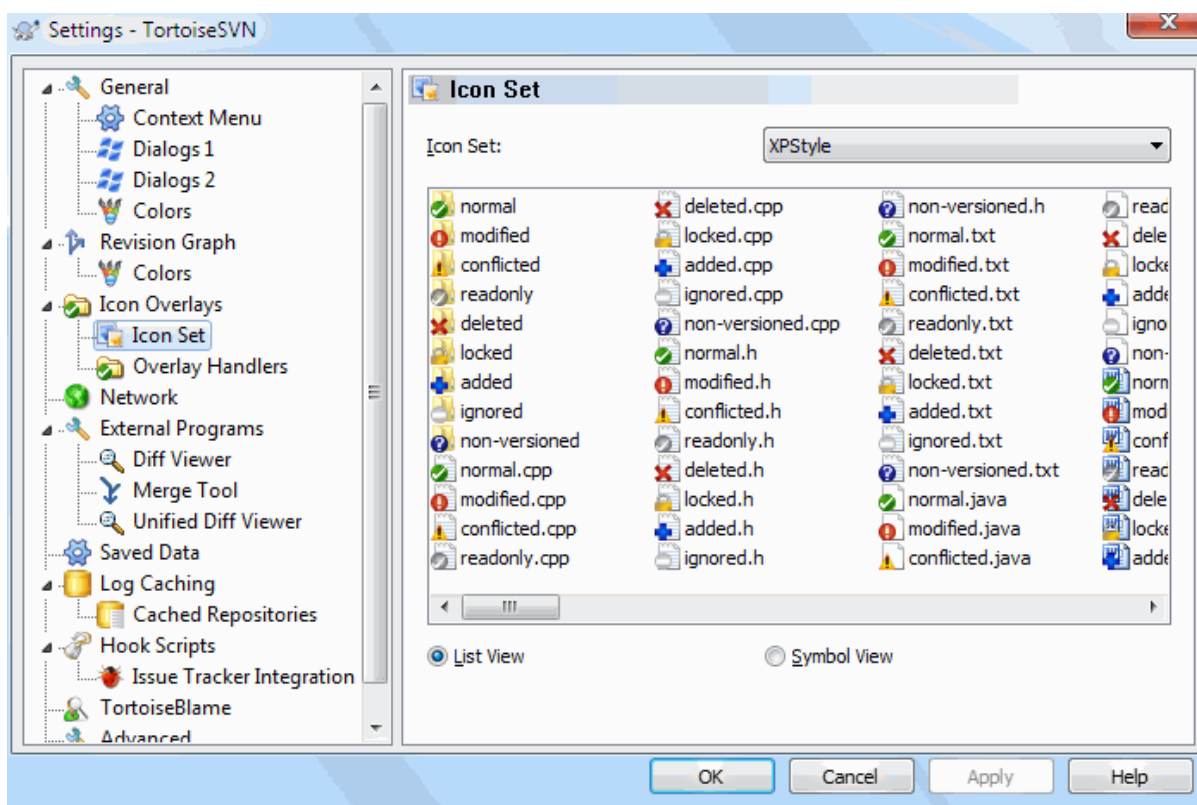
However this can cause the overlays not to update, as TSVNCache will only receive one notification when a file changes, and that is normally for the original path. This means that your overlays on the subst path may never be updated.

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 poderás excluir áreas que contêm cópias de trabalho, o que poupa o TSVNCache de efectuar a pesquisa e monitorização de alterações, mantendo a indicação visual de que a pasta contém uma cópia de trabalho. A caixa de verificação **Mostrar pastas raiz excluídas como 'normal'** permite-te fazer isso. Com esta opção, as pastas raiz de cópias de trabalho, em qualquer área excluída (tipo de navegação não verificado, ou especificamente excluído) serão mostradas como normais e actualizadas, com a marca verde de verificação. Isto lembrar-te-á que estás a olhar para uma cópia de trabalho, mesmo que os ícones de sobreposição das pastas não estejam correctos. Os ficheiros não terão nenhuns ícones de sobreposição. Ter em atenção que os menus de contexto funcionaram na mesma, mesmo que não sejam mostrados os ícones de sobreposição.

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.

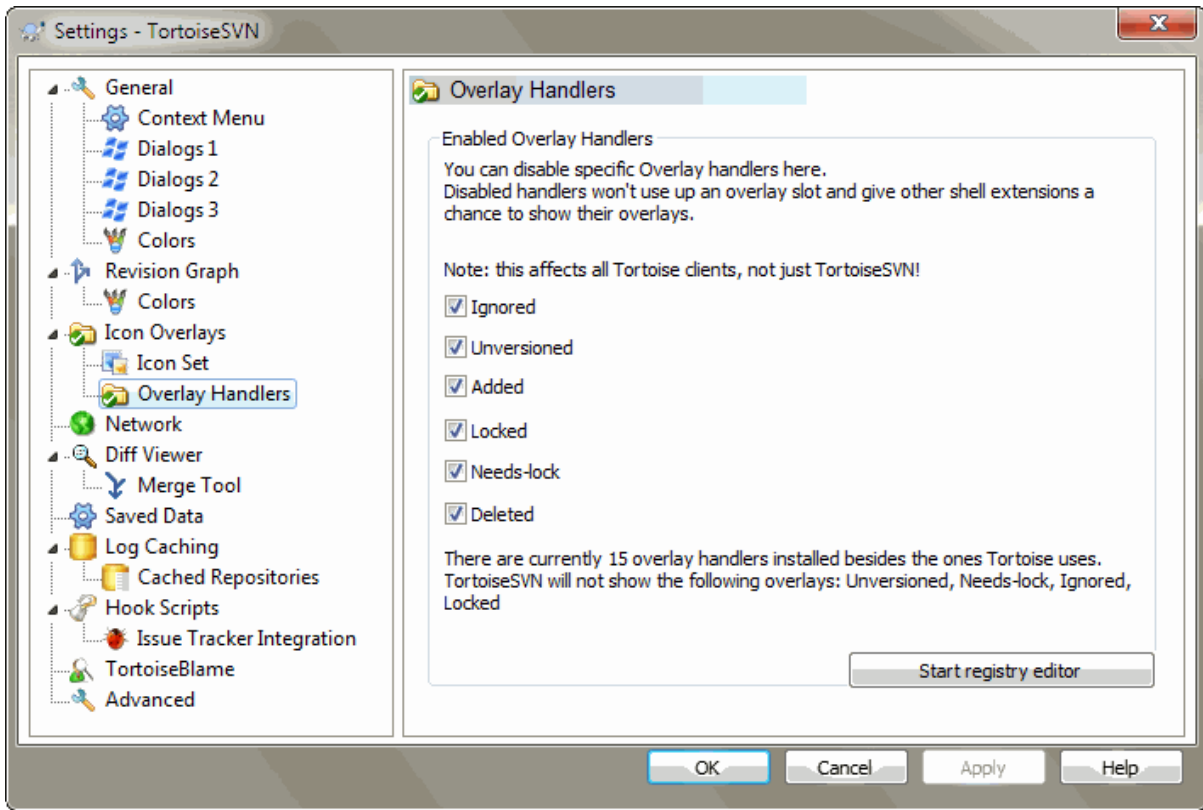
### 4.31.3.1. Seleção de Conjunto de Ícones



**Figura 4.78. 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.

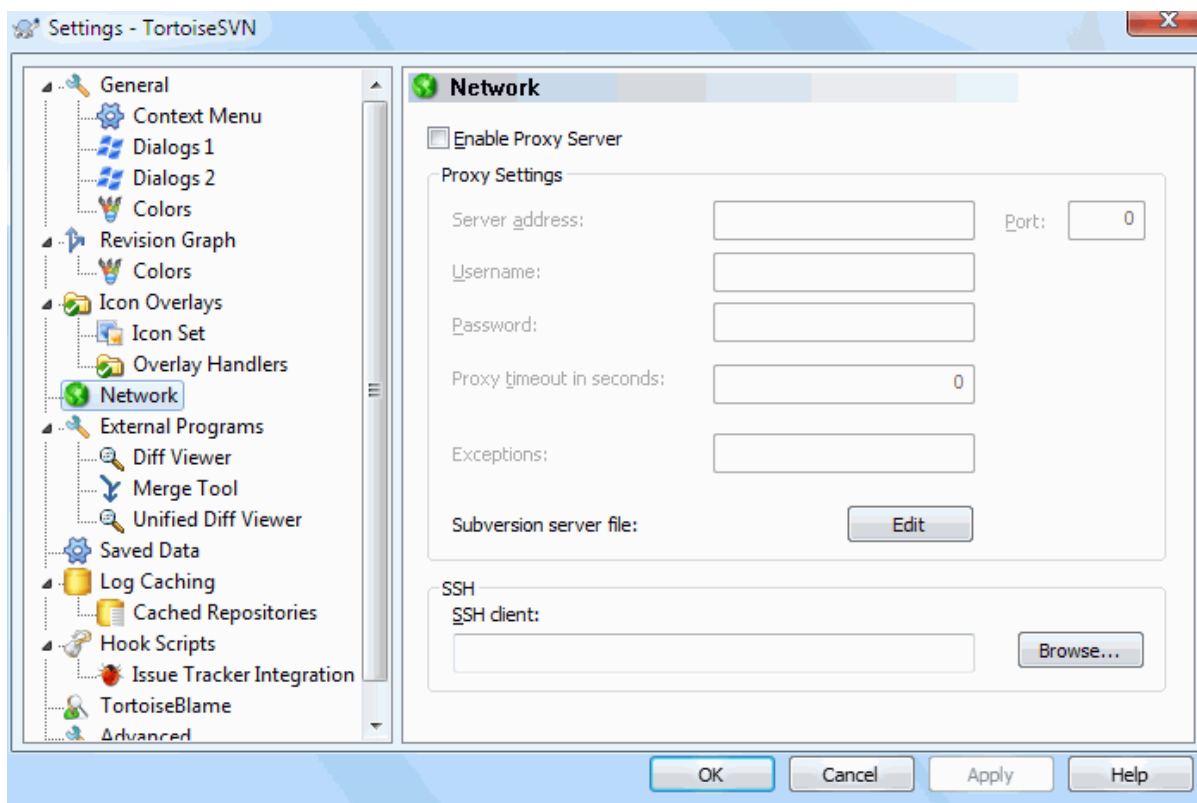
### 4.31.3.2. Activados manípulos de Sobreposição



**Figura 4.79. A Caixa de Diálogo preferências, Página de Manípulos de Ícones**

Porque o número de sobreposições disponíveis é severamente restrito, podes optar por desactivar alguns manipuladores para garantir que o que tu quiseres será carregado. O TortoiseSVN porque usa o componente TortoiseOverlays comum que é compartilhado com os outros clientes do Tortoise (por exemplo o TortoiseCVS e o TortoiseHg), esta configuração também irá afectar os clientes.

### 4.31.4. Preferências de Rede



**Figura 4.80. 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.

If you need to set up per-repository proxy settings, you will need to use the Subversion `servers` file to configure this. Use `Edit` to get there directly. Consult the [Runtime Configuration Area](http://svnbook.red-bean.com/en/1.8/svn.advanced.confarea.html) [http://svnbook.red-bean.com/en/1.8/svn.advanced.confarea.html] for details on how to use this file.

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.

You must specify the full path to the executable. For `TortoisePlink.exe` this is the standard TortoiseSVN bin directory. Use the `BROWSE` button to help locate it. Note that if the path contains spaces, you must enclose it in quotes, e.g.

```
"C:\Program Files\TortoiseSVN\bin\TortoisePlink.exe"
```

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.

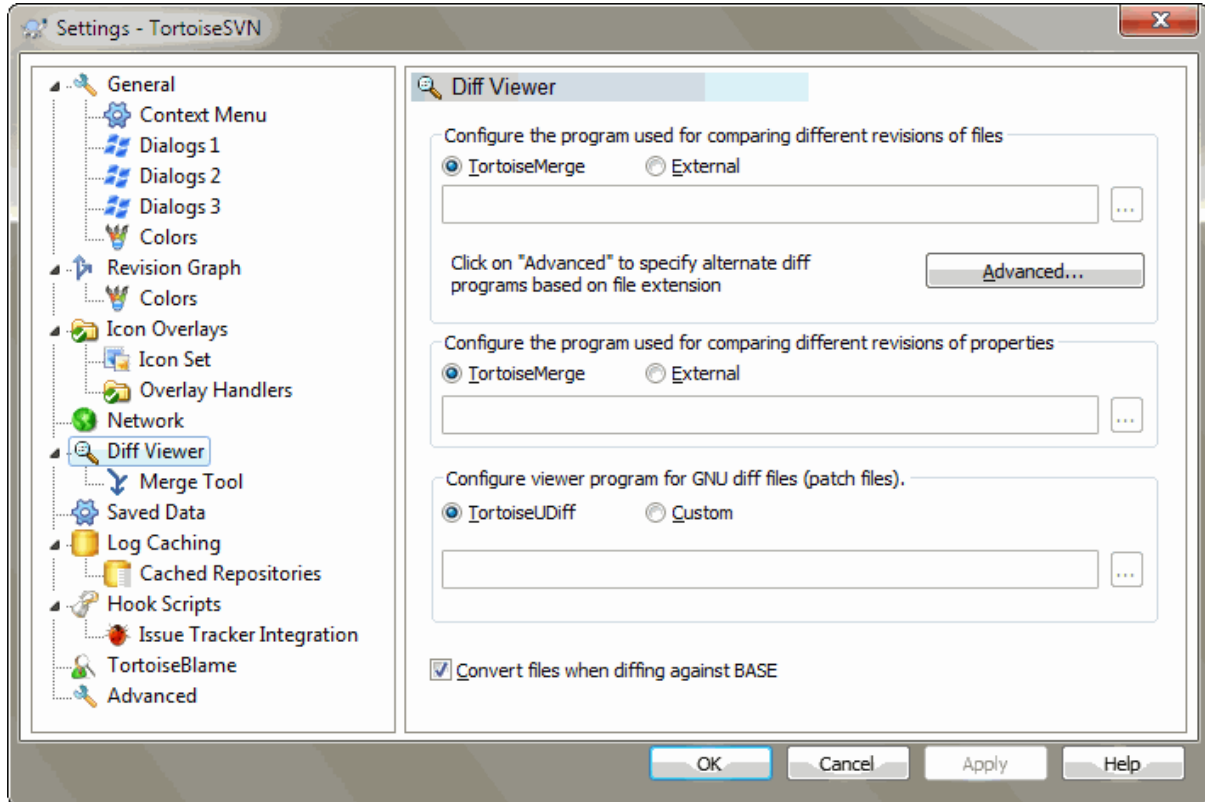
`TortoisePlink` does not have any documentation of its own because it is just a minor variant of `Plink`. Find out about command line parameters from the [PuTTY website](https://www.chiark.greenend.org.uk/~sgtatham/putty/) [https://www.chiark.greenend.org.uk/~sgtatham/putty/].

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`.



Finally, setting up SSH on server and clients is a non-trivial process which is beyond the scope of this help file. However, you can find a guide in the TortoiseSVN FAQ listed under [Subversion/TortoiseSVN SSH How-To](https://tortoisesvn.net/ssh_howto.html) [https://tortoisesvn.net/ssh\_howto.html].

### 4.31.5. Preferências de Programas Externos



**Figura 4.81. 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 4.11.6, “Ferramentas de Comparação/Integração”](#), para uma lista de alguns programas de Comparação/Integração que as pessoas podem usar com o TortoiseSVN.

#### 4.31.5.1. 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

%nqbasename

The window title for the base file, without quotes

%mine

O teu ficheiro, com as tuas alterações.

`%yname`  
O título da janela do teu ficheiro

`%nqyname`  
The window title for your file, without quotes

`%burl`  
O URL do ficheiro original, se disponível

`%nqburl`  
The URL of the original file, if available, without quotes

`%yurl`  
O URL do segundo ficheiro, se disponível

`%nqyurl`  
The URL of the second file, if available, without quotes

`%brev`  
A revisão do ficheiro original, se disponível

`%nqbrev`  
The revision of the original file, if available, without quotes

`%yrev`  
A revisão do segundo ficheiro, se disponível

`%nqyrev`  
The revision of the second file, if available, without quotes

`%peg`  
A revisão cavilha, se disponível

`%nqpeg`  
The peg revision, if available, without quotes

`%fname`  
The name of the file. This is an empty string if two different files are diffed instead of two states of the same file.

`%nqfname`  
The name of the file, without quotes

Os títulos das janelas não são nomes de ficheiros puros. O TortoiseSVN trata-os como nomes a mostrar, e cria os nomes de acordo. Pelo que e.g., se efectuares 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 .

For example, with ExamDiff Pro:

```
C:\Path-To\ExamDiff.exe %base %mine --left_display_name:%bname
                                --right_display_name:%yname
```

or with KDiff3:

```
C:\Path-To\kdiff3.exe %base %mine --L1 %bname --L2 %yname
```

or with WinMerge:

```
C:\Path-To\WinMerge.exe -e -ub -dl %bname -dr %ynname %base %mine
```

or with Araxis:

```
C:\Path-To\compare.exe /max /wait /title1:%bname /title2:%ynname
                        %base %mine
```

or with UltraCompare:

```
C:\Path-To\uc.exe %base %mine -title1 %bname -title2 %ynname
```

or with DiffMerge:

```
C:\Path-To\DiffMerge.exe -nosplash -t1=%bname -t2=%ynname %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.

No fundo da caixa de diálogo podes configurar um programa para visualizar os ficheiros comp-unificada (ficheiros correcção). Não são requeridos parâmetros. A preferência omissão dá indicação para usar o TortoiseUDiff, que é instalado com o TortoiseSVN, e os códigos de cores para as linhas adicionadas e removidas.

Visto que o Unified Diff é apenas um formato de texto, poderás usar o teu editor de texto preferido se preferires.

#### 4.31.5.2. 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

**%nqbasename**  
The window title for the base file, without quotes

**%mine**  
O teu ficheiro, com as tuas alterações

**%yname**  
O título da janela do teu ficheiro

**%nqyname**  
The window title for your file, without quotes

**%theirs**  
o ficheiro, como está no repositório

**%tname**  
O título da janela para o ficheiro no repositório

**%nqtname**  
The window title for the file in the repository, without quotes

**%merged**  
o ficheiro em conflito, o resultado da operação de integração

**%mname**  
O título da janela para o ficheiro integrado

**%nqmname**  
The window title for the merged file, without quotes

**%fname**  
The name of the conflicted file

**%nqfname**  
The name of the conflicted file, without quotes

For example, with Perforce Merge:

```
C:\Path-To\P4Merge.exe %base %theirs %mine %merged
```

or with KDiff3:

```
C:\Path-To\kdiff3.exe %base %mine %theirs -o %merged  
--L1 %basename --L2 %yname --L3 %tname
```

or with Araxis:

```
C:\Path-To\compare.exe /max /wait /3 /title1:%tname /title2:%basename  
/title3:%yname %theirs %base %mine %merged /a2
```

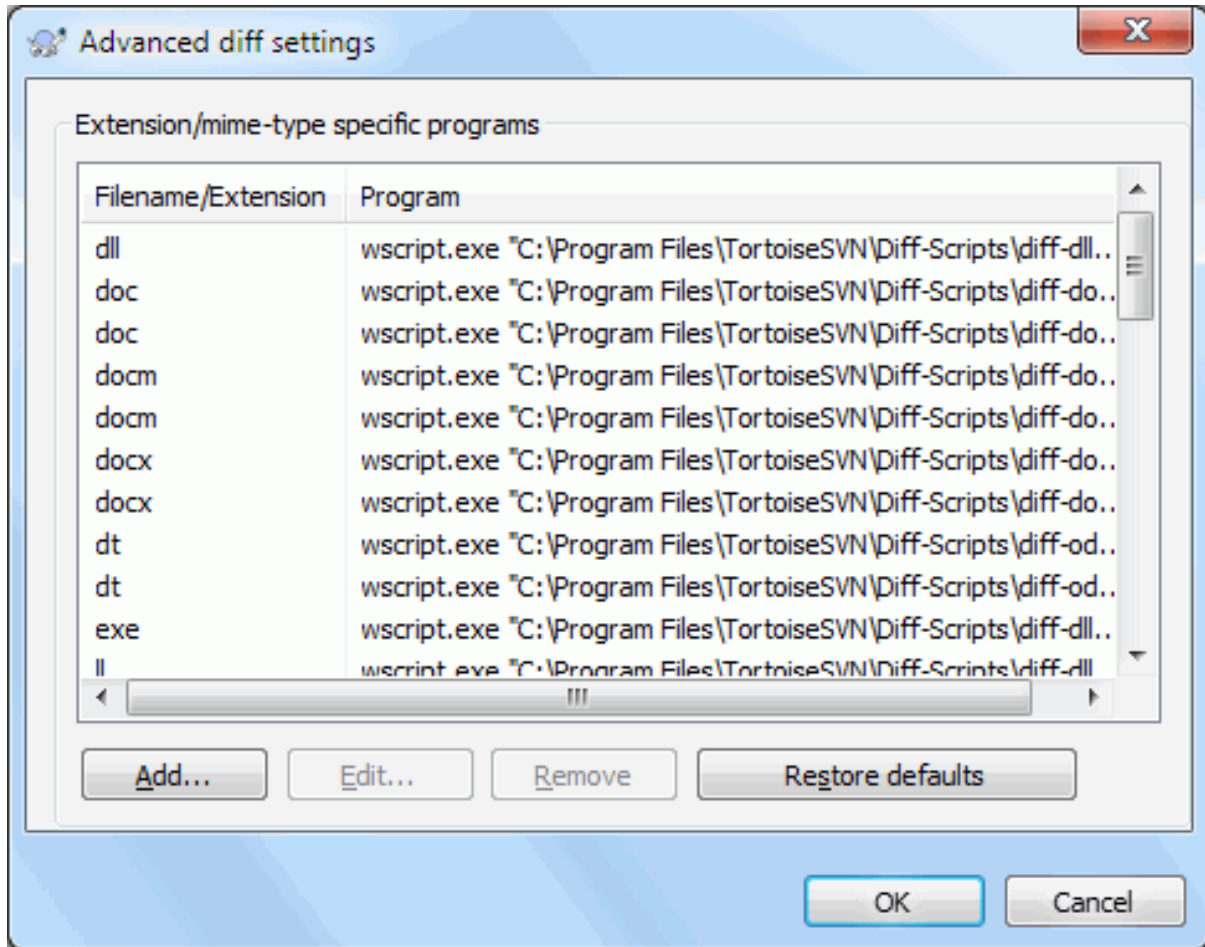
or with WinMerge (2.8 or later):

```
C:\Path-To\WinMerge.exe %merged
```

or with DiffMerge:

```
C:\Path-To\DiffMerge.exe -caption=%mname -result=%merged -merge
-nosplash -t1=%yname -t2=%bname -t3=%tname %mine %base %theirs
```

### 4.31.5.3. Preferências Avançadas de Comparar/Integrar

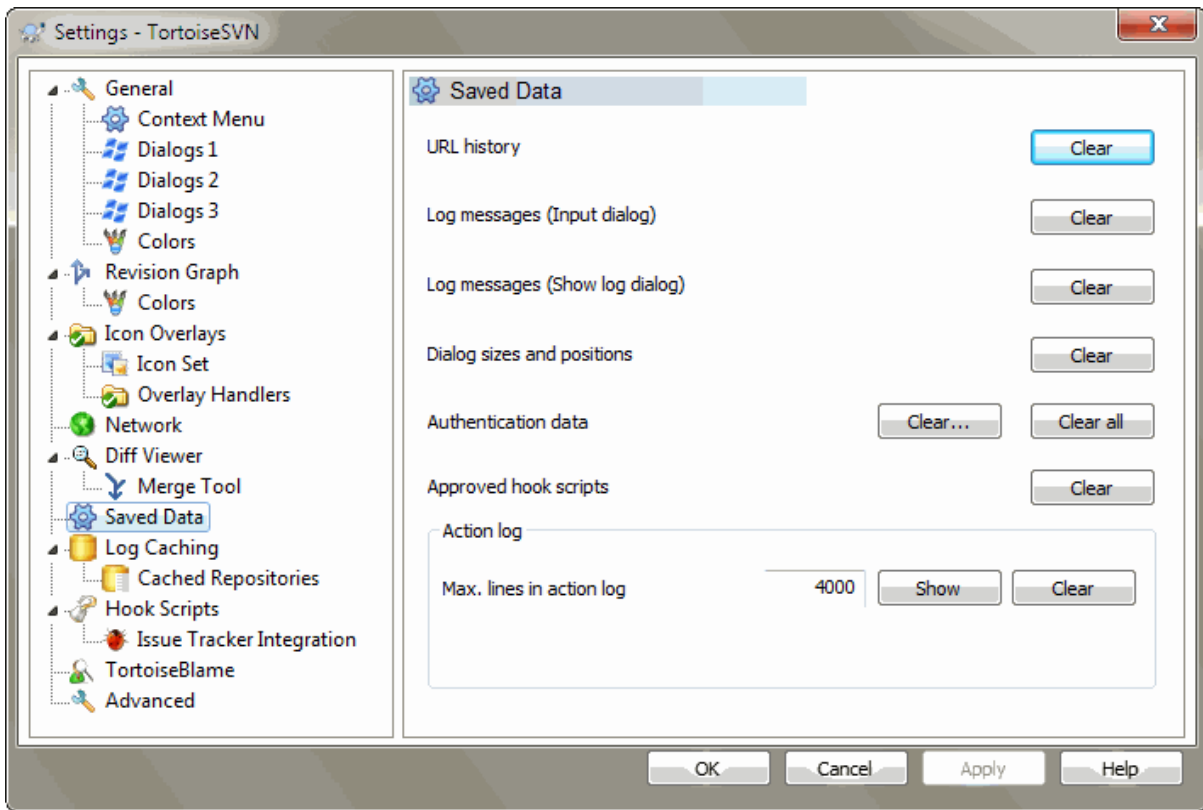


**Figura 4.82.** A Caixa de Diálogo Preferências, Caixa de Diálogo Comparar/Integrar Avançados

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`.

### 4.31.6. Preferências de Dados Guardados



**Figura 4.83. 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?

If you want to clear authentication data for one particular server only, use the **Clear...** instead of the **Clear all** button.

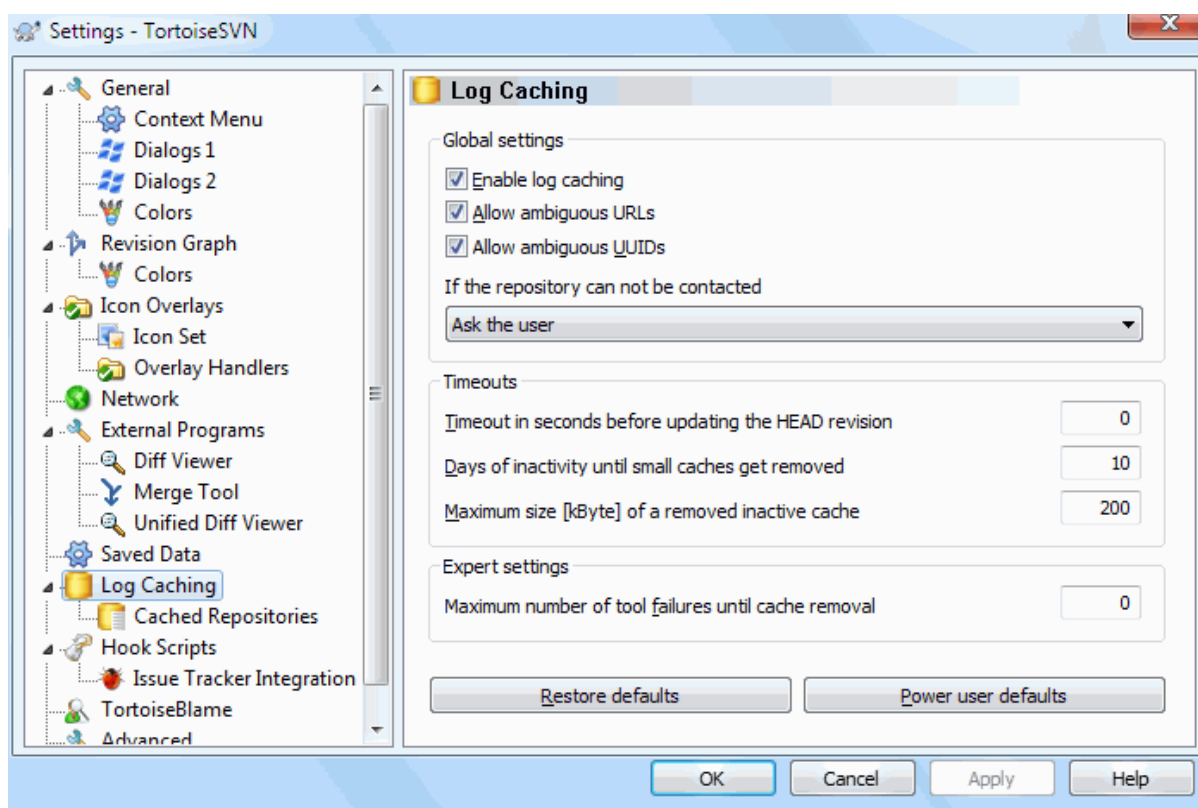
#### Registo de acções

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.

### 4.31.7. Cache de Registo



**Figura 4.84. 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 estouro. 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.

### 4.31.7.1. 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.

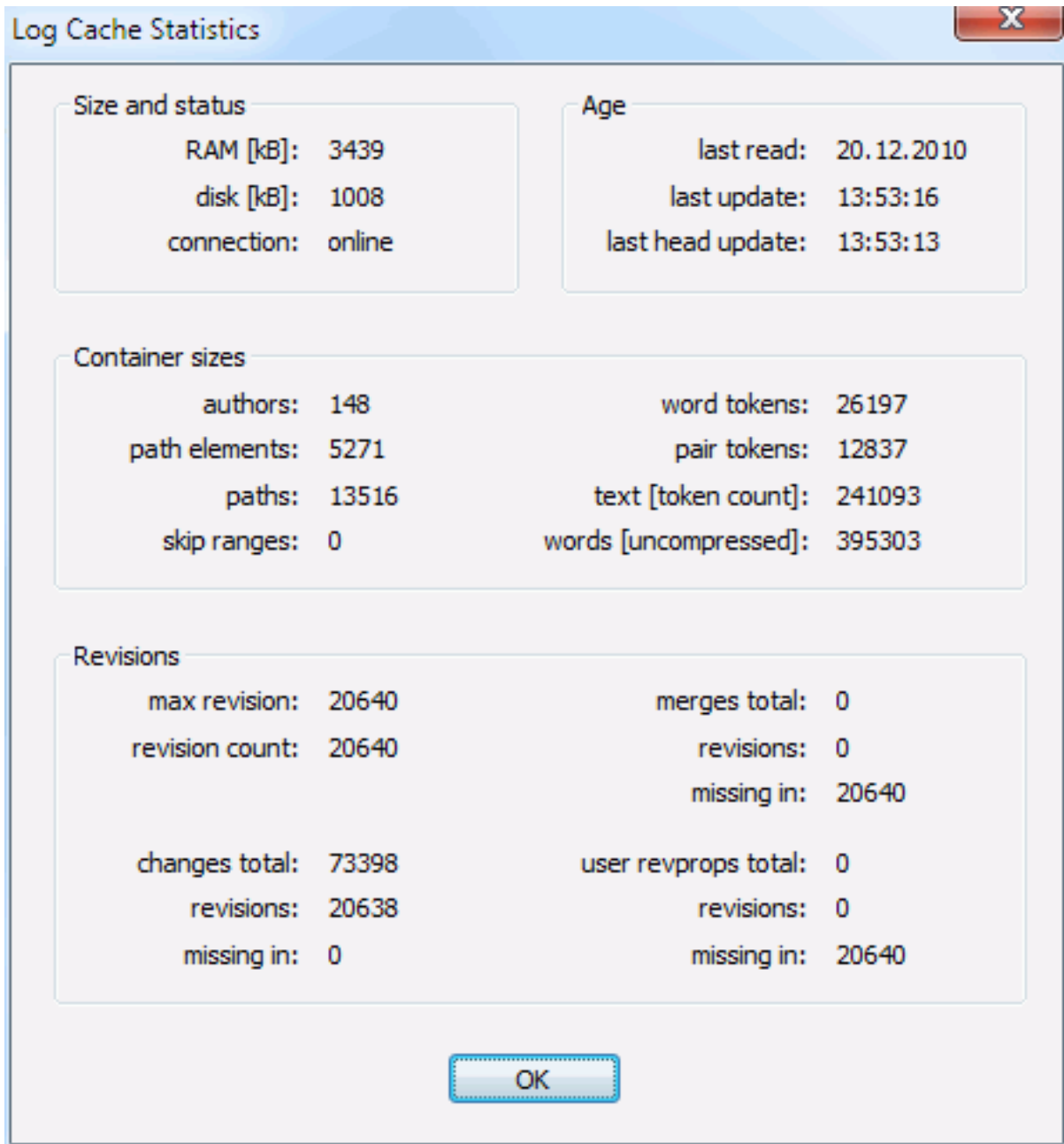
Clica no **Actualizar** para refrescar completamente a cache e preencher qualquer lacuna. Para um grande repositório isto poderá consumir muito tempo, mas será útil se estiveres prestes a entrar em modo fora-de-linha e queres a melhor cache disponível.

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.



#### 4.31.7.2. Estatísticas da Cache de Registo



**Figura 4.85. 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.

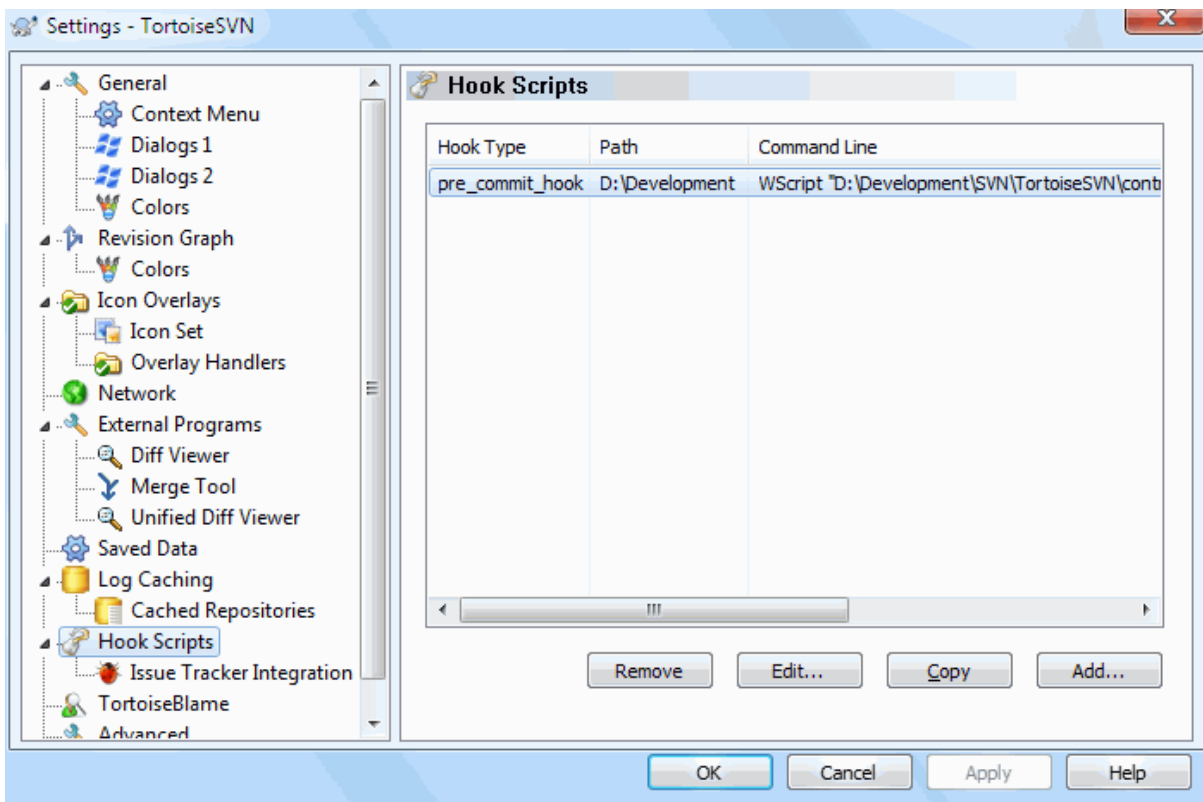
Revisão máxima

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.

#### 4.31.8. Scripts de Gancho do Lado do Cliente

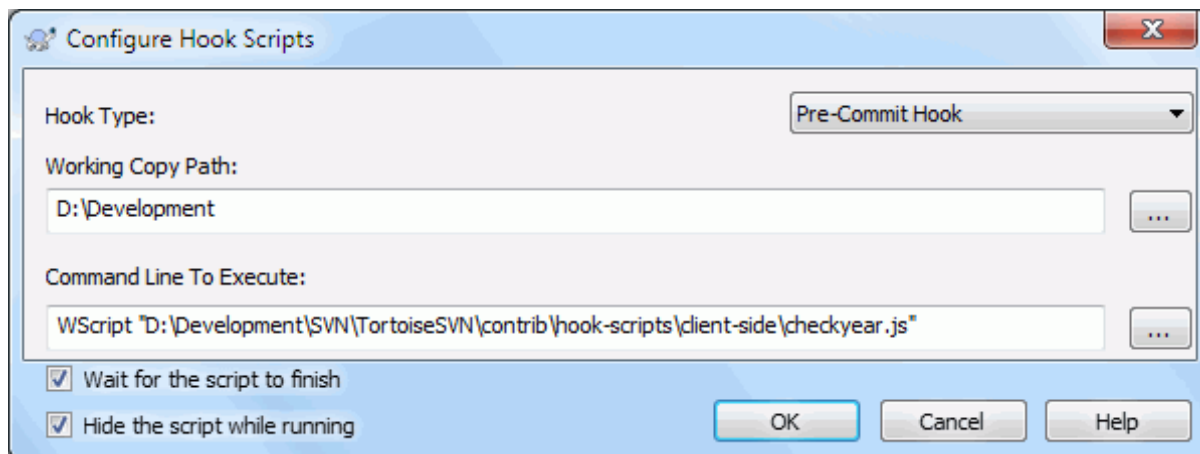


**Figura 4.86. A Caixa de Diálogo Preferências, Página Scripts de Gancho**

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.

Note that you can also specify such hook scripts using special properties on your working copy. See the section [Secção 4.18.2, “Propriedades de Projecto TortoiseSVN”](#) for details.



**Figura 4.87. A Caixa de Diálogo Preferências, Configurar Scripts de Gancho**

Para adicionar um novo script de gancho, clica simplesmente em Adicionar e preenche os detalhes.

There are currently these types of hook script available

#### 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.

#### Manual Pre-commit

If this is specified, the commit dialog shows a button Run Hook which when clicked runs the specified hook script. The hook script receives a list of all checked files and folders and the commit message if there was one entered.

#### Check-commit

Called after the user clicks OK in the commit dialog, and before the commit dialog closes. This hook gets a list of all the checked files. If the hook returns an error, the commit dialog stays open.

If the returned error message contains paths on newline separated lines, those paths will get selected in the commit dialog after the error message is shown.

#### 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

Called after the commit finishes successfully.

#### Início-actualização

Chamado antes da caixa de diálogo actualizar-para-revisão aparecer.

#### Pré-actualização

Chamado antes da actualização ou troca do Subversion realmente começar.

#### Pós-actualização

Chamado após a actualização, troca ou checkout terminar (com ou sem sucesso).

**Pré-conexão**

Chamado antes de uma tentativa para contactar o repositório. Chamado no máximo uma vez em cada cinco minutos.

**Pre-lock**

Called before an attempt to lock a file.

**Post-lock**

Called after a file has been locked.

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.

Em seguida, deverás especificar a linha de comando a executar, a começar pelo caminho para o script de gancho, ou para o executável. Este poderá ser um ficheiro batch, executável, ou qualquer outro ficheiro que tenha uma associação de ficheiro válida no Windows, e.g. um script de perl. Toma nota que o script não deverá ser especificado usando um caminho UNC, já que a execução da shell do Windows não irá permitir tal script correr devido a restrições de segurança.

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

**Manual Pre-commit**

PATHMESSAGEFILECWD

**Check-commit**

PATHMESSAGEFILECWD

**Pré-Submissão**

PATHDEPTHMESSAGEFILECWD

**Pós-submissão**

PATHDEPTHMESSAGEFILEREVISIONERRORCWD

**Início-actualização**

PATHCWD

**Pré-actualização**

PATHDEPTHREVISIONCWD

**Pós-actualização**

PATHDEPTHREVISIONERRORCWDRESULTPATH

**Pré-conexão**

no parameters are passed to this script. You can pass a custom parameter by appending it to the script path.

**Pre-lock**

PATHLOCKFORCEMESSAGEFILEERRORCWD

**Post-lock**

PATHLOCKFORCEMESSAGEFILEERRORCWD

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.

Note that for operations done remotely, e.g. in the repository browser, those paths are not local paths but the urls of the affected items.

#### DEPTH

Nível 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.

#### LOCK

Either `true` when locking, or `false` when unlocking.

#### FORCE

Either `true` or `false`, depending on whether the operation was forced or not.

#### 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.

#### RESULTPATH

A path to a temporary file which contains all the paths which were somehow touched by the operation. Each path is on a separate line in the temp file.

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 **Espera 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.

Sample client hook scripts can be found in the contrib folder in the *TortoiseSVN repository* [https://svn.code.sf.net/p/tortoisesvn/code/trunk/contrib/hook-scripts]. (Secção 3, “Licença” explains how to access the repository.)

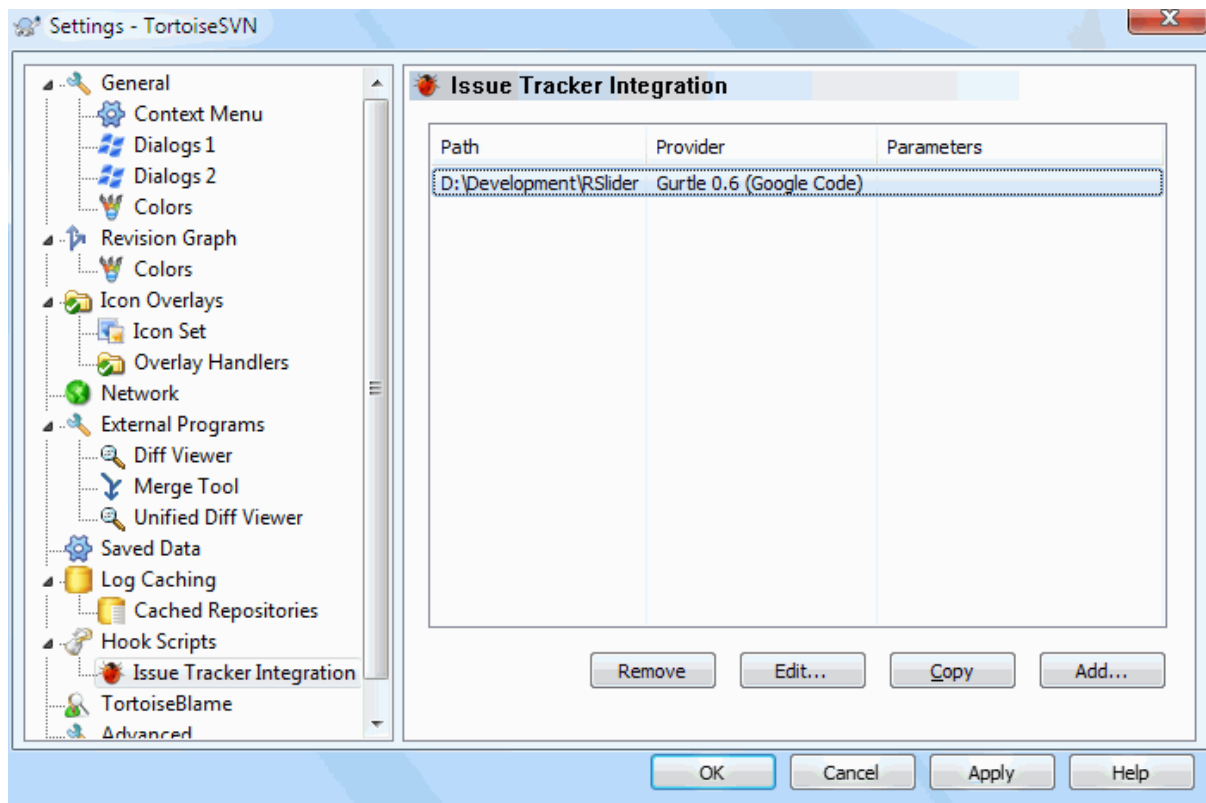
When debugging hook scripts you may want to echo progress lines to the DOS console, or insert a pause to stop the console window disappearing when the script completes. Because I/O is redirected this will not normally work. However you can redirect input and output explicitly to CON to overcome this. e.g.

```
echo Checking Status > con
pause < con > con
```

Uma pequena ferramenta chamada ConnectVPN.exe está incluída na pasta de instalação do TortoiseSVN. Poderás usar esta ferramenta, configurada como pré-connect, para te ligares automaticamente à tua VPN antes do TortoiseSVN se tentar ligar a um repositório. Terás apenas de passar o nome da conexão VPN como primeiro parâmetro da ferramenta.

### 4.31.8.1. Integração do Controlador de Problemas

O TortoiseSVN pode usar um plugin COM para consultar os controladores de problemas quando estiver na caixa de diálogo submeter. O uso de tais plugins está descrito em Secção 4.29.2, “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 4.88. 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 para teu projecto, poderás especificá-lo também com as seguintes propriedades `bugtraq:provideruuid`, `bugtraq:provideruuid64` e `bugtraq:providerparams`.

`bugtraq:provideruuid`

Esta propriedade especifica o UUID COM do `IBugtraqProvider`, por exemplo `{91974081-2DC7-4FB1-B3BE-0DE1C8D6CE4E}`. (Este exemplo é o UUID do *fornecedor do 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:provideruuid64`

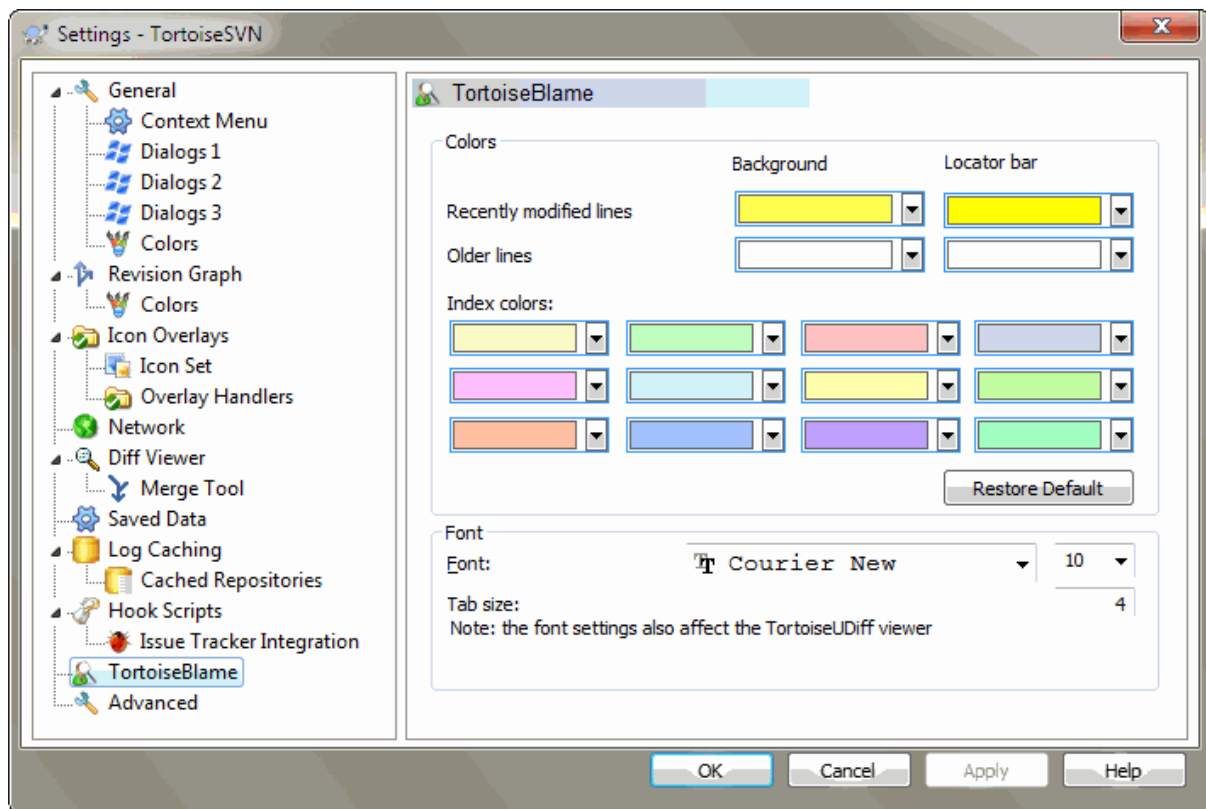
Isto é o mesmo que `bugtraq:provideruuid`, mas para a versão 64-bit do `IBugtraqProvider`.

`bugtraq:providerparams`

Esta propriedade especifica os parâmetros passados para o `IBugtraqProvider`.

Por favor consulta a documentação para o teu plugin `IBugtraqProvider` para saberes o que especificar nessas duas propriedades.

### 4.31.9. Preferências do TortoiseBlame



**Figura 4.89. 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.

Poderás especificar diferentes cores a usar na barra de localização. Por defeito é usado um grande contraste na barra de localização enquanto é mantido claro o fundo da janela principal, para que possas ainda ler o texto.

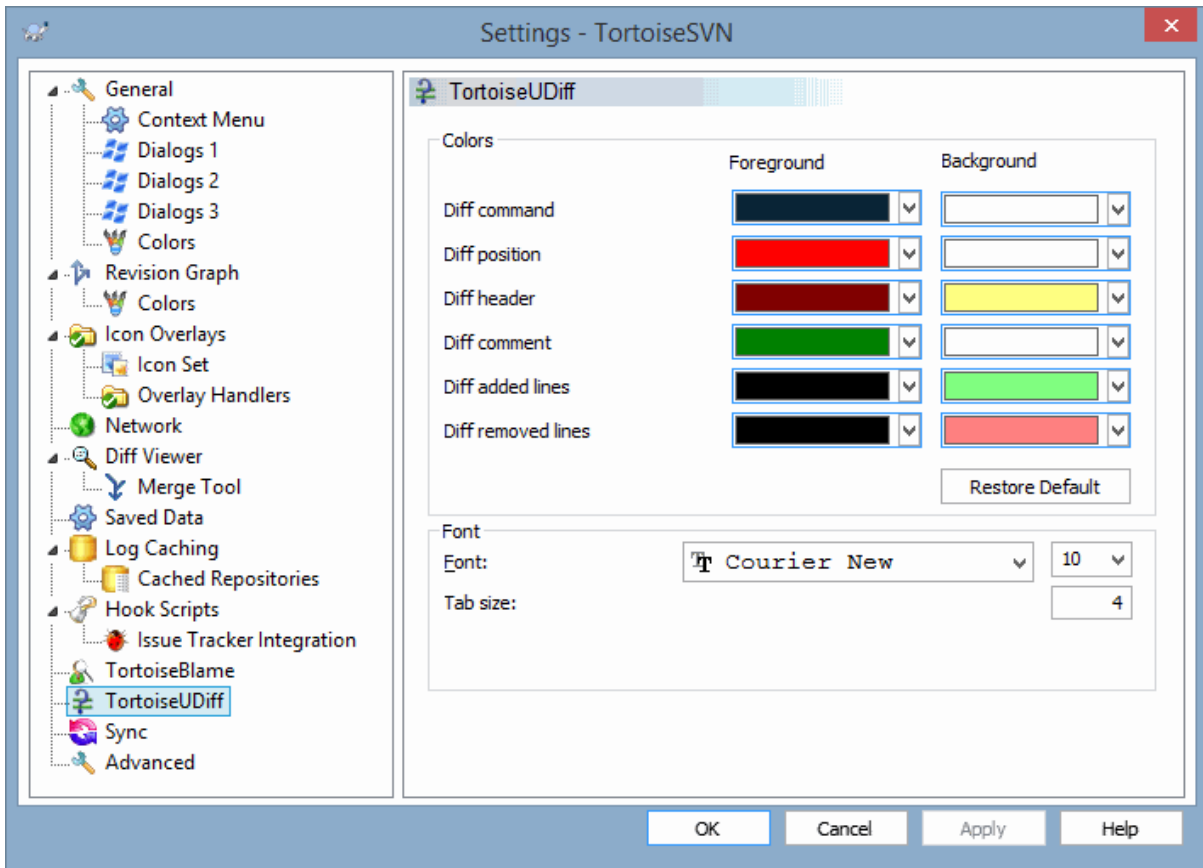
**Tipo de letra**

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.

**4.31.10. TortoiseUDiff Settings**



**Figura 4.90. The Settings Dialog, TortoiseUDiff Page**

The settings used by TortoiseUDiff are controlled from the main context menu, not directly with TortoiseUDiff itself.

**Cores**

The default colors used by TortoiseUDiff are usually ok, but you can configure them here.

**Tipo de letra**

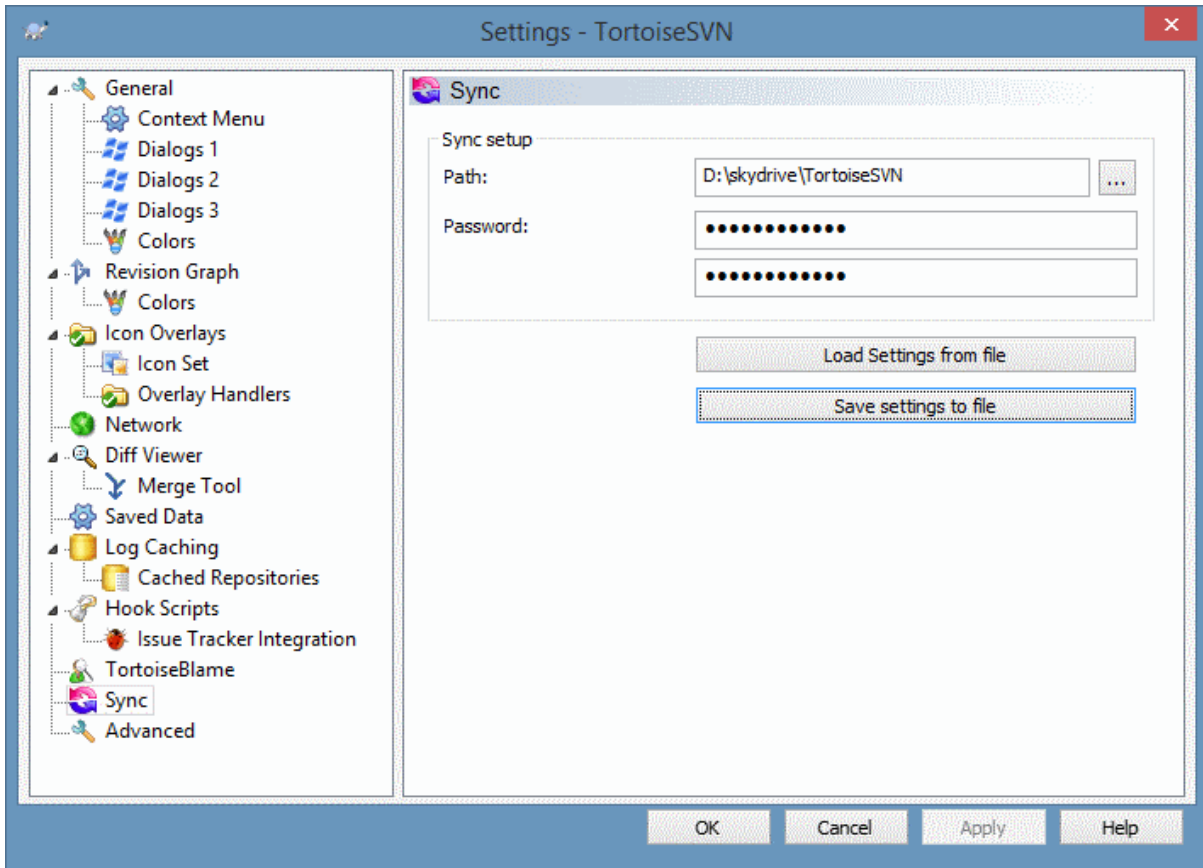
You can select the font used to display the text, and the point size to use.

**Tabuladores**

Defines how many spaces to use for expansion when a tab character is found in the file diff.



### 4.31.11. Exportando preferências TSVN



**Figura 4.91. The Settings Dialog, Sync Page**

You can sync all TortoiseSVN settings to and from an encrypted file. The file is encrypted with the password you enter so you don't have to worry if you store that file on a cloud folder like OneDrive, GDrive, DropBox, ...

When a path and password is specified, TortoiseSVN will sync all settings automatically and keep them in sync.

You can also export/import an encrypted files with all the settings manually. When you do that, you're asked for the path of the file and the password to encrypt/decrypt the settings file.

When exporting the settings manually, you can also optionally include all local settings which are not included in a normal export or in a sync. Local settings are settings which include local paths which usually vary between computers. These local settings include the configured diff and merge tools and hook scripts.

### 4.31.12. Preferências Avançadas

Algumas preferências raramente usadas estão só disponíveis na página avançada da caixa de diálogo de preferências. Essas preferências modificam directamente o registo, e tens de saber para o que cada uma dessas preferências é usada, e o que isso implica. Não modifiques essas preferências a não ser que tenhas a certeza que necessitas realmente de as alterar.

#### AllowAuthSave

Por vezes, múltiplos utilizadores usam a mesma conta no mesmo computador. Nessas situações não é desejável guardar os dados de autenticação. Configurando este valor a `false` desactiva o botão guardar autenticação na caixa de diálogo de autenticação.

#### AllowUnversionedObstruction

Se uma actualização adiciona um novo ficheiro do repositório que já existe na cópia de trabalho local como ficheiro não versionado, a acção por defeito será manter o ficheiro local, mostrando-o como (possivelmente)

uma versão modificada do novo ficheiro do repositório. Mas se preferires que o TortoiseSVN crie um conflito em tais situações, configura este valor a `false`.

#### AlwaysExtendedMenu

Como no Explorador, o TortoiseSVN ,outra comandos adicionais se a tecla **Shift** estiver premida enquanto o menu de contexto for aberto. Para forçar o TortoiseSVN a mostrar sempre esse comando extendidos, configura este valor a `true`.

#### AutoCompleteMinChars

The minimum amount of chars from which the editor shows an auto-completion popup. The default value is 3.

#### AutocompleteRemovesExtensions

A lista de preenchimento automático, do editor de mensagem de submissão, mostra os nomes dos ficheiros listados para submissão. Para também incluir esses nomes com extensões removidas, fica este valor a `true`.

#### BlockPeggedExternals

File externals that are pegged to a specific revision are blocked by default from being selected for a commit. This is because a subsequent update would revert those changes again unless the pegged revision of the external is adjusted.

Set this value to `false` in case you still want to commit changes to such external files.

#### BlockStatus

Se não queres que o explorador actualize o estado das sobreposições enquanto outro comando do TortoiseSVN está a correr (e.g. Actualizar, Submeter, ...) e fixa este valor a `true`.

#### CacheTrayIcon

Para adicionar um ícone de bandeja de cache para o programa TSVNCache, fixa este valor a `true`. Isto só é realmente útil aos programadores já que permite terminar o programa graciosamente.

#### ColumnsEveryWhere

The extra columns the TortoiseSVN adds to the details view in Windows Explorer are normally only active in a working copy. If you want those to be accessible everywhere, not just in working copies, set this value to `true`. Note that the extra columns are only available in XP. Vista and later doesn't support that feature any more. However some third-party explorer replacements do support those even on Windows versions later than XP.

#### ConfigDir

Tu podes especificar aqui uma localização diferente para o teu ficheiro de configuração do Subversion. Isto irá afectar todas as operações do TortoiseSVN.

#### CtrlEnter

Na maioria das caixas de diálogo do TortoiseSVN, tu podes usar **Ctrl+Enter** para fechar a caixa de diálogo, como se clicasses no botão OK. Se não desejas isto, configura este valor para `false`.

#### Debug

Configura isto a `true` se desejas que apareça uma caixa de diálogo para cada comando, mostrando a linha de comando usada para arrancar o TortoiseProc.exe.

#### DebugOutputString

Coloca esta opção como `verdadeiraliteral` se quiseres que o TortoiseSVN imprima mensagens de compila

#### DialogTitles

O formato padrão (valor 0) dos títulos do diálogo é `url/caminho - nome do diálogo - TortoiseSVN`. Se definires esse valor para 1, o formato é alterado para `nome do diálogo - url/caminho - TortoiseSVN`.

#### DiffBlamesWithTortoiseMerge

O TortoiseSVN permite-te atribuir um visualizador de comparações externo. No entanto, a maior parte desses visualizadores não são adequados para modificar responsabilidades ([Secção 4.24.2, "Diferenças de](#)

**responsabilidade**”), pelo que tu poderás desejar voltar neste caso ao TortoiseMerge. Para o fazer, fixa este valor a `true`.

#### DlgStickySize

This value specifies the number of pixels a dialog has to be near a border before the dialog sticks to it. The default value is 3. To disable this value set the value to zero.

#### FixCaseRenames

Algumas aplicações alteram a altura da letra dos nomes de ficheiros sem aviso, mas essas alterações não são desejadas nem necessárias. Por exemplo uma mudança de `file.txt` para `FILE.TXT` não afectaria aplicações normais de Windows, mas o Subversion é sensível à altura da letra nessas situações. Pelo que, o TortoiseSVN repara automaticamente essas situações.

Se não pretenderes que o TortoiseSVN corrija automaticamente por ti essas alterações de altura de letra, podes colocar este valor a `false`.

#### FullRowSelect

O controlo da lista de estado, que é usada em várias caixas de diálogo (e.g. submeter, verificar-alterações, adicionar, reverter, ...) usa a selecção de linha completa (i.e., se seleccionares uma entrada, a linha completa será seleccionada e não somente a primeira coluna). Isto está correcto, mas a linha seleccionada também cobre a imagem de fundo à direita, o que pode ser visualmente desagradável. Para desactivar a selecção de linha completa, configura este valor a `false`.

#### GroupTaskbarIconsPerRepo

This option determines how the Win7 taskbar icons of the various TortoiseSVN dialogs and windows are grouped together. This option has no effect on Vista!

1. O valor padrão é 0. Com esta definição, os ícones são agrupados por tipo de aplicação. Todas as caixas de diálogo do TortoiseSVN são agrupadas, todas as janelas do TortoiseMerge são agrupadas, ...



**Figura 4.92. Barra de tarefas com agrupamento padrão**

2. Se definido como 1, então em vez de todos os diálogos em um único grupo por aplicação, eles são agrupados por repositório. Por exemplo, se você tem um diálogo aberto e um log de diálogo de confirmação para repositório A, e uma caixa de diálogo de check-para-modificações e uma caixa de diálogo registo para o repositório B, Então existem dois grupos de aplicativos ícone mostrado na barra de tarefas Win7, um grupo para cada repositório. Mas as janelas TortoiseMerge não são agrupadas com diálogos TortoiseSVN.



**Figura 4.93. Barra de tarefas com agrupamento de repositório**

3. Se definido como 2, então o agrupamento funciona como se tivesse a configuração definida para 1, excepto que o TortoiseSVN, TortoiseMerge, TortoiseBlame, TortoiseIDiff TortoiseUDiff e janelas são todas agrupadas. Por exemplo, se você tiver a janela de commit aberta e, em seguida, clicar duas vezes num arquivo modificado, a janela TortoiseMerge diff que se abriu será colocada no grupo mesmo ícone na barra de tarefas como o ícone de diálogo de confirmação.



**Figura 4.94. Barra de tarefas com agrupamento de repositório**

4. Se estiver definido para 3, então o grupo funciona como se a configuração estivesse definida para 1, mas o agrupamento não é feito de acordo com o repositório, mas de acordo com a cópia de trabalho. Isso é útil se você tiver todos os seus projetos no mesmo repositório, mas diferentes cópias de trabalho para cada projeto.
5. Se definido para 4, então o grupo funciona como se tivesse a configuração definida para 2, mas o agrupamento não é feito de acordo com o repositório, mas de acordo com a cópia de trabalho.

#### HideExternalInfo

Se for posto a `false`, então é mostrado em separado cada `svn:externals` durante uma actualização.

Se for colocado `true` (por defeito), então a actualização de informação para os externos só é mostrada se os externos forem afectados pela actualização, i.e. alteradas de algum modo. De outro modo nada é mostrado, tal como com os ficheiros e pastas normais.

#### GroupTaskbarIconsPerRepoOverlay

Isto não tem efeito se a opção `GroupTaskbarIconsPerRepo` está definida para 0 (veja acima).

Se esta opção estiver definida para `true`, Então cada ícone na barra de tarefas do Win7 mostra uma sobreposição de cor no pequeno rectângulo, indicando o repositório os diálogos/janelas são usados.



**Figura 4.95. Agrupamento da barra de tarefas com sobreposições de cores do repositório**

#### IncludeExternals

Por defeito, o TortoiseSVN corre sempre uma actualização incluindo os externos. Isto evita problemas com cópias de trabalho inconsistentes. No entanto, uma actualização poderá levar algum tempo, se tiveres a usar muitos externos. Configura este valor para `false` para correr a actualização por defeito, com os externos incluídos. Para actualizar os externos incluídos, ou correr a caixa de diálogo Actualizar para revisão... ou configura de novo este valor a `true`.

#### LogFindCopyFrom

When the log dialog is started from the merge wizard, already merged revisions are shown in gray, but revisions beyond the point where the branch was created are also shown. These revisions are shown in black because those can't be merged.

If this option is set to `true` then TortoiseSVN tries to find the revision where the branch was created from and hide all the revisions that are beyond that revision. Since this can take quite a while, this option is disabled by default. Also this option doesn't work with some SVN servers (e.g., Google Code Hosting, see [issue #5471](http://code.google.com/p/support/issues/detail?id=5471) [<http://code.google.com/p/support/issues/detail?id=5471>]).

#### LogMultiRevFormat

A format string for the log messages when multiple revisions are selected in the log dialog.

You can use the following placeholders in your format string:

`%1!ld!`

gets replaced with the revision number text

`%2!s!`

gets replaced with the short log message of the revision

#### LogStatusCheck

A caixa de diálogo registo mostra a revisão em que está o caminho da cópia de trabalho a negrito. Mas isto requer que a caixa de diálogo de registo obtenha o estado para esse caminho. Visto que para cópias de trabalho muito grandes, isto pode demorar algum tempo, tu podes configurar este valor a `false`, para deste modo desactivares esta funcionalidade.

#### MaxHistoryComboItems

Comboboxes for URLs and paths show a history of previously used URLs/paths if possible. This settings controls how many previous items are saved and shown. The default is 25 items.

#### MergeLogSeparator

Quando tu integras revisões de outro ramo, e a informação de seguimento de integração está disponível, as mensagens de registo das revisões que integras serão colhidas de modo a construir uma mensagem de registo de submissão. A cadeia de caracteres pré-definida é usada para separar mensagens de registo individuais das revisões integradas. Se preferires, podes fixar isto a um valor que contenha uma cadeia de caracteres de separação da tua escolha.

#### NumDiffWarning

If you want to show the diff at once for more items than specified with this settings, a warning dialog is shown first. The default is 10.

#### OldVersionCheck

O TortoiseSVN verifica se existe disponível uma nova versão cerca de uma vez uma por semana. Se for encontrada uma versão actualizada, a caixa de diálogo submeter mostrará um controlo de hiperligação com essa informação. Se preferires o comportamento antigo em que uma caixa de diálogo aparece a notificar-te sobre a actualização, configura este valor para `true`.

#### RepoBrowserTrySVNParentPath

The repository browser tries to fetch the web page that's generated by an SVN server configured with the `SVNParentPath` directive to get a list of all repositories. To disable that behavior, set this value to `false`.

#### ScintillaBidirectional

This option enables the bidirectional mode for the commit message edit box. If enabled, right-to-left language text editing is done properly. Since this feature is expensive, it is disabled by default. You can enable this by setting this value to `true`.

#### ScintillaDirect2D

Esta opção activa o uso de aceleração Direct2D em desenho no controlo Scintilla, que é usado como caixa de edição como por exemplo na janela Submeter, e também para o visualizador de comparação unificada. Com algumas placas gráficas esta funcionalidade por vezes não funciona correctamente, pelo que o cursor para introdução de texto não está sempre visível. Se isso acontecer, podes desligar esta funcionalidade colocando este valor a `false`.

#### OutOfDateRetry

This parameter specifies how TortoiseSVN behaves if a commit fails due to an out-of-date error:

0

The user is asked whether to update the working copy or not, and the commit dialog is not reopened after the update.

- 1 This is the default. The user is asked whether to update the working copy or not, and the commit dialog is reopened after the update so the user can proceed with the commit right away.
- 2 Similar to 1, but instead of updating only the paths selected for a commit, the update is done on the working copy root. This helps to avoid inconsistent working copies.
- 3 The user is not asked to update the working copy. The commit simply fails with the out-of-date error message.

#### PlaySound

If set to `true`, TortoiseSVN will play a system sound when an error or warning occurs, or another situation which is important and requires your attention. Set this to `false` if you want to keep TortoiseSVN quiet. Note that the project monitor has its own setting for playing sounds, which you can configure in its settings dialog.

#### ShellMenuAccelerators

O TortoiseSVN usa aceleradores para as entradas do seu menu de contexto. Visto que isto pode levar a aceleradores duplicados (e.g. o SVN Submeter tem o acelerador **Alt-C**, mas também o tem a entrada Copiar do explorador). Se não queres ou necessitas dos aceleradores das entradas do TortoiseSVN, configura este valor a `false`.

#### ShowContextMenuIcons

Isto pode ser útil se usares algo mais antigo que o explorador do Windows, ou se tiveres problemas com a visualização incorrecta do menu de contexto. Fixa este valor a `false` se não queres que o TortoiseSVN mostre ícones para os itens do menu de contexto da shell. Fixa este valor a `true` para mostrar os ícones outra vez.

#### ShowAppContextMenuIcons

Se não queres que o TortoiseSVN mostre ícones para os menus de contexto nas suas próprias caixas de diálogo, configura este valor a `false`.

#### StyleCommitMessages

As caixas de diálogo submeter e registo usam estilos (e.g. negrito, itálico) nas mensagens de submissão (visualiza [Secção 4.4.5, "Mensagens de Registo de Submeter"](#) para mais detalhes). Se não queres usar isto, configura o valor a `false`.

#### UpdateCheckURL

Este valor contém o URL a partir do qual o TortoiseSVN tenta descarregar um ficheiro de texto, para tentar descobrir se existem actualizações disponíveis. Isto poderá ser útil a administradores de empresas, que não quererão que os seus utilizadores actualizem o TortoiseSVN sem a sua autorização.

#### UseCustomWordBreak

The standard edit controls do not stop on forward slashes like they're found in paths and urls. TortoiseSVN uses a custom word break procedure for the edit controls. If you don't want that and use the default instead, set this value to 0. If you only want the default for edit controls in combo boxes, set this value to 1.

#### VersionCheck

O TortoiseSVN verifica se existe disponível uma nova versão cerca de uma vez uma por semana. Se não queres que o TortoiseSVN faça esta verificação, configura este valor para `false`.

## 4.32. Passo Final

### Doa!

Even though TortoiseSVN and TortoiseMerge are free, you can support the developers by sending in patches and playing an active role in the development. You can also help to cheer us up during the endless hours we spend in front of our computers.

While working on TortoiseSVN we love to listen to music. And since we spend many hours on the project we need a *lot* of music. Therefore we have set up some wish-lists with our favourite music CDs and DVDs: <https://tortoisesvn.net/donate.html> Please also have a look at the list of people who contributed to the project by sending in patches or translations.

# Capítulo 5. Project Monitor

The project monitor is a helpful tool that monitors repositories and notifies you in case there are new commits.

The projects can be monitored via a working copy path or directly via their repository URLs.

The project monitor scans each project in a configurable interval, and every time new commits are detected a notification popup is shown. Also the icon that is added to the system tray changes to indicate that there are new commits.

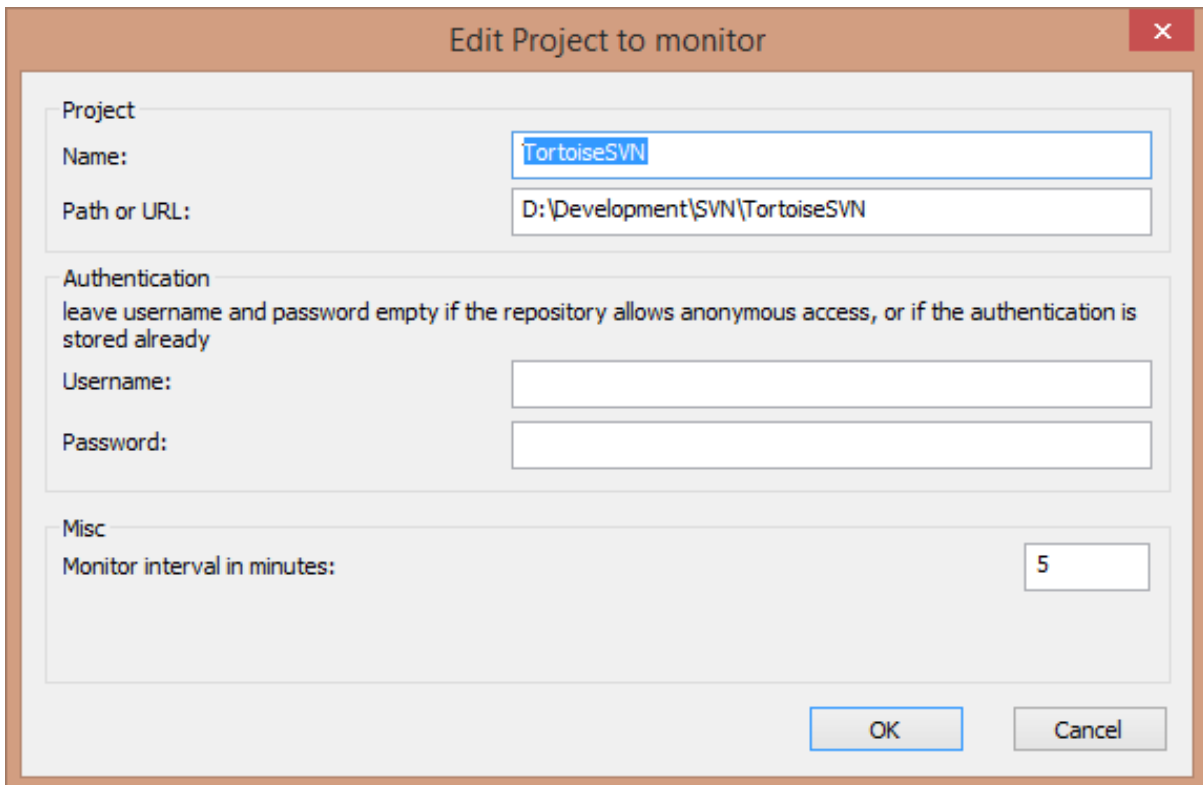


## Snarl

If *Snarl* [<https://snarl.fullphat.net/>] is installed and active, then the project monitor automatically uses Snarl to show the notifications about newly detected commits.

## 5.1. Adding projects to monitor

If you first start the project monitor, the tree view on the left side is empty. To add projects, click on the button at the top of the dialog named Add Project.



**Figura 5.1. The edit project dialog of the project monitor**

To add a project for monitoring, fill in the required information. The name of the project is not optional and must be filled in, all other information is optional.

If the box for `Path or URL` is left empty, then a folder is added. This is useful to group monitored projects.

The fields `Username` and `Password` should only be filled in if the repository does not provide anonymous read access, and only if the authentication is not stored by Subversion itself. If you're accessing the monitored repository with TortoiseSVN or other svn clients and you've stored the authentication already, you should leave this empty: you won't have to edit those projects manually if the password changes.



The `Monitor interval` in minutes specifies the minutes to wait in between checks. The smallest interval is one minute.



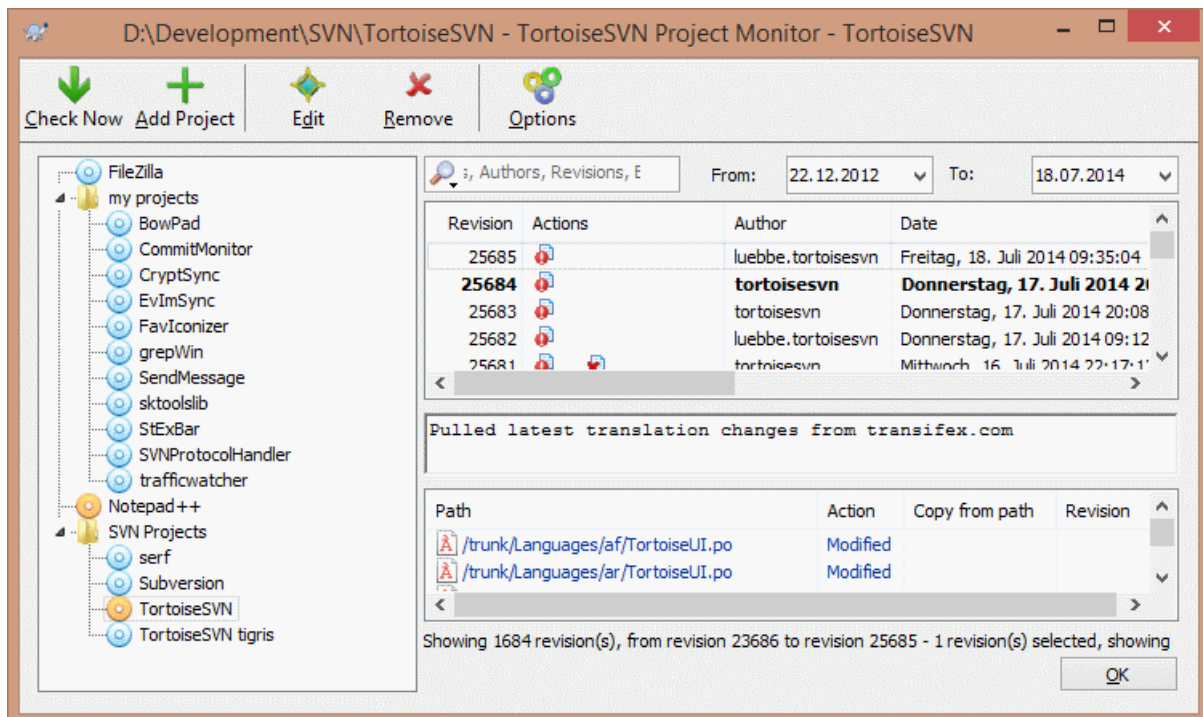
## check interval

If there are a lot of users monitoring the same repository and the bandwidth on the server is limited, a repository admin can set the minimum for check intervals using an `svnrobots.txt` file. A detailed explanation on how this works can be found on the project monitor website:

<http://stefanstools.sourceforge.net/svnrobots.html>

[<http://stefanstools.sourceforge.net/svnrobots.html>]

## 5.2. Monitor dialog



**Figura 5.2.** The main dialog of the project monitor

The project monitor shows all monitored projects on the left in a tree view. The projects can be moved around, for example one project can be moved below another project, making it a child/subproject.

A click on a project shows all the log messages of that project on the right.

Projects that have updates are shown in bold, with the number of new commits in brackets at the right. A click on a project marks it automatically as read.

### 5.2.1. Main operations

The toolbar at the top of the dialog allows to configure and operate the project monitor.

#### Check Now

While each monitored project is checked according to the interval that's set up, clicking this button will force a check of all projects immediately. Note that if there are updates, the notification won't show up until all projects have been checked.

#### Add Project

Opens a new dialog to set up a new project for monitoring.

**Editar**

Opens the configuration dialog for the selected project.

**Remover**

Removes the selected project after a confirmation dialog is shown.

**Mark all as read**

Marks all revisions in all projects as read. Note that if you select a project with unread revisions, those revisions are automatically marked as read when you select another project.

If you hold down the **Shift** key when clicking the button, all error states are also cleared if there are any.

**Update all**

Runs an **Update** on all monitored working copies. Projects that are monitored via an url are not updated, only those that are set up with a working copy path.

**Opções**

Shows a dialog to configure the behavior of the project monitor.

---

# Capítulo 6. 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”.

## 6.1. A Linha de Comando SubWCRev

O SubWCRev lê por defeito o estado Subversion de todos os ficheiros numa cópia de trabalho, incluindo os externos. Ele grava o número da mais alta revisão submetida que for encontrada e também a sua marca temporal, também regista a existência de alterações locais na cópia de trabalho ou, actualizações de revisões misturadas. O número de revisão, intervalo de revisões actualizadas e o estado de modificações, são mostrados no stdout.

SubWCRev.exe is called from the command line or a script, and is controlled using the command line parameters.

```
SubWCRev WorkingCopyPath [SrcVersionFile DstVersionFile] [-nmdfe]
```

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.

You can specify ignore patterns for SubWCRev to prevent specific files and paths from being considered. The patterns are read from a file named `.subwcrevignore`. The file is read from the specified path, and also from the working copy root. If the file does not exist, no files or paths are ignored. The `.subwcrevignore` file can contain multiple patterns, separated by newlines. The patterns are matched against the paths relative to the repository root and paths relative to the path of the `.subwcrevignore` file. For example, to ignore all files in the `doc` folder of the TortoiseSVN working copy, the `.subwcrevignore` would contain the following lines:

```
/trunk/doc  
/trunk/doc/*
```

Or, assuming the `.subwcrevignore` file is in the working copy root which is checked out from trunk, using the patterns

```
doc  
doc/*
```

is the same as the example above.

To ignore all images, the ignore patterns could be set like this:

```
*.png  
*.jpg  
*.ico  
*.bmp
```



### Importante

The ignore patterns are case-sensitive, just like Subversion is.



## Dica

To create a file with a starting dot in the Windows explorer, enter `.subwcrevignore..` Note the trailing dot.

Existe um conjunto de opções opcionais que afectam a maneira como o SubWCRev funciona. Se usares mais do que uma, elas devem ser especificadas como um único grupo, e.g. `-nm`, e não `-n -m`.

Trocar	Descrição
-n	Se é usada esta opção o SubWCRev irá sair com <code>ERRORLEVEL 7</code> se a cópia de trabalho contiver alterações locais. Isto pode ser usado para evitar construir com alterações não submetidas presentes.
-N	If this switch is given, SubWCRev will exit with <code>NÍVEL de ERRO 11</code> if the working copy contains unversioned items that are not ignored.
-m	Se esta opção for usada o SubWCRev irá sair com <code>ERRORLEVEL 8</code> se a cópia de trabalho contiver revisões misturadas. Isto pode ser útil para evitar construir com uma cópia de trabalho parcialmente actualizada.
-d	Se esta opção for usada o SubWCRev irá sair com <code>ERRORLEVEL 9</code> se o ficheiro de destino já existe.
-f	Se esta opção for usada o SubWCRev irá incluir a última revisão alterada de pastas. O comportamento por defeito será usar só ficheiros quando a obter os números de revisão.
-e	Se esta opção for usada o SubWCRev irá examinar as pastas que estão incluídas com <code>svn:externals</code> , mas só se elas forem do mesmo repositório. O comportamento por defeito é ignorar os externos.
-E	If this switch is given, same as <code>-e</code> , but it ignores the externals with explicit revisions, when the revision range inside of them is only the given explicit revision in the properties. So it doesn't lead to mixed revisions.
-x	Se esta opção for usada o SubWCRev irá produzir os números de revisão em HEX.
-X	Se esta opção for usada o SubWCRev irá produzir os números de revisão em HEX, com o prefixo '0X'
-F	If this switch is given, SubWCRev will ignore any <code>.subwcrevignore</code> files and include all files.
-q	If this switch is given, SubWCRev will perform the keyword substitution without showing working copy status on stdout.

**Tabela 6.1. Lista de opções de linha de comando disponíveis**

If there is no error, SubWCRev returns zero. But in case an error occurs, the error message is written to `stderr` and shown in the console. And the returned error codes are:

Código do Erro	Descrição
1	Erro de sintaxe. Um ou mais parâmetros da linha de comando são inválidos.
2	The file or folder specified on the command line was not found.
3	The input file could not be opened, or the target file could not be created.
4	Could not allocate memory. This could happen if e.g. the source file is too big.
5	The source file can not be scanned properly.
6	SVN error: Subversion returned with an error when SubWCRev tried to find the information from the working copy.
7	The working copy has local modifications. This requires the <code>-n</code> switch.
8	The working copy has mixed revisions. This requires the <code>-m</code> switch.

Código do Erro	Descrição
9	The output file already exists. This requires the <code>-d</code> switch.
10	The specified path is not a working copy or part or one.
11	The working copy has unversioned files or folders in it. This requires the <code>-N</code> switch.

Tabela 6.2. List of SubWCRev error codes

## 6.2. 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>\$WCREV&amp;\$</code>	Substituída com a revisão de submissão mais elevada da cópia de trabalho, AND com o valor após o carácter <code>&amp;</code> . Por exemplo: <code>\$WCREV&amp;0xFFFF\$</code>
<code>\$WCREV-\$</code> , <code>\$WCREV+\$</code>	Substituída com a revisão de submissão mais elevada da cópia de trabalho, com o valor adicionado ou subtraído após o carácter <code>+</code> ou <code>-</code> . Por exemplo: <code>\$WCREV-1000\$</code>
<code>\$WCDATE\$</code> , <code>\$WCDATEUTC\$</code>	Remover ficheiros e pastas ignorados e não-versionados
<code>\$WCNOW\$</code> , <code>\$WCNOWUTC\$</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 revisão, actualizada na cópia de trabalho. Se a cópia de trabalho está num estado inconsistente, esta será apenas uma única revisão. Se a cópia de trabalho contem uma miscelânea de revisões, seja por estar desactualizada ou devido a uma actualização para uma revisão deliberada, então o intervalo de revisões será mostrado no formato <code>100:200</code> .
<code>\$WCMIXED\$</code>	<code>\$WCMIXED?TText:FText\$</code> é substituída por <code>TText</code> se existem revisões de actualização misturadas, ou <code>FText</code> se não for o caso.
<code>\$WCMODS\$</code>	<code>\$WCMODS?TText:FText\$</code> é substituída por <code>TText</code> se existem alterações locais, ou <code>FText</code> se não for o caso.
<code>\$WCUNVER\$</code>	<code>\$WCUNVER?TText:FText\$</code> is replaced with <code>TText</code> if there are unversioned items in the working copy, or <code>FText</code> if not.
<code>\$WCEXTALLFIXED\$</code>	<code>\$WCEXTALLFIXED?TText:FText\$</code> is replaced with <code>TText</code> if all externals are fixed to an explicit revision, or <code>FText</code> if not.
<code>\$WCISTAGGED\$</code>	<code>\$WCISTAGGED?TText:FText\$</code> is replaced with <code>TText</code> if the repository URL contains the tags classification pattern, or <code>FText</code> if not.
<code>\$WCURL\$</code>	Substituída pelo URL do repositório correspondente ao caminho da cópia de trabalho passada ao SubWCRev.
<code>\$WCINSVN\$</code>	<code>\$WCINSVN?TText:FText\$</code> é substituída pelo <code>TText</code> se a entrada é versionada, ou <code>FText</code> se não o é.
<code>\$WCNEEDSLOCK\$</code>	<code>\$WCNEEDSLOCK?TText:FText\$</code> é substituída por <code>TText</code> se a entrada tiver a propriedade <code>svn:needs-lock</code> configurada, ou por <code>FText</code> se não for o caso.
<code>\$WCISLOCKED\$</code>	<code>\$WCISLOCKED?TText:FText\$</code> é substituída por <code>TText</code> se a entrada estiver bloqueada, ou por <code>FText</code> se não o estiver.
<code>\$WCLOCKDATE\$</code> , <code>\$WCLOCKDATEUTC\$</code>	Substituída pela data de bloqueio. A formatação do tempo poderá ser usada de acordo com o descrito para o <code>\$WCDATE\$</code> .

Palavra-chave	Descrição
\$WCLOCKOWNER\$	Substituída pelo nome do autor do bloqueio.
\$WCLOCKCOMMENT\$	Substituída pelo comentário do bloqueio.
\$WCUNVER\$	\$WCUNVER?TText:FText\$ is replaced with TText if there are unversioned files or folders in the working copy, or FText if not.

**Tabela 6.3. Lista de palavras-chave disponíveis**

SubWCRev does not directly support nesting of expressions, so for example you cannot use an expression like:

```
#define SVN_REVISION      "$WCMIXED?$WCRANGE$: $WCREV$$ "
```

But you can usually work around it by other means, for example:

```
#define SVN_RANGE        $WCRANGE$
#define SVN_REV          $WCREV$
#define SVN_REVISION     "$WCMIXED?SVN_RANGE:SVN_REV$ "
```



### Dica

Algumas dessas palavras-chave aplicam-se 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\$, \$WCISLOCKED\$, \$WCLOCKDATE\$, \$WCLOCKOWNER\$ e \$WCLOCKCOMMENT\$.

## 6.3. Exemplo de Palavra-Chave

O exemplo abaixo mostra como palavras-chave num ficheiro template são substituídas no ficheiro de saída.

```
// Test file for SubWCRev

char *Revision          = "$WCREV$";
char *Revision16       = "$WCREV&0xFF$";
char *Revisionp100    = "$WCREV+100$";
char *Revisionm100    = "$WCREV-100$";
char *Modified         = "$WCMODS?Modified:Not modified$";
char *Unversioned     = "$WCUNVER?Unversioned items found:no unversioned items$";
char *Date             = "$WCDATE$";
char *CustDate        = "$WCDATE=%a, %d %B %Y$";
char *DateUTC         = "$WCDATEUTC$";
char *CustDateUTC     = "$WCDATEUTC=%a, %d %B %Y$";
char *TimeNow         = "$WCNOW$";
char *TimeNowUTC      = "$WCNOWUTC$";
char *RevRange        = "$WCRANGE$";
char *Mixed           = "$WCMIXED?Mixed revision WC:Not mixed$";
char *ExtAllFixed     = "$WCEXTALLFIXED?All externals fixed:Not all externals fixed$";
char *IsTagged        = "$WCISTAGGED?Tagged:Not tagged$";
char *URL             = "$WCURL$";
char *isInSVN         = "$WCINSVN?versioned:not versioned$";
char *needslock       = "$WCNEEDSLOCK?TRUE:FALSE$";
char *islocked        = "$WCISLOCKED?locked:not locked$";
```

```

char *lockdateutc = "$WCLOCKDATEUTC$";
char *lockdate = "$WCLOCKDATE$";
char *lockcustutc = "$WCLOCKDATEUTC=%a, %d %B %Y$";
char *lockcust = "$WCLOCKDATE=%a, %d %B %Y$";
char *lockown = "$WCLOCKOWNER$";
char *lockcmt = "$WCLOCKCOMMENTS$";

```

```

#if $WCMODS?1:0$
#error Source is modified
#endif

```

```
// End of file
```

Depois de correr o SubWCRev.exe path\to\workingcopy testfile.tmpl testfile.txt, o ficheiro de saída testfile.txt irá ficar assim:

```
// Test file for SubWCRev
```

```

char *Revision = "22837";
char *Revision16 = "53";
char *Revisionp100 = "22937";
char *Revisionm100 = "22737";
char *Modified = "Modified";
char *Unversioned = "no unversioned items";
char *Date = "2012/04/26 18:47:57";
char *CustDate = "Thu, 26 April 2012";
char *DateUTC = "2012/04/26 16:47:57";
char *CustDateUTC = "Thu, 26 April 2012";
char *TimeNow = "2012/04/26 20:51:17";
char *TimeNowUTC = "2012/04/26 18:51:17";
char *RevRange = "22836:22837";
char *Mixed = "Mixed revision WC";
char *ExtAllFixed = "All externals fixed";
char *IsTagged = "Not tagged";
char *URL = "https://svn.code.sf.net/p/tortoisesvn/code/trunk";
char *isInSVN = "versioned";
char *needslock = "FALSE";
char *islocked = "not locked";
char *lockdateutc = "1970/01/01 00:00:00";
char *lockdate = "1970/01/01 01:00:00";
char *lockcustutc = "Thu, 01 January 1970";
char *lockcust = "Thu, 01 January 1970";
char *lockown = "";
char *lockcmt = "";

```

```

#if 1
#error Source is modified
#endif

```

```
// End of file
```



## 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.

## 6.4. 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
<code>.GetWCInfo</code>	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 <code>true</code> se queres incluir as revisões das pastas. Equivalente à opção de linha de comando <code>-f</code> . O terceiro parâmetro deverá ser <code>true</code> se queres incluir <code>svn:externals</code> . Equivalente à opção de linha de comando <code>-e</code> .
<code>.GetWCInfo2</code>	O mesmo que <code>GetWCInfo()</code> mas com um quarto parâmetro que fixa o equivalente ao switch de linha de comando <code>-E</code> .
<code>.Revision</code>	A mais alta revisão submetida, na cópia de trabalho. Equivalente ao <code>\$WCREV\$</code> .
<code>.Date</code>	A data/tempo de submissão da mais alta revisão submetida. Equivalente ao <code>\$WCDATE\$</code> .
<code>.Author</code>	O autor da mais alta revisão de submissão, isto é, a última pessoa que submeteu alterações na cópia de trabalho.
<code>.MinRev</code>	A revisão de actualização mínima, como mostrada em <code>\$WCRANGE\$</code>
<code>.MaxRev</code>	A revisão de actualização máxima, como mostrado em <code>\$WCRANGE\$</code>
<code>.HasModifications</code>	<code>True</code> se existem modificações locais
<code>.HasUnversioned</code>	<code>True</code> if there are unversioned items
<code>.Url</code>	Substituído com o URL do repositório do caminho de cópia de trabalho usada em <code>GetWCInfo</code> . Equivalente a <code>\$WCURL\$</code> .
<code>.IsSvnItem</code>	<code>True</code> se o item é versionado.
<code>.NeedsLocking</code>	<code>True</code> se o item tem a propriedade <code>svn:needs-lock</code> configurada.
<code>.IsLocked</code>	<code>True</code> se o item está bloqueado.
<code>.LockCreationDate</code>	String que representa a data em que o bloqueio foi criado, ou uma string vazia se o item não está bloqueado.
<code>.LockOwner</code>	String que representa o autor do bloqueio, ou uma string vazia se o item não está bloqueado.
<code>.LockComment</code>	A mensagem introduzida quando o bloqueio foi criado.

**Tabela 6.4. Métodos COM/automação suportados**

O seguinte exemplo mostra como a interface pode ser usada.

```
// testCOM.js - ficheiro javascript
// script de teste para 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.GetWCInfo2(
    filesystem.GetAbsolutePathName("../.."), 1, 1, 1);

wcInfoString1 = "Revision = " + revObject1.Revision +
    "\nMin Revision = " + revObject1.MinRev +
    "\nMax Revision = " + revObject1.MaxRev +
    "\nDate = " + revObject1.Date +
    "\nURL = " + revObject1.Url + "\nAuthor = " +
    revObject1.Author + "\nHasMods = " +
    revObject1.HasModifications + "\nIsSvnItem = " +
    revObject1.IsSvnItem + "\nNeedsLocking = " +
    revObject1.NeedsLocking + "\nIsLocked = " +
    revObject1.IsLocked + "\nLockCreationDate = " +
    revObject1.LockCreationDate + "\nLockOwner = " +
    revObject1.LockOwner + "\nLockComment = " +
    revObject1.LockComment;

wcInfoString2 = "Revision = " + revObject2.Revision +
    "\nMin Revision = " + revObject2.MinRev +
    "\nMax Revision = " + revObject2.MaxRev +
    "\nDate = " + revObject2.Date +
    "\nURL = " + revObject2.Url + "\nAuthor = " +
    revObject2.Author + "\nHasMods = " +
    revObject2.HasModifications + "\nIsSvnItem = " +
    revObject2.IsSvnItem + "\nNeedsLocking = " +
    revObject2.NeedsLocking + "\nIsLocked = " +
    revObject2.IsLocked + "\nLockCreationDate = " +
    revObject2.LockCreationDate + "\nLockOwner = " +
    revObject2.LockOwner + "\nLockComment = " +
    revObject2.LockComment;

wcInfoString3 = "Revision = " + revObject3.Revision +
    "\nMin Revision = " + revObject3.MinRev +
    "\nMax Revision = " + revObject3.MaxRev +
    "\nDate = " + revObject3.Date +
    "\nURL = " + revObject3.Url + "\nAuthor = " +
    revObject3.Author + "\nHasMods = " +
    revObject3.HasModifications + "\nIsSvnItem = " +
    revObject3.IsSvnItem + "\nNeedsLocking = " +
    revObject3.NeedsLocking + "\nIsLocked = " +
    revObject3.IsLocked + "\nLockCreationDate = " +
    revObject3.LockCreationDate + "\nLockOwner = " +
    revObject3.LockOwner + "\nLockComment = " +
    revObject3.LockComment;

wcInfoString4 = "Revision = " + revObject4.Revision +
    "\nMin Revision = " + revObject4.MinRev +
    "\nMax Revision = " + revObject4.MaxRev +
    "\nDate = " + revObject4.Date +
    "\nURL = " + revObject4.Url + "\nAuthor = " +
    revObject4.Author + "\nHasMods = " +
    revObject4.HasModifications + "\nIsSvnItem = " +
    revObject4.IsSvnItem + "\nNeedsLocking = " +
    revObject4.NeedsLocking + "\nIsLocked = " +
    revObject4.IsLocked + "\nLockCreationDate = " +
    revObject4.LockCreationDate + "\nLockOwner = " +
```

```
revObject4.LockOwner + "\nLockComment = " +  
revObject4.LockComment;
```

```
WScript.Echo(wcInfoString1);  
WScript.Echo(wcInfoString2);  
WScript.Echo(wcInfoString3);  
WScript.Echo(wcInfoString4);
```

A seguinte listagem é um exemplo de como usar o objecto COM SubWCRev a partir de C#:

```
using LibSubWCRev;  
SubWCRev sub = new SubWCRev();  
sub.GetWCInfo("C:\\PathToMyFile\\MyFile.cc", true, true);  
if (sub.IsSvnItem == true)  
{  
    MessageBox.Show("versionado");  
}  
else  
{  
    MessageBox.Show("nao versionado");  
}
```

---

# Capítulo 7. 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#, na pasta `contrib/issue-tracker-plugins` do no nosso repositório. Nessa pasta poderás também encontrar os ficheiros de include reueridos para construíres o teu plugin. . (Secção 3, “Licença” explica como aceder ao repositório.)



## Importante

Deverás fornecer ambas as versões 32-bit e 64-bit do teu plugin. Porque a versão-x64 do TortoiseSVN não pode usar um plugin 32-bit e vice-versa.

## 7.1. Convenções de nomenclatura

Se publicares um plugin de rastreador de problemas para o TortoiseSVN, por favor *não* lhe dês o nome de *Tortoise<QualquerCoisa>*. Gostaríamos de reservar o prefixo *Tortoise* para um cliente de controlo de versões integrado na shell do Windows. Por exemplo: TortoiseCVS, TortoiseSVN, TortoiseHg, TortoiseGit e o TortoiseBzr são todos clientes de controlo de versões.

Dá, por favor, um nome ao teu plugin de cliente Tortoise *Tartaruga <something>*, onde *<something>* será referente ao controlador de problemas ao qual estás ligado. Em alternativa, escolhe um nome que soe como *Tartaruga* mas tenha uma primeira letra diferente. Bons exemplos são:

- Grutle - Um plugin de rastreador de problemas para o código Google
- TurtleMine - Um plugin de rastreador de problemas para o Redmine
- VurtleOne - Um plugin de rastreador de problemas para o VersionOne

## 7.2. O interface do IBugtraqProvider

O TortoiseSVN 1.5 e superior pode usar os plugins que implementam a interface IBugtraqProvider. A interface providencia alguns métodos que os plugins podem usar para interagir com o seguidor 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  
);
```

Este método é chamado a partir da caixa de diálogo preferências, onde o utilizador pode adicionar e configurar o plugin. A string `parameters` pode ser usada pelo plugin para obter informação adicional requerida, e.g., o URL para o seguidor de problemas, informação de login, etc. O plugin deverá verificar a string `parameters` e mostrar uma mensagem de erro se a string não é valida. O parâmetro `hParentWnd` deverá ser usado por qualquer

caixa de diálogo que o plugin mostre, como a janela pai. O plugin deverá retornar TRUE se a validação da string `parameters` for efectuada com sucesso. Se o plugin retornar FALSE, a janela de preferências não permitirá ao utilizador adicionar o plugin ao caminho da cópia de trabalho.

```
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 providenciar aqui uma string, que é usada pela caixa de diálogo de submissão do TortoiseSVN, para o botão que invoca o plugin, e.g., "Escolhe o problema" ou "Selecciona bilhete". Tem a certeza que a string não é muito longa, de outro modo poderá não caber no botão. Se o método retornar um erro (e.g., `E_NOTIMPL`), o texto por defeito será usado 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  
);
```

Este é o método principal do plugin. Este método é chamado a partir da caixa de diálogo submeter do TortoiseSVN, quando o utilizador clica no botão do plugin.

A string `parameters` é a string que o utilizador tem de introduzir na caixa de diálogo preferências, quando configura o plugin. Normalmente o plugin irá fazer uso disto para encontrar o URL do seguidor de problemas e/ou informação de login ou outros.

A string `commonRoot` contém o caminho pai de todos os itens seleccionados, ao abrir a caixa de diálogo submeter. Ter em atenção que este *não* é o caminho raiz de todos os itens que o utilizador seleccionou na janela de submissão. Para a caixa de diálogo ramo/etiqueta, este é o caminho que será copiado.

O parâmetro `pathList` contém um array de caminhos (como strings) que o utilizador tem de seleccionar para a submissão.

O parâmetro `originalMessage` contém o texto introduzido na caixa de diálogo mensagem de registo, na caixa de diálogo submeter. Se o utilizador não introduziu ainda qualquer texto, a string estará vazia.

A string de retorno `newMessage` é copiada para a caixa de edição mensagem de registo na caixa de diálogo submeter, substituindo o que já lá esteja. Se o plugin não modificar a string `originalMessage`, deverá retornar de novo a mesma string, de outro modo qualquer texto que o utilizador introduziu será perdido.

### 7.3. 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  
);
```

Este método é chamado a partir da caixa de diálogo submeter do TortoiseSVN, quando o utilizador clica no botão do plugin. Este método é chamado em vez do `GetCommitMessage()`. Consultar, por favor, a documentação do `GetCommitMessage` para informação sobre os parâmetros que também são usados aqui.

O parâmetro `commonURL` representa o URL pai de todos os itens seleccionados para serem mostrados na caixa de diálogo submeter. Isto é basicamente o URL do caminho `commonRoot`.

O parâmetro `bugID` contém o conteúdo do campo bug-ID ( se for mostrado, configurado com a propriedade `bugtraq:message`).

O parâmetro de retorno `bugIDOut` é usado para preencher o campo bug-ID, quando o método retorna.

Os parâmetros de retorno `revPropNames` e `revPropValues` podem conter pares nome/valor de propriedades de revisão que a submissão deverá usar. Um plugin deverá ter a certeza que ambos os arrays têm o mesmo tamanho, ao retornar! Cada nome de propriedade no `revPropNames` deverá ter também o correspondente valor no `revPropValues`. Se não forem configuradas nenhuma propriedade de revisão, o plugin deverá retornar arrays vazios.

```
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
);
```

Este método é chamado, mesmo antes de ser fechada a caixa de diálogo submeter, e a submissão começar. O plugin pode usar este método para validar os ficheiros/pastas seleccionados para a submissão e/ou a mensagem de submissão introduzida pelo utilizador. Os parâmetros são os mesmos do `GetCommitMessage2()`, com a diferença que o `commonURL` é agora o URL comum de todos os itens *verificados*, e o `commonRoot` o caminho raiz de todos os itens verificados.

Para a caixa de diálogo ramo/etiqueta, o `commonURL` é o URL fonte da cópia, e o `commonRoot` é configurado para o URL destino da cópia.

O parâmetro de retorno `errorMessage` deverá conter uma mensagem de erro, que o TortoiseSVN mostra ao utilizador, ou então ser vazio para começar a submissão. Se for retornada uma mensagem de erro, o TortoiseSVN mostra a mensagem numa caixa de diálogo e mantém a caixa de diálogo submeter aberta, para que o utilizador possa corrigir o que quer que esteja errado. Um plugin deverá por isso retornar uma string que informe o utilizador *do que está errado e como o corrigir*.

```
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-lhe informação sobre a submissão. Os parâmetros são os mesmos do `GetCommitMessage2`.

```
HRESULT HasOptions(  
    // Para quando o fornecedor providenciar opções  
    [out, retval] VARIANT_BOOL *ret  
);
```

Este método é chamado a partir da caixa de diálogo preferências, onde o utilizador pode configurar os plugins. Se um plugin fornecer a sua própria caixa de diálogo de configuração com o `ShowOptionsDialog`, este deverá retornar `TRUE` aqui, ou de outro modo retornar `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  
);
```

Este método é chamado a partir da caixa de diálogo preferências quando o utilizador clica no botão "Opções", que é mostrado se a `HasOptions` retorna `TRUE`. Um plugin poderá mostrar uma caixa de diálogo opções, de modo a facilitar ao utilizador a configuração do plugin.

A string `parameters` contém a string dos parâmetros do plugin que forma já configurados/introduzidos.

O parâmetro de retorno `newparameters` deverá conter a string de parâmetros que o plugin construiu a partir da informação recolhida na sua caixa de diálogo opções. Essa string `parameters` será passada a todos os outros métodos `IBugtraqProvider` e `IBugtraqProvider2`.

---

# Apêndice A. Questões Mais Frequentes (FAQ)

Because TortoiseSVN is being developed all the time it is sometimes hard to keep the documentation completely up to date. We maintain an *online FAQ* [<https://tortoisesvn.net/faq.html>] which contains a selection of the questions we are asked the most on the TortoiseSVN mailing lists <dev@tortoisesvn.tigris.org> and <users@tortoisesvn.tigris.org>.

We also maintain a project *Issue Tracker* [<https://sourceforge.net/p/tortoisesvn/tickets/>] which tells you about some of the things we have on our To-Do list, and bugs which have already been fixed. If you think you have found a bug, or want to request a new feature, check here first to see if someone else got there before you.

Se tens uma questão que ainda não foi respondida em nenhum lado, o melhor local para a colocar será numa das listas de correio:

- <users@tortoisesvn.tigris.org> será aquela a usar se tens questões sobre como utilizar o TortoiseSVN.
- Se necessitas de ajuda com o desenvolvimento do TortoiseSVN então deverás tomar parte nas discussões da lista <dev@tortoisesvn.tigris.org>.
- Se queres ajudar na tradução do interface de utilizador do TortoiseSVN ou da documentação, envia um e-mail para <translators@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 à direita os ficheiros para uma nova localização. Simplesmente clica à direita nos ficheiros que queres mover/copiar, isto sem largar o botão do rato. Arrasta então os ficheiros para uma nova localização e largando em seguida o botão do rato. O menu de contexto aparecerá, onde então poderás escolher Menu de Contexto → SVN Copiar para aqui arquivos versionados ou Menu de Contexto → SVN Mover para aqui item(s) versionados.

## 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, consulta por favor [Secção 4.18, “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

De longe a maneira mais fácil para reverter alterações de uma ou mais revisões, é usar a caixa de diálogo registo de revisões.

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. Select the revision you wish to revert. If you want to undo a range of revisions, select the first one and hold the **Shift** key while selecting the last one. If you want to pick out individual revisions and ranges, use the **Ctrl** key while selecting revisions. Right click on the selected revision(s), then select Context Menu → Revert changes from this revision.
4. 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

Se desejas inserir numeros de revisões como uma lista, podes usar a caixa de diálogo Integrar. O método anterior usa integração nos bastidores; este método usa-a explicitamente.

1. Na tua cópia de trabalho selecciona TortoiseSVN → Integrar.
2. Na caixa de diálogo Tipo Integração selecciona Integra um intervalo de revisões.
3. In the From: field enter the full repository URL of your working copy folder. This should come up as the default URL.
4. In the Revision range to merge field enter the list of revisions to roll back (or use the log dialog to select them as described above).
5. Tem a certeza que a caixa Integração inversa está verificada.
6. Na caixa de diálogo Opções de integração aceita os valores por defeito.
7. Clica Integrar para completar a integração.

Reverteste as alterações na tua cópia de trabalho. Verifica se os resultados são os expectáveis e submete 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.

The only way to remove data from the repository is to use the Subversion command line tool `svnadmin`. You can find a description of how this works in the [Repository Maintenance](http://svnbook.red-bean.com/en/1.8/svn.reposadmin.maint.html) [http://svnbook.red-bean.com/en/1.8/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 4.11.3, “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

Sometimes you will want to include another project within your working copy, perhaps some library code. There are at least 4 ways of dealing with this.

### B.6.1. 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 queres usar como pasta de checkout para código comum, e o URL do repositório do qual queres efectuar aqui checkout. Para mais detalhes consultar [Secção 4.19, “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, qualquer alteração que lá fizeres, será incluída na lista de submissões, quando submeteres o teu projecto principal.

If the external project is in a different repository, any changes you make to the external project will be shown or indicated when you commit the main project, but you have to commit those external changes separately.

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

If you use the same common core code in several projects, and you do not want to keep multiple working copies of it for every project that uses it, you can just check it out to a separate location which is related to all the other projects which use it. For example:

```
C:\Projects\Proj1
C:\Projects\Proj2
C:\Projects\Proj3
C:\Projects\Common
```

and refer to the common code using a relative path, e.g. `..\..\Common\DSPcore`.

If your projects are scattered in unrelated locations you can use a variant of this, which is to put the common code in one location and use drive letter substitution to map that location to something you can hard code in your projects, e.g. Checkout the common code to `D:\Documents\Framework` or `C:\Documents and Settings\{login}\My Documents\framework` then use

```
SUBST X: "D:\Documents\framework"
```

to create the drive mapping used in your source code. Your code can then use absolute locations.

```
#include "X:\superio\superio.h"
```

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.6.4. Adicionar projecto ao repositório

The maybe easiest way is to simply add the project in a subfolder to your own project working copy. However this has the disadvantage that you have to update and upgrade this external project manually.

To help with the upgrade, TortoiseSVN provides a command in the explorer right-drag context menu. Simply right-drag the folder where you unzipped the new version of the external library to the folder in your working copy, and then select Context Menu → SVN Vendorbranch here. This will then copy the new files over to the target folder while automatically adding new files and removing files that aren't in the new version anymore.

## B.7. Criar um atalho para um repositório

If you frequently need to open the repository browser at a particular location, you can create a desktop shortcut using the automation interface to TortoiseProc. Just create a new shortcut and set the target to:

```
TortoiseProc.exe /command:repobrowser /path:"url/to/repository"
```

Of course you need to include the real repository URL.

## B.8. 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 estes não sejam novamente adicionados por engano.

Se os ficheiros já estiverem no repositório, então terão de ser removidos do repositório e adicionados à lista de ignorados. Felizmente o TortoiseSVN tem um atalho conveniente para efectuar isto. TortoiseSVN → Desversionar e adicionar à lista de ignorados irá primeiro marcar o ficheiro para remoção do repositório, mantendo a cópia local. Também adiciona este item à lista de ignorados para que não seja adicionado novamente, por engano, ao Subversion. Uma vez que isto esteja feito, só precisas de submeter a pasta pai.

## B.9. Remover uma cópia de trabalho do controlo de versões

If you have a working copy which you want to convert back to a plain folder tree without the `.svn` directory, you can simply export it to itself. Read [Secção 4.27.1, “Remover uma cópia de trabalho do controlo de versões”](#) to find out how.

## 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 - apaga-a apenas no Explorador do Windows! As cópias de trabalho são entidades locais privadas, auto contidas. Apagar uma cópia de trabalho no Explorador do Windows não afecta de modo algum os dados no repositório.

---

# 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.

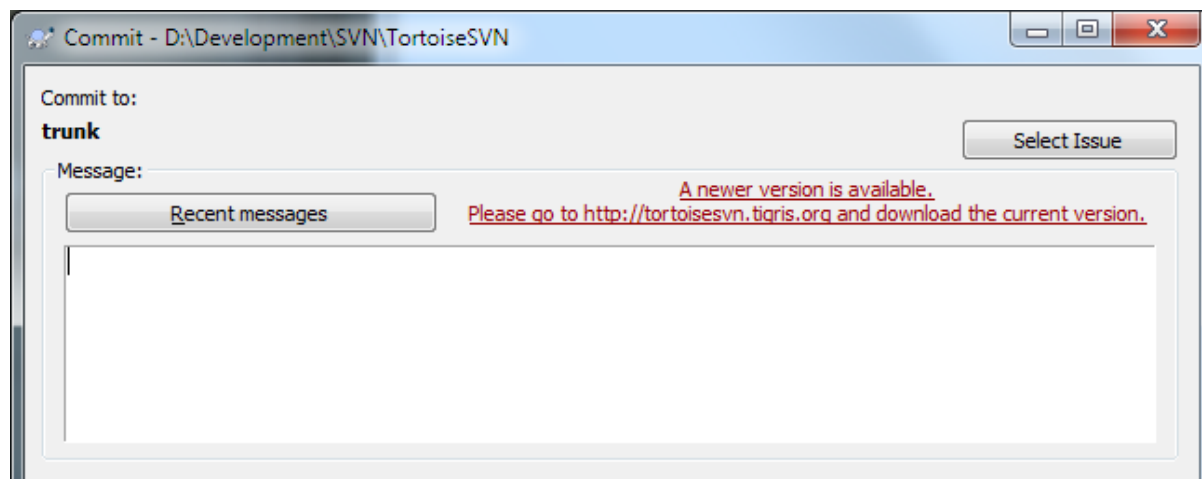
Uma boa abordagem de como fazer isso poderá ser encontrada no artigo 314934 da base de conhecimento da Microsoft: <http://support.microsoft.com/?kbid=314934>.

TortoiseSVN must be installed under *Computer Configuration* and not under *User Configuration*. This is because TortoiseSVN needs the CRT and MFC DLLs, which can only be deployed *per computer* and not *per user*. If you really must install TortoiseSVN on a per user basis, then you must first install the MFC and CRT package version 12 from Microsoft on each computer you want to install TortoiseSVN as per user.

You can customize the MSI file if you wish so that all your users end up with the same settings. TSVN settings are stored in the registry under `HKEY_CURRENT_USER\Software\TortoiseSVN` and general Subversion settings (which affect all Subversion clients) are stored in config files under `%APPDATA%\Subversion`. If you need help with MSI customization, try one of the MSI transform forums or search the web for “MSI transform”.

## C.2. Redireccionar a verificação de actualização

O TortoiseSVN verifica se existe uma nova versão disponível, com a periodicidade de alguns dias. Se existir disponível uma nova versão, uma notificação é mostrada na caixa de diálogo submeter.



**Figura C.1. A caixa de diálogo submeter, mostrando a notificação de actualização**

Se és responsável por muitos utilizadores num domínio, poderás desejar que os teus utilizadores apenas usem versões que aprovaste e não tenham sempre instalada a última versão. Provavelmente não queres que apareça a caixa de diálogo de actualização, para que os teus utilizadores não vão actualizar imediatamente.

Versions 1.4.0 and later of TortoiseSVN allow you to redirect that upgrade check to your intranet server. You can set the registry key `HKCU\Software\TortoiseSVN\UpdateCheckURL` (string value) to an URL pointing to a text file in your intranet. That text file must have the following format:

1.9.1.6000

A new version of TortoiseSVN is available for you to download!  
<http://192.168.2.1/downloads/TortoiseSVN-1.9.1.6000-svn-1.9.1.msi>

The first line in that file is the version string. You must make sure that it matches the exact version string of the TortoiseSVN installation package. The second line is a custom text, shown in the commit dialog. You can write there whatever you want. Just note that the space in the commit dialog is limited. Too long messages will get truncated! The third line is the URL to the new installation package. This URL is opened when the user clicks on the custom message label in the commit dialog. You can also just point the user to a web page instead of the MSI file directly. The URL is opened with the default web browser, so if you specify a web page, that page is opened and shown to the user. If you specify the MSI package, the browser will ask the user to save the MSI file locally.

### C.3. Configurar a variável de ambiente SVN\_ASP\_DOT\_NET\_HACK

Como na versão 1.4.0 e superior, o instalador do TortoiseSVN não fornece mais ao utilizador a opção de configurar a variável de ambiente SVN\_ASP\_DOT\_NET\_HACK, visto que essa causava muitos problemas e confusão aos utilizadores que instalam sempre *tudo* independentemente de saberem para que é que serve.

But the feature is still available in TortoiseSVN and other svn clients. To enable it you have to set the Windows environment variable named ASPDOTNETHACK to 1. Actually, the value of that environment variable doesn't matter: if the variable exists the feature is active.



#### Importante

Toma nota que este hack só é necessário se estiveres ainda a utilizar o VS.NET2002. Todas as versões posteriores do Visual Studio *não* requerem que este hack esteja activo! Pelo que a não ser que estejas a usar uma ferramenta antiga, NÃO USES ISTO!

### 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
0x0000000100000000	Integrar Todos
0x0000000200000000	Comparar com a revisão anterior
0x0000000400000000	Colar
0x0000000800000000	Actualizar cópia de trabalho
0x0000001000000000	Diff mais tarde
0x0000002000000000	Diff with 'filename'
0x0000004000000000	Unified diff
0x2000000000000000	Preferências
0x4000000000000000	Ajuda
0x8000000000000000	Sobre

**Tabela C.1. Entradas de menu e seus valores**

Example: to disable the “Relocate” the “Delete unversioned items” and the “Settings” menu entries, add the values assigned to the entries like this:

```
0x0000000000008000
+ 0x0000000080000000
+ 0x2000000000000000
```



= 0x2000000080080000

The lower DWORD value (0x80080000) must then be stored in HKEY\_CURRENT\_USER\Software\TortoiseSVN\ContextMenuEntriesMaskLow, the higher DWORD value (0x20000000) in HKEY\_CURRENT\_USER\Software\TortoiseSVN\ContextMenuEntriesMaskHigh.

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"`.

There's a special command that does not require the parameter `/command:abcd` but, if nothing is specified on the command line, starts the project monitor instead. If `/tray` is specified, the project monitor starts hidden and only adds its icon to the system tray.

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 carácter `*`.

You can also specify a file which contains a list of paths, separated by newlines. The file must be in UTF-16 format, without a *BOM* [[https://en.wikipedia.org/wiki/Byte-order\\_mark](https://en.wikipedia.org/wiki/Byte-order_mark)]. If you pass such a file, use `/pathfile` instead of `/path`. To have TortoiseProc delete that file after the command is finished, you can pass the parameter `/deletepathfile`. If you don't pass `/deletepathfile`, you have to delete the file yourself or the file gets left behind.

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 a caixa de diálogo de progresso automaticamente no fim de um comando, sem usar a preferência permanente, tu 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

Para fechar a caixa de diálogo de progresso para operações locais, se não existem erros ou conflitos, passa o parâmetro `/closeforlocal`.

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	<p>Opens the log dialog. The /path specifies the file or folder for which the log should be shown. Additional options can be set:</p> <ul style="list-style-type: none"> <li>• /startrev:xxx,</li> <li>• /endrev:xxx,</li> <li>• /strict enables the 'stop-on-copy' checkbox,</li> <li>• /merge enables the 'include merged revisions' checkbox,</li> <li>• /datemin: "{datestring}" sets the start date of the filter, and</li> <li>• /datemax: "{datestring}" sets the end date of the filter. The date format is the same as used for svn date revisions.</li> <li>• /findstring: "filterstring" fills in the filter text,</li> <li>• /findtext forces the filter to use text, not regex, or</li> <li>• /findregex forces the filter to use regex, not simple text search, and</li> <li>• /findtype:X with X being a number between 0 and 511. The numbers are the sum of the following options: <ul style="list-style-type: none"> <li>• /findtype:0 filtrar por tudo</li> <li>• /findtype:1 filtrar por mensagens</li> <li>• /findtype:2 filtrar por caminho</li> <li>• /findtype:4 filtrar por autores</li> <li>• /findtype:8 filtrar por revisões</li> <li>• / Findtype: 16 não utilizado</li> <li>• /findtype:32 filtrar por ID de bug</li> <li>• / Findtype: 64 não usado</li> <li>• / Findtype: 128 filtrar por data</li> <li>• / Findtype: 256 filtrar por intervalo de datas</li> </ul> </li> <li>• If /outfile:path\to\file is specified, the selected revisions are written to that file when the log dialog is closed. The revisions are written in the same format as is used to specify revisions in the merge dialog.</li> </ul> <p>An svn date revision can be in one of the following formats:</p> <ul style="list-style-type: none"> <li>• {2006-02-17}</li> <li>• {15:30}</li> </ul>

Comando	Descrição
	<ul style="list-style-type: none"> <li>• {15:30:00.200000}</li> <li>• {"2006-02-17 15:30"}</li> <li>• {"2006-02-17 15:30 +0230"}</li> <li>• {2006-02-17T15:30}</li> <li>• {2006-02-17T15:30Z}</li> <li>• {2006-02-17T15:30-04:00}</li> <li>• {20060217T1530}</li> <li>• {20060217T1530Z}</li> <li>• {20060217T1530-0500}</li> </ul>
:checkout	Abre a caixa de diálogo checkout. O /path especifica a pasta destino e o /url especifica o URL de onde efectuar o checkout. Se especificares a chave /blockpathadjustments, são bloqueados os ajustamentos automáticos dos caminhos de checkout. O /revision:XXX especifica a revisão a efectuar checkout.
:import	Abre a caixa de diálogo importar. O /path especifica a pasta com os dados a importar. Podes também especificar a opção /logmsg para passar uma mensagem de registo pré-definida para a caixa de diálogo importar. Ou, senão desejas passar a mensagem de registopela linha de comandos, usa /logmsgfile:path, onde caminho aponta para um ficheiro que contém a mensagem de registo.
:update	Updates the working copy in /path to HEAD. If the option /rev 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 /rev:1234. Other options are /nonrecursive, /ignoreexternals and /includeexternals. The /stickydepth indicates that the specified depth should be sticky, creating a sparse checkout. The /skipprechecks can be set to skip all checks that are done before an update. If this is specified, then the <b>Mostrar registo</b> button is disabled, and the context menu to show diffs is also disabled after the update.
:commit	Abre a caixa de diálogo submeter. O /path especifica a pasta de destino ou a lista de ficheiros a submeter. Tu podes também especificar a opção /logmsg 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 /logmsgfile:path, 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 /bugid:"o id do bug aqui".
:add	Adiciona ao controlo de versões os ficheiros em /path
:revert	Reverte as modificações locais de uma cópia de trabalho. O /path diz quais os itens a reverter.
:cleanup	Cleans up interrupted or aborted operations and unlocks the working copy in /path. You also have to pass the /cleanup to actually do the cleanup. Use /noui to prevent the result dialog from popping up (either telling about the cleanup being finished or showing an error message). /noprogessui also disables the progress dialog. /nodlg disables showing the cleanup dialog where the user can choose what exactly should be done in the cleanup. The available actions can be specified with the options /cleanup for status cleanup, /breaklocks to break all locks, /revert to revert uncommitted changes, /delunversioned,

Comando	Descrição
	/deligned, /refreshshell, /externals, /fixtimestamps and /vacuum.
:resolve	Marca um ficheiro em conflito em /path como resolvido. Se for dado o /noquestion então a resolução é feita sem perguntar primeiro ao utilizador se ela deverá ser mesmo feita.
:reprocreate	Cria um repositório em /path
:switch	Abre a caixa de diálogo trocar. O /path especifica a pasta de destino e o /url o URL de destino da troca.
:export	Exporta a cópia de trabalho em /path para outra pasta. Se o /path apontar para uma pasta não versionada, a caixa de diálogo perguntará por um URL para exportar para a pasta em /path. Se especificares a chave /blockpathadjustments, são bloqueados os ajustamentos automáticos de caminhos de exportação.
:dropexport	Exporta a cópia de trabalho em /path para a pasta especificada em /droptarget. Esta exportação não usa a caixa de diálogo exportar, mas executa-a directamente. A opção /overwrite indica que os ficheiros existentes são sobrepostos sem informação ao utilizador, e a opção /autorename indica que se os ficheiros já existirem, os ficheiros exportados serão automaticamente renomeados para evitar sobreposições. A opção /extended pode indicar localchanges para só exportar ficheiros que foram alterados localmente, ou não-versionado exportar também todos os itens não versionados.
:dropvndor	Copies the folder in /path recursively to the directory specified in /droptarget. New files are added automatically, and missing files get removed in the target working copy, basically ensuring that source and destination are exactly the same. Specify /noui to skip the confirmation dialog, and /noprogessui to also disable showing the progress dialog.
:merge	Opens the merge dialog. The /path specifies the target directory. For merging a revision range, the following options are available: /fromurl:URL, /revrange:string. For merging two repository trees, the following options are available: /fromurl:URL, /tourl:URL, /fromrev:xxx and /torev:xxx.
:mergeall	Abre a caixa de diálogo integrar todos. O /path especifica a pasta de destino.
:copy	Brings up the branch/tag dialog. The /path is the working copy to branch/tag from. And the /url is the target URL. If the urls starts with a ^ it is assumed to be relative to the repository root. To already check the option Troca a cópia de trabalho para o novo ramo/etiqueta you can pass the /switchaftercopy switch. To check the option Criar pastas intermédias pass the /makeparents switch. You can also specify the /logmsg switch to pass a predefined log message to the branch/tag dialog. Or, if you don't want to pass the log message on the command line, use /logmsgfile:path, where caminho points to a file containing the log message.
:settings	Abre a caixa de diálogo preferências.
:remove	Remove do controlo de versões o(s) ficheiro(s) em /path.
:rename	Altera o nome do ficheiro em /path. 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 /noquestion.
:diff	Starts the external diff program specified in the TortoiseSVN settings. The /path specifies the first file. If the option /path2 is set, then the diff program is started with those two files. If /path2 is omitted, then the diff is done between the file in /path and its BASE. If the specified file also has property modifications, the external diff tool is also started for each modified property. To prevent that, pass the option /ignoreprops. To explicitly set the revision numbers use /

Comando	Descrição
	startrev:xxx and /endrev:xxx, and for the optional peg revision use /Pegrevision: xxx. If /blame is set and /path2 is not set, then the diff is done by first blaming the files with the given revisions. The parameter /line:xxx specifies the line to jump to when the diff is shown.
:shelve	Shelves the specified paths in a new shelf. The option /shelfname:name specifies the name of the shelf. An optional log message can be specified with /logmsg:message. If option /checkpoint is passed, the modifications of the files are kept.
:unshelve	Applies the shelf with the name /shelfname:name to the working copy path. By default the last version of the shelf is applied, but you can specify a version with /version:X.
:showcompare	<p>Dependendo dos URLs e revisões a comparar, isto ou mostra uma comparação unificada (se a opção unified 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.</p> <p>As opções url1, url2, revision1 e revision2 devem ser especificadas. As opções pegrevision, ignoreancestry, blame and unified são opcionais.</p> <p>If the specified url also has property modifications, the external diff tool is also started for each modified property. To prevent that, pass the option /ignoreprops.</p> <p>If a unified diff is requested, an optional prettyprint option can be specified which will show the merge-info properties in a more user readable format.</p>
:conflicteditor	Arranca o editor de conflitos especificado nas preferências do TortoiseSVN com os ficheiros correctos para o ficheiro em conflito no /path.
:relocate	Abre a caixa de diálogo reposicionar. O /path especifica o caminho da cópia de trabalho a reposicionar.
:help	Abre o ficheiro de ajuda.
:repostatus	Abre a caixa de diálogo verificar-alterações. O /path especifica a pasta da cópia de trabalho. Se for especificada a /remote, a caixa de diálogo contacta imediatamente o repositório no arranque, como se o utilizador clicasse no botão Verificar repositório.
:repobrowser	<p>Arranca a caixa de diálogo do navegador de repositório, apontando para o URL da cópia de trabalho dada pelo /path ou o /path aponta directamente para um URL.</p> <p>Uma opção adicional /rev:xxx pode ser usada para especificar a revisão que será mostrada no navegador de repositório. Se a opção /rev:xxx for omitida, será usada a HEAD por defeito.</p> <p>Se /path apontar para um URL, a /projectpropertiespath:path/to/wc especifica o caminho de onde ler e usar as propriedades de projecto.</p> <p>Se for especificada a /outfile:path o\file, o URL seleccionado e o número de revisão são escritos para esse ficheiro, quando o navegador de repositório for fechado. A primeira linha nesse ficheiro de texto contém o URL, a segunda linha a revisão no formato de texto.</p>
:ignore	Adiciona todos os objectos em /path à lista de ignorados, i.e. adiciona a propriedade svn:ignore a esses ficheiros.
:blame	Abre a caixa de diálogo responsabilizar paa o ficheiro especificado em /path.

Comando	Descrição
	<p>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.</p> <p>Se a opção <code>/line:nnn</code> for usada, o TortoiseBlame irá abrir mostrando a linha cujo número foi especificado.</p> <p>As opções <code>/ignoreeol</code>, <code>/ignorespaces</code> e <code>/ignoreallspaces</code> são também suportadas.</p>
<code>:cat</code>	Guarda um ficheiro de um URL ou cópia de trabalho dada em <code>/path</code> para a localização especificada em <code>/savepath:path</code> . A revisão é dada em <code>/revision:xxx</code> . Isto pode ser usado para obter um ficheiro com uma revisão específica.
<code>:createpatch</code>	Creates a patch file for the path given in <code>/path</code> . To skip the file Save-As dialog you can pass <code>/savepath:path</code> to specify the path where to save the patch file to directly. To prevent the unified diff viewer from being started showing the patch file, pass <code>/noview</code> . If a unified diff is requested, an optional <code>prettyprint</code> option can be specified which will show the merge-info properties in a more user readable format.
<code>:revisiongraph</code>	<p>Mostra o gráfico de revisões para o caminho dado em <code>/path</code>.</p> <p>Para criar um ficheiro de imagem do gráfico de revisão para um caminho específico, mas sem mostrar a janela do gráfico, <code>/output:path</code> com o caminho para o ficheiro de saída. O ficheiro de saída deve ter uma extensão que o gráfico de revisão possa exportar. São suportados os formatos: <code>.svg</code>, <code>.wmf</code>, <code>.png</code>, <code>.jpg</code>, <code>.bmp</code> and <code>.gif</code>.</p> <p>Visto o gráfico de revisões ter muitas opções que afectam o modo como é visualizado, podes também activar opções para usares ao criar o ficheiro de imagem de saída. Passa essas opções com <code>/options:XXXX</code>, onde <code>XXXX</code> é um valor decimal. A melhor forma para encontrares as opções requeridas é arrancar o gráfico de revisões do modo usual, colocar todas as opções do interface gráfico e fechar o gráfico. Agora as opções que necessitas de passar na linha de comandos podem ser lidas a partir do registo <code>HKCU\Software\TortoiseSVN\RevisionGraphOptions</code>.</p>
<code>:lock</code>	Bloqueia um ficheiro ou todos os ficheiros numa pasta dada por <code>/path</code> . É mostrada a caixa de diálogo 'Bloquear' para que o utilizador possa introduzir um comentário para o bloqueio.
<code>:unlock</code>	Desbloqueia um ficheiro ou todos os ficheiros numa pasta dada por <code>/path</code> .
<code>:rebuildiconcache</code>	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 <code>/noquestion</code> .
<code>:properties</code>	<p>Mostra a caixa de diálogo propriedades para o caminho especificado por <code>/path</code>.</p> <p>Para este comando lidar com propriedades versionadas é necessário uma cópia de trabalho.</p> <p>As propriedades da revisão podem ser vistas/alteradas se <code>/path</code> for um URL e for especificado <code>/rev:XXX</code>.</p> <p>Para abrir directamente a caixa de diálogo de propriedades para uma propriedade específica, passa o seu nome como <code>/property:name</code>.</p>

Comando	Descrição
:sync	<p>Exports/imports settings, either depending on whether the current settings or the exported settings are newer, or as specified.</p> <p>If a path is passed with /path, then the path is used to store or read the settings from.</p> <p>The parameter /askforpath will show a file open/save dialog for the user to chose the export/import path.</p> <p>If neither /load nor /save is specified, then TortoiseSVN determines whether to export or import the settings by looking at which ones are more recent. If the export file is more recent than the current settings, then the settings are loaded from the file. If the current settings are more recent, then the settings are exported to the settings file.</p> <p>If /load is specified, the settings are imported from the settings file.</p> <p>If /save is specified, the current settings are exported to the settings file.</p> <p>The parameter /local forces a settings export to include local settings, i.e. settings that refer to local paths.</p>

### 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. manípulo URL do Tsvncmd

Usando URLs especiais, é também possível chamar o TortoiseProc a partir de uma página web.

O TortoiseSVN regista um novo protocolo `tsvncmd`: que pode ser usado para criar hiperligações que executam comandos TortoiseSVN. Esses comandos e parâmetros são os mesmos que quando automatizas o TortoiseSVN a partir da linha de comandos.

O formato do URL `tsvncmd`: é o seguinte:

```
tsvncmd:command:cmd?parameter:paramvalue?parameter:paramvalue
```

com `cmd` a ser um dos comandos permitidos, `parameter` a ser o nome de um parâmetro como `path` ou `revision`, e `paramvalue` a ser o valor a usar para esse parâmetro. A lista de parâmetros permitidos depende do comando usado.

Os seguintes comandos são permitidos com os URLs `tsvncmd`:

- :update
- :commit
- :diff



- :repobrowser
- :checkout
- :export
- :blame
- :repostatus
- :revisiongraph
- :showcompare
- :log

Um simples URL de exemplo poderá se assemelhar a isto:

```
<a href="tsvncmd:command:update?path:c:\svn_wc?rev:1234">Update</a>
```

, ou num caso mais complexo:

```
<a href="tsvncmd:command:showcompare?
url1:https://svn.code.sf.net/p/stefanstools/code/trunk/StExBar/src/setup/Setup.wxs?
url2:https://svn.code.sf.net/p/stefanstools/code/trunk/StExBar/src/setup/Setup.wxs?
revision1:188?revision2:189">compare</a>
```

### D.3. 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.
: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
```

## D.4. TortoiseUDiff Commands

The unified diff viewer has only two command line options:

Opção	Descrição
:patchfile	Path to the unified diff file.
:p	Activates pipe mode. The unified diff is read from the console input.

### Tabela D.3. Lista de opções disponíveis

Examples (which should be entered on one line):

```
TortoiseUDiff.exe /patchfile:"c:\diff.patch"
```

If you create the diff from another command, you can use TortoiseUDiff to show that diff directly:

```
svn diff | TortoiseUDiff.exe /u
```

this also works if you omit the /p parameter:

```
svn diff | TortoiseUDiff.exe
```

---

# 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

In the descriptions which follow, the URL for a repository location is shown simply as URL, and an example might be `https://svn.code.sf.net/p/tortoisesvn/code/trunk/`. The working copy path is shown simply as PATH, and an example might be `C:\TortoiseSVN\trunk`.



## 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 [-depth ARG] [--ignore-externals] [-r rev] URL PATH
```

Os itens de nível da caixa combo estão relacionados com o parâmetro `-depth`.

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] [-depth ARG] [--ignore-externals] PATH
```

Os itens de nível da caixa combo estão relacionados com o parâmetro `-depth`.

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 "LogMessage" [-depth ARG] [--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.

What TortoiseSVN does is an

```
svn info URL_of_WC  
svn log -v URL
```

where URL is the repository *root* and then analyzes the data returned.

## E.2.9. 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 cleanup 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 propset 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 correção no formato comparação-unificada ao comparar a cópia de trabalho com a sua versão BASE.

### E.2.29. Aplicar Correção

Aplicar correções é um trabalho complicado, a não ser que a correção 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 `conflict`, 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 `modified`, 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 `removed`, 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 que um ficheiro tem fixada a propriedade `svn:needs-lock`.



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.

If an item has Subversion status none (the item is not within a working copy) then no overlay is shown. If you have chosen to disable the *Ignored* and *Unversioned* overlays then no overlay will be shown for those files either.

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 esses ícones são usados no teu sistema. Isto é porque o número de sobreposições permitida pelo Windows está limitada a 15. O Windows usa 4, e os restantes 11 podem ser usados por outras aplicações. Se não existir um número de slots de sobreposições disponíveis, o TortoiseSVN tenta ser um *Bom Cidadão (TM)* e limita o seu uso de sobreposições, para dar oportunidade a outras aplicações.

Visto estarem disponíveis clientes Tortoise para outros sistemas de controlo de versões, criámos um componente partilhado responsável pela visualização das sobreposições de ícones. Os pormenores técnicos não são aqui importantes, tudo o que necessitas de saber é que este componente partilhado permite a todos os clientes Tortoise usar as mesmas sobreposições, e por isso o limite das 11 vagas disponíveis não é afectado pela instalação de mais um cliente Tortoise. É claro que existe um senão; todos os clientes Tortoise usam as mesmas sobreposições de ícones, pelo que não consegues distinguir pelas sobreposições, que sistema de controlo de versões está a ser usado pela cópia de trabalho.

- *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* é se possível carregado, mas cai para *Normal* se não existem vagas suficientes.
- *Adicionado* é se possível carregado, mas cai para *Modificado* se não existem vagas suficientes.

---

# Apêndice G. Pacotes de Linguagens e Verificadores de Sintaxe

O instalador por defeito só tem suporte para o Inglês, mas após a instalação, poderás descarregar separadamente os pacotes de linguagens e dicionários para correção de sintaxe.

## G.1. Pacotes de Linguas

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

Cada pacote de linguagem está empacotado como um instalador com extensão .exe. Corre apenas o ficheiro de instalação e segue então as instruções. Da próxima vez que reiniciares o computador a tradução estará já disponível.

The documentation has also been translated into several different languages. You can download translated manuals from the [support page](https://tortoisesvn.net/support.html) [https://tortoisesvn.net/support.html] on our website.

## G.2. Corretor Ortográfico

TortoiseSVN uses the Windows spell checker if it's available (Windows 8 or later). Which means that if you want the spell checker to work in a different language than the default OS language, you have to install the spell checker module in the Windows settings (Settings > Time & Language > Region & Language).

TortoiseSVN will use that spell checker if properly configured with the `tsvn:projectlanguage` project property.

In case the Windows spell checker is not available, TortoiseSVN can also use spell checker dictionaries from [OpenOffice](https://openoffice.org) [https://openoffice.org] and [Mozilla](https://mozilla.org) [https://mozilla.org].

O instalador instala automaticamente os dicionários de Inglês UK e US. Se pretendes 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 de utilizador local do TortoiseSVN. Após o fim da instalação, estará também disponível o dicionário.

Ou poderás instalar manualmente os dicionários. 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/Dictionaries>.

Once you have got the dictionary files, you probably need to rename them so that the filenames only have the locale chars in it. Example:

- en\_US.aff
- en\_US.dic

Then just copy them into the `%APPDATA%\TortoiseSVN\dic` folder. If that folder isn't there, you have to create it first. TortoiseSVN will also search the Languages sub-folder of the TortoiseSVN installation folder (normally this will be `C:\Program Files\TortoiseSVN\Languages`); this is the place where the language packs put their files. However, the `%APPDATA%`-folder doesn't require administrator privileges and, thus, has higher priority. The next time you start TortoiseSVN, the spell checker will be available.

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 4.18, "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 exacta do Windows não funcionar, experimenta a língua “Base”, i.e. de\_CH (Alemão-Suíço) retornará para de\_DE (Alemão).
4. Se nenhuma das abordagens anteriores funcionou, então a língua por defeito será o Inglês, que está incluído na instalação standard.

---

# 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.
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	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.  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.

---

Limpar		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.
Propriedade		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.
Propriedade (revprop)	de Revisão	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.
Ramo		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”.
Registo		Mostra o histórico de revisões de um ficheiro ou pasta. Também conhecido como “Histórico”.
Remendo		Se uma cópia de trabalho tem alterações, só em ficheiros de texto, é possível usar o comando comparar 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.
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		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.  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

---

	<p>movido. Em qualquer outra circunstância, o que provavelmente necessitarás, é do comando “Trocar”.</p>
Repositório	<p>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.</p>
Resolver	<p>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.</p>
Responsabilizar	<p>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.</p>
Reverter	<p>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.</p>
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	<p>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.</p>
Revisão HEAD	<p>A última revisão de um ficheiro ou pasta no <i>repositório</i>.</p>
Submeter	<p>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.</p>
SVN	<p>Uma abreviatura para Subversion, frequentemente utilizada.</p> <p>O nome do protocolo personalizado do Subversion, usado pelo servidor de repositório “svnserve”.</p>
Trocar	<p>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.</p>



---

# Índice Remissivo

## Símbolos

"plugin", 196  
"Shell" do Windows, xi  
'cópia de trabalho' não versionada, 136

## A

a bloquear, 120  
acções lado-do-servidor, 128  
Aceder, 19  
actualizar, 41, 202  
adicionar, 80  
adicionar ficheiros ao repositório, 29  
alterações, 204  
anotar, 125  
arrastar, 26  
arrastar com o botão direito, 26  
arrastar-e-largar, 26  
atalho, 205  
autenticação, 27  
auto-props, 92  
automação, 211, 217, 218, 219

## B

bugtracker, 138

## C

cache de autenticação, 27  
cache de registo, 168  
caminhos UNC, 19  
caracteres de substituição, 82  
checkout disperso, 31  
checkout parcial, 31  
CLI, 220  
cliente de linha de comando, 220  
clique à direita, 24  
COM, 188, 196  
COM SubWCRRev interface, 193  
commit monitor, 185  
comparar, 52, 75  
comparar ficheiros, 204  
comparar imagens, 78  
comparar pastas, 204  
comparar revisões, 76  
conflito, 11, 43  
conflito de árvore, 43  
controlador de assuntos, 138, 196  
controlador de domínio, 207  
controlador de problemas, 138  
controlo de problema, 138  
Controlo de versões, xi  
cópia, 107, 128  
cópia de segurança, 21  
cópia de trabalho, 12  
copiar ficheiros, 80

correção, 123  
corretor ortográfico, 229  
corrpondência de padrões, 82  
Criar, 18  
    TortoiseSVN, 18  
criar repositório, 18  
criar uma cópia de trabalho, 31  
culpa, 125

## D

dar entrada, 34  
desfazer, 86  
desfazer alteração, 203  
desfazer submissão, 203  
desligar do repositório, 205  
desversionar, 137, 205  
dicionário, 229  
dif, 75, 123  
diff unificado, 123

## E

editar registo/autor, 68  
elogiar, 125  
entradas do menu de contexto, 208  
enviar alterações, 34  
estado, 47, 50  
estado da cópia de trabalho, 47  
estatísticas, 70  
etiqueta, 80, 107  
expande as palavras-chave, 90  
explorador, xi  
exportar, 136  
exportar alterações, 76  
externos, 104, 204  
extracção de versão, 188

## F

FAQ, 201  
ferramentas de comparação, 79  
ferramentas de integração, 79  
ficheiros especiais, 31  
ficheiros temporários, 29  
ficheiros/pastas não-versionadas, 82  
filtro, 69  
funções desactivadas, 208

## G

ganchos, 21  
ganchos de cliente, 171  
GPO, 207  
grafo, 131  
grafo de revisões, 131

## H

hiperligação, 22  
hiperligação de checkout, 22  
hiperligação TortoiseSVN, 22

história, 57

## I

IBugtraqProvider, 196

ícones, 47

ignorados globais, 146

ignorar, 82

Importar, 29

importar no local, 30

instalar, 1

Instalar, 207

integrar, 111

duas árvores, 115

faixa de revisão, 112

integrar conflitos, 118

## L

limpar, 86

limpeza, 88

linha de comando, 211, 218, 219

lista de alterações, 53

Livro sobre o Subversion, 7

## M

manípulo URL, 217

marcar versão, 107

maximizar, 28

mensagem de registo, 202

mensagem de submissão, 202

mensagem vazia, 202

mensagens de registo, 57

mensagens de submissão, 57

menu de contexto, 24

Microsoft Word, 79

modificações, 50

monitoring projects, 185

Monitorizar a integração, 117

mover, 85, 202

mover ficheiros, 80

msi, 207

## N

navegador-de-repositório, 128

números de versão nos ficheiros, 188

## O

obter alterações, 41

## P

pacotes de linguas, 229

padrão de exclusão, 146

palavras-chave, 90

Partilha de rede, 19

políticas de grupo, 207, 208

preferências, 144

prioridade de sobreposição, 227

project monitor, 185

projectos ASP, 208

projectos comuns, 204

projectos proprietários, 204

propriedades de janelas, 49

propriedades de projecto, 93

propriedades de revisão, 68

Propriedades do TortoiseSVN, 93

Propriedades Subversion, 89

## R

ramo, 80, 107

readonly, 120

rebobinar, 203

registo, 57, 178

registo de controlo de integração, 67

remote commits, 185

remove, 84, 84

remove versionamento, 205

renomear, 85, 128, 202

renomear ficheiros, 80

reorganizar, 202

reposicionar, 137

repositório, 7, 29

repositórios externos, 104

resolver, 43

reverter, 86, 203

revisão, 14, 131

revprops, 68

## S

scripts de gancho, 21, 171

scripts de gancho do lado do servidor, 21

servidor movido, 137, 137

servidor proxy, 160

shelve, 55

sítio de internet, 22

sobreposições, 47, 227

sons, 144

submiter, 34

SubWCRev, 188

SVN exportar, 31

SVN\_ASP\_DOT\_NET\_HACK, 208

## T

TortoiseIDiff, 78

traduções, 229

trocar, 110

## U

unidades SUBST, 158

unshelve, 55

URL alterado, 137

URL do repositório alterado, 137

## V

ver alterações, 47

verificar actualizações, 207

verificar nova versão, 207  
versão, 207  
versionar ficheiros novos, 80  
ViewVC, 143  
visualizador de repositório, 143  
visualizador de servidor, 128  
visualizador web, 143  
VS2003, 208

## **W**

WebSVN, 143