

# **TortoiseSVN**

**Un cliente de Subversion para Windows**

**Version 1.12**

**Stefan Küng  
Lübbe Onken  
Simon Large**

---

# **TortoiseSVN: Un cliente de Subversion para Windows: Version 1.12**

por Stefan Küng, Lübbe Onken, y Simon Large

Traducción: Fernando P. Najera Cano (yo@FernandoNajera.com), Juan Carlos Fuentes Michel (jcfuentes@gmail.com), Tomás Christie (tommychristie@gmail.com), Ariel Abaca (abacrotto@hotmail.com)

fecha de publicación 2019/03/11 21:25:21 (r28524)

---

---

# Tabla de contenidos

Prefacio .....	xi
1. ¿Qué es TortoiseSVN? .....	xi
2. Características de TortoiseSVN .....	xi
3. Licencia .....	xii
4. Desarrollo .....	xii
4.1. Historia de TortoiseSVN .....	xii
4.2. Reconocimientos .....	xiii
5. Guía de Lectura .....	xiii
6. Terminología usada en este documento .....	xiv
1. Empezando .....	1
1.1. Instalando TortoiseSVN .....	1
1.1.1. Requerimientos del sistema .....	1
1.1.2. Instalación .....	1
1.2. Conceptos básicos de control de versiones .....	1
1.3. Prueba del producto .....	2
1.3.1. Crear un Repositorio .....	2
1.3.2. Importar un Proyecto .....	2
1.3.3. Obteniendo una Copia de Trabajo .....	3
1.3.4. Haciendo Cambios .....	4
1.3.5. Añadiendo Más Archivos .....	4
1.3.6. Viendo la Historia del Proyecto .....	4
1.3.7. Deshaciendo Cambios .....	5
1.4. Siguiendo adelante... .....	5
2. Conceptos básicos de control de versiones .....	7
2.1. El repositorio .....	7
2.2. Modelos de versionado .....	7
2.2.1. El problema de compartir archivos .....	8
2.2.2. La solución bloquear-modificar-desbloquear .....	8
2.2.3. La solución copiar-modificar-fusionar .....	9
2.2.4. ¿Qué hace Subversion? .....	11
2.3. Subversion en acción .....	11
2.3.1. Copias de trabajo .....	11
2.3.2. URLs de repositorio .....	12
2.3.3. Revisiones .....	13
2.3.4. Cómo reconocen las copias de trabajo al repositorio .....	15
2.4. Sumario .....	15
3. El repositorio .....	16
3.1. Creación de repositorios .....	16
3.1.1. Creando un repositorio con el cliente de línea de comandos .....	16
3.1.2. Creando el repositorio con TortoiseSVN .....	16
3.1.3. Acceso local al repositorio .....	17
3.1.4. Accediendo a un repositorio en una unidad de red .....	17
3.1.5. Organización del repositorio .....	17
3.2. Copia de seguridad del Repositorio .....	19
3.3. Scripts gancho en el lado del servidor .....	19
3.4. Enlaces de obtener .....	20
3.5. Accediendo al repositorio .....	20
4. Guía de uso diario .....	22
4.1. Características generales .....	22
4.1.1. Íconos superpuestos .....	22
4.1.2. Menús contextuales .....	22
4.1.3. Arrastrar y soltar .....	24
4.1.4. Atajos comunes .....	25
4.1.5. Autenticación .....	25
4.1.6. Maximizando ventanas .....	26

---

4.2. Importando datos en un repositorio .....	26
4.2.1. Importar .....	26
4.2.2. Importar en el sitio .....	27
4.2.3. Archivos especiales .....	28
4.3. Obteniendo una copia de trabajo .....	28
4.3.1. Profundidad de obtención .....	29
4.4. Confirmando sus cambios en el repositorio .....	31
4.4.1. El diálogo de Confirmación .....	31
4.4.2. Listas de cambios .....	34
4.4.3. Confirmar ficheros parcialmente .....	34
4.4.4. Excluyendo ítems de la lista de confirmación .....	35
4.4.5. Mensajes de registro de confirmación .....	35
4.4.6. Progreso de confirmación .....	37
4.5. Actualice su copia de trabajo con los cambios de otros .....	38
4.6. Resolviendo conflictos .....	40
4.6.1. Conflictos de archivos .....	40
4.6.2. Conflictos de propiedad .....	41
4.6.3. Conflictos de árbol .....	41
4.7. Obteniendo información del estado .....	44
4.7.1. Íconos superpuestos .....	44
4.7.2. Estado detallado .....	45
4.7.3. Estado local y remoto .....	46
4.7.4. Viendo diferencias .....	49
4.8. Listas de cambios .....	49
4.9. Shelving .....	51
4.10. Diálogo de Registro de revisiones .....	53
4.10.1. Invocando el diálogo de Registro de revisiones .....	54
4.10.2. Acciones del registro de revisiones .....	54
4.10.3. Obteniendo información adicional .....	55
4.10.4. Obteniendo más mensajes de registro .....	61
4.10.5. Revisión actual de la copia de trabajo .....	62
4.10.6. Características de registro de fusión .....	62
4.10.7. Cambiando el mensaje de registro y el autor .....	63
4.10.8. Filtrando los mensajes de registro .....	63
4.10.9. Información estadística .....	65
4.10.10. Modo sin conexión .....	68
4.10.11. Refrescando la vista .....	68
4.11. Viendo diferencias .....	68
4.11.1. Diferencias de archivos .....	69
4.11.2. Opciones de fin de línea y espacios en blanco .....	70
4.11.3. Comparando carpetas .....	70
4.11.4. Diferenciando imágenes utilizando TortoiseIDiff .....	72
4.11.5. Diferenciando los documentos de Office .....	73
4.11.6. Herramientas externas de diferencias/fusión .....	73
4.12. Añadiendo nuevos archivos y directorios .....	74
4.13. Copiando/Moviendo/Renombrando archivos y carpetas .....	74
4.14. Ignorando archivos y directorios .....	75
4.14.1. Concordancia de patrones en las listas de ignorados .....	76
4.15. Eliminando, moviendo y renombrando .....	77
4.15.1. Eliminando archivos y carpetas .....	78
4.15.2. Moviendo archivos y carpetas .....	79
4.15.3. Lidiando con conflictos en las mayúsculas/minúsculas de un nombre de archivo .....	80
4.15.4. Reparando renombrados de archivos .....	80
4.15.5. Eliminando archivos no versionados .....	80
4.16. Deshacer cambios .....	80
4.17. Limpieza .....	82
4.18. Configuración del proyecto .....	83
4.18.1. Propiedades de Subversion .....	83

---

4.18.2. Propiedades de proyecto TortoiseSVN .....	86
4.18.3. Editores de Propiedades .....	92
4.19. Ítems externos .....	99
4.19.1. Carpetas externas .....	99
4.19.2. Archivos externos .....	101
4.19.3. Creando exteriores vía arrastrar y soltar .....	101
4.20. Haciendo ramas / etiquetas .....	102
4.20.1. Crando una rama o etiqueta .....	102
4.20.2. Otras formas de crear una rama o etiqueta. ....	105
4.20.3. Obtener o cambiar... .....	105
4.21. Fusionando .....	106
4.21.1. Fusionando un rango de revisiones .....	107
4.21.2. Fusionando dos árboles diferentes .....	109
4.21.3. Opciones de fusión .....	110
4.21.4. Revisando los resultados de la fusión .....	110
4.21.5. Registro de fusión .....	111
4.21.6. Handling Conflicts after Merge .....	112
4.21.7. Mantenimiento de ramas de características .....	113
4.22. Bloqueando .....	114
4.22.1. Cómo trabaja el bloqueo en Subversion .....	114
4.22.2. Obteniendo un bloqueo .....	115
4.22.3. Quitando un bloqueo .....	116
4.22.4. Comprobando el estado de los bloqueos .....	116
4.22.5. Configurando los archivos no-bloqueados como de sólo-lectura .....	117
4.22.6. Los scripts ganchos de bloqueo .....	117
4.23. Creando y aplicando parches .....	117
4.23.1. Creando un archivo parche .....	117
4.23.2. Aplicando un archivo parche .....	119
4.24. ¿Quién cambió qué línea? .....	119
4.24.1. Autoría de archivos .....	120
4.24.2. Autoría de las diferencias .....	122
4.25. El navegador de repositorios .....	122
4.26. Gráficos de revisión .....	125
4.26.1. Nodos del gráfico de revisión .....	126
4.26.2. Cambiando la vista .....	126
4.26.3. Usando el gráfico .....	128
4.26.4. Refrescando la vista .....	129
4.26.5. Podando árboles .....	129
4.27. Exportando una copia de trabajo de Subversion .....	129
4.27.1. Eliminando una copia de trabajo del control de versiones .....	131
4.28. Relocalizando una copia de trabajo .....	131
4.29. Integración con sistemas de control de errores / seguimiento de incidencias .....	132
4.29.1. Añadiendo números de incidencia en los mensajes de registro .....	132
4.29.2. Obteniendo información desde el gestor de incidencias .....	136
4.30. Integración con visores de repositorios basados en web .....	137
4.31. Configuración de TortoiseSVN .....	138
4.31.1. Configuración general .....	138
4.31.2. Configuración del gráfico de revisión .....	147
4.31.3. Configuración de los íconos superpuestos .....	150
4.31.4. Configuración de red .....	154
4.31.5. Configuración de programas externos .....	156
4.31.6. Datos de configuración almacenados .....	161
4.31.7. Caché de registro .....	162
4.31.8. Scripts gancho del lado del cliente .....	165
4.31.9. Configuración de TortoiseBlame .....	170
4.31.10. TortoiseUDiff Settings .....	171
4.31.11. Configuración de exportación TSVN .....	172
4.31.12. Configuración Avanzada .....	172

---

---

4.32. Último paso .....	177
5. Project Monitor .....	178
5.1. Adding projects to monitor .....	178
5.2. Monitor dialog .....	179
5.2.1. Main operations .....	179
6. El programa SubWCRev .....	181
6.1. La línea de comandos de SubWCRev .....	181
6.2. Sustitución de palabras clave .....	183
6.3. Ejemplo de palabras clave .....	184
6.4. Interfaz COM .....	186
7. Interfaz IBugtraqProvider .....	189
7.1. Convenciones de nombres .....	189
7.2. La interfaz de IBugtraqProvider .....	189
7.3. La interfaz IBugtraqProvider2 .....	191
A. Preguntas más frecuentes (FAQ) .....	194
B. ¿Cómo...? .....	195
B.1. Mover/copiar muchos archivos de golpe .....	195
B.2. Obligar a los usuarios a introducir un mensaje de registro .....	195
B.2.1. Script gancho en el servidor .....	195
B.2.2. Propiedades del proyecto .....	195
B.3. Actualizar los archivos seleccionados desde el repositorio .....	195
B.4. Deshacer revisiones en el repositorio .....	196
B.4.1. Utilice el diálogo Registro de revisiones .....	196
B.4.2. Utilice el diálogo Fusionar .....	196
B.4.3. Utilice svndumpfilter .....	196
B.5. Comparar dos revisiones de un archivo o carpeta .....	197
B.6. Incluir un sub-proyecto común .....	197
B.6.1. Utilice svn:externals .....	197
B.6.2. Utilice una copia de trabajo anidada .....	197
B.6.3. Utilice una ruta relativa .....	198
B.6.4. Add the project to the repository .....	198
B.7. Crear un acceso directo a un repositorio .....	198
B.8. Ignorar archivos que ya están versionados .....	198
B.9. Desversionar una copia de trabajo .....	199
B.10. Eliminar una copia de trabajo .....	199
C. Trucos útiles para los administradores .....	200
C.1. Instalar TortoiseSVN utilizando políticas de grupo .....	200
C.2. Redirigir la comprobación de actualización .....	200
C.3. Estableciendo la variable de entorno SVN_ASP_DOT_NET_HACK .....	201
C.4. Deshabilitar entradas del menú contextual .....	201
D. Automatizando TortoiseSVN .....	204
D.1. Comandos de TortoiseSVN .....	204
D.2. Manejador de URLs Tsvncmd .....	210
D.3. Comandos de TortoiseIDiff .....	211
D.4. TortoiseUDiff Commands .....	211
E. Referencia cruzada del interface de línea de comandos .....	213
E.1. Convenciones y reglas básicas .....	213
E.2. Comandos de TortoiseSVN .....	213
E.2.1. Obtener .....	213
E.2.2. Actualizar .....	213
E.2.3. Actualizar a la revisión .....	214
E.2.4. Confirmar .....	214
E.2.5. Diff .....	214
E.2.6. Mostrar registro .....	215
E.2.7. Comprobar modificaciones .....	215
E.2.8. Gráfico de revisión .....	215
E.2.9. Navegador de repositorios .....	215
E.2.10. Editar conflictos .....	216

---

---

E.2.11. Resuelto .....	216
E.2.12. Renombrar .....	216
E.2.13. Eliminar .....	216
E.2.14. Revertir .....	216
E.2.15. Limpieza .....	216
E.2.16. Obtener bloqueo .....	216
E.2.17. Quitar bloqueo .....	217
E.2.18. Rama/Etiqueta .....	217
E.2.19. Cambiar .....	217
E.2.20. Fusionar .....	217
E.2.21. Exportar .....	218
E.2.22. Relocalizar .....	218
E.2.23. Crear repositorio aquí .....	218
E.2.24. Añadir .....	218
E.2.25. Importar .....	218
E.2.26. Autoría .....	218
E.2.27. Añadir a la lista de ignorados .....	218
E.2.28. Crear parche .....	219
E.2.29. Aplicar parche .....	219
F. Detalles de implementación .....	220
F.1. Íconos superpuestos .....	220
G. Paquetes de Idiomas y Correctores Ortográficos .....	222
G.1. Paquetes de idiomas .....	222
G.2. Corrector ortográfico .....	222
Glosario .....	224
Índice .....	227

---

## Lista de figuras

1.1. El menú de TortoiseSVN para carpetas no versionadas .....	2
1.2. El diálogo Importar .....	3
1.3. Examinador de Diferencias de Archivos .....	4
1.4. El Diálogo de Registro .....	5
2.1. Un sistema típico cliente/servidor .....	7
2.2. El problema a evitar .....	8
2.3. La solución bloquear-modificar-desbloquear .....	9
2.4. La solución copiar-modificar-fusionar .....	10
2.5. ... continuación de Copiar-modificar-fusionar .....	10
2.6. El sistema de archivos del repositorio .....	12
2.7. El repositorio .....	14
3.1. El menú de TortoiseSVN para carpetas no versionadas .....	16
4.1. Explorador mostrando íconos superpuestos .....	22
4.2. Menú contextual para un directorio bajo el control de versiones .....	23
4.3. Menú archivo del explorador para un acceso directo en una carpeta versionada .....	24
4.4. Menú de arrastre con el botón derecho para un directorio bajo el control de versiones .....	24
4.5. Diálogo de autenticación .....	25
4.6. El diálogo Importar .....	27
4.7. El diálogo Obtener .....	29
4.8. El diálogo de Confirmación .....	32
4.9. El corrector ortográfico del diálogo de Confirmación .....	36
4.10. El diálogo Progreso mostrando el progreso de una confirmación .....	37
4.11. Diálogo de progreso mostrando una actualización terminada .....	38
4.12. Explorador mostrando íconos superpuestos .....	44
4.13. Página de propiedades del Explorador, pestaña Subversion .....	46
4.14. Comprobar modificaciones .....	47
4.15. Diálogo de confirmación con listas de cambios .....	50
4.16. Diálogo de archivado .....	52
4.17. Unshelve dialog .....	53
4.18. El diálogo de Registro de revisiones .....	54
4.19. El panel superior del diálogo de Registro de revisiones con el menú contextual .....	55
4.20. The Code Collaborator Settings Dialog .....	58
4.21. Menú contextual del panel superior para 2 revisiones seleccionadas .....	58
4.22. El panel inferior del diálogo de Registro con el menú contextual .....	59
4.23. The Log Dialog Bottom Pane with Context Menu When Multiple Files Selected. ....	60
4.24. El diálogo de registro mostrando revisiones con registro de fusión .....	62
4.25. Histograma de confirmaciones por autor .....	65
4.26. Gráfico de tarta de confirmaciones por autor .....	66
4.27. Gráfico de confirmaciones por fecha .....	67
4.28. Diálogo para terminar la conexión .....	68
4.29. El diálogo Comparar Revisiones .....	71
4.30. El visor de diferencias de imágenes .....	72
4.31. Menú contextual del explorador para archivos no versionados .....	74
4.32. Menú de arrastre con el botón derecho para un directorio bajo el control de versiones .....	75
4.33. Menú contextual del explorador para archivos no versionados .....	76
4.34. Menú contextual del explorador para archivos versionados .....	78
4.35. Diálogo de Revertir .....	81
4.36. The Cleanup dialog .....	82
4.37. Página de propiedades de Subversion .....	83
4.38. Añadiendo propiedades .....	84
4.39. Property dialog for hook scripts .....	89
4.40. Property dialog boolean user types .....	89
4.41. Cuadro de diálogo de tipos de estados de usuario. ....	90
4.42. Diálogo de propiedades de tipos de usuario de una línea. ....	91
4.43. Property dialog multi-line user types .....	91



---

4.44. svn:Página de propiedades de externos .....	93
4.45. svn: Página de propiedades de palabras clave .....	93
4.46. svn:Página de propiedades de estilo EOL .....	94
4.47. tsvn: Página de propiedades de bugtraq .....	95
4.48. Tamaño de la página de propiedades del registro de mensajes .....	96
4.49. Página de propiedades de idioma. ....	96
4.50. svn:página de propiedades tipo-mime .....	97
4.51. svn:página de propiedades necesita-bloquea .....	97
4.52. svn:página de propiedades de ejecutables .....	97
4.53. Property dialog merge log message templates .....	98
4.54. El diálogo Rama/Etiqueta .....	103
4.55. El diálogo Cambiar .....	106
4.56. El asistente de fusionado - Seleccionar el rango de revisiones .....	107
4.57. El asistente de fusión - Fusión de árboles .....	109
4.58. The Merge Conflict Dialog .....	112
4.59. The Merge Tree Conflict Dialog .....	113
4.60. The Merge-All Dialog .....	113
4.61. El diálogo Bloquear .....	115
4.62. El diálogo Comprobar modificaciones .....	116
4.63. El diálogo Crear parche .....	118
4.64. El diálogo Anotar / Autoría .....	120
4.65. TortoiseBlame .....	121
4.66. El navegador de repositorios .....	123
4.67. Un gráfico de revisiones .....	125
4.68. El diálogo Exportar-desde-URL .....	130
4.69. El diálogo Relocalizar .....	131
4.70. El diálogo de propiedades de Bugtraq .....	133
4.71. Diálogo de ejemplo de la interacción con el gestor de incidencias .....	137
4.72. El diálogo Configuración, página General .....	139
4.73. El diálogo Configuración, página de Menú contextual .....	141
4.74. El diálogo Configuración, página de Diálogos 1 .....	142
4.75. El diálogo Configuración, página de Diálogos 2 .....	144
4.76. The Settings Dialog, Dialogs 3 Page .....	145
4.77. El diálogo Configuración, página de Colores .....	146
4.78. El diálogo Configuración, página Gráfico de revisión .....	147
4.79. El diálogo Configuración, página Colores del gráfico de revisión .....	148
4.80. El diálogo Configuración, página de íconos superpuestos .....	150
4.81. El diálogo Configuración, página de Conjunto de íconos .....	153
4.82. El Diálogo de Configuración, Página de Manejadores de Íconos .....	154
4.83. El diálogo Configuración, página de Red .....	155
4.84. El diálogo Configuración, página de Visor de diferencias .....	156
4.85. El diálogo Configuración, diálogo de Diferencias/Fusión avanzadas .....	160
4.86. El diálogo Configuración, página de Datos almacenados .....	161
4.87. El diálogo Configuración, página Caché de registro .....	162
4.88. El diálogo Configuración, página Estadísticas de la caché de registro .....	164
4.89. El diálogo Configuración, página de scripts gancho .....	165
4.90. El diálogo Configuración, configurar scripts gancho .....	166
4.91. El diálogo Configuración, página Integración con control de incidencias .....	169
4.92. El diálogo Configuración, página TortoiseBlame .....	170
4.93. The Settings Dialog, TortoiseUDiff Page .....	171
4.94. The Settings Dialog, Sync Page .....	172
4.95. Barra de tareas con agrupamiento por defecto .....	174
4.96. Barra de tareas con agrupamiento por repositorio .....	174
4.97. Barra de tareas con agrupamiento por repositorio .....	175
4.98. Agrupamiento de barra de tareas con superposición de colores de repositorio .....	175
5.1. The edit project dialog of the project monitor .....	178
5.2. The main dialog of the project monitor .....	179
C.1. El diálogo de confirmación, mostrando la notificación de actualización .....	200

---

## Lista de tablas

2.1. URLs de acceso al repositorio .....	13
4.1. Pinned Revision .....	104
6.1. Lista de opciones de línea de comandos disponible .....	182
6.2. List of SubWCRev error codes .....	182
6.3. List of available keywords .....	183
6.4. Métodos de automatización/COM soportados .....	186
C.1. Entradas de menú y sus valores .....	201
D.1. Lista de comandos y opciones disponibles .....	205
D.2. Lista de las opciones disponibles .....	211
D.3. Lista de las opciones disponibles .....	212

---

# Prefacio



TortoiseSVN

El control de versiones es el arte de manejar cambios en la información. Ha sido desde siempre una herramienta crítica para los programadores, quienes típicamente emplean su tiempo haciendo pequeños cambios al software y luego deshaciendo o comprobando esos cambios al día siguiente. Imagine un equipo de estos programadores trabajando concurrentemente - ¡y quizás también simultáneamente en los mismos archivos! - y podrá ver por qué se necesita un buen sistema para *manejar el caos potencial*.

## 1. ¿Qué es TortoiseSVN?

TortoiseSVN es un cliente gratuito de código abierto para el sistema de control de versiones *Apache™ Subversion®*. Esto significa que TortoiseSVN administra archivos y directorios a lo largo del tiempo. Los archivos se almacenan en un *repositorio* central. El repositorio es prácticamente lo mismo que un servidor de archivos ordinario, con la excepción de que recuerda todos los cambios que se hayan hecho a sus archivos y directorios. Esto le permite al usuario recuperar versiones antiguas de sus archivos y examinar la historia de cómo y cuándo cambiaron sus datos, y quién hizo el cambio. Esta es la razón por la que mucha gente piensa en Subversion, y los sistemas de control de versiones en general, como una especie de “máquinas del tiempo”.

Algunos sistemas de control de versiones también son sistemas de manejo de configuración del software (SCM por sus iniciales en inglés). Estos sistemas están diseñados específicamente para manejar árboles de código fuente, y tienen muchas características que son específicas para el desarrollo de software - tales como el entendimiento nativo de los lenguajes de programación, o proporcionan herramientas para compilar software. Subversion, sin embargo, no es uno de estos sistemas; es un sistema general que puede ser utilizado para manejar *cualquier* colección de archivos, incluyendo código fuente.

## 2. Características de TortoiseSVN

¿Qué hace de TortoiseSVN un cliente tan bueno de Subversion? Aquí hay una pequeña lista de sus características.

### Integración con Windows

TortoiseSVN se integra perfectamente con Windows (por ejemplo, en el Explorador de archivos). Esto significa que puede seguir trabajando con las herramientas que ya conoce. Y que no tiene que cambiar a una aplicación diferente cada vez que necesite las funciones del control de versiones.

Y ni siquiera está obligado a usar el Explorador de archivos de Windows; los menús contextuales de TortoiseSVN también funcionan en otros administradores de archivos, y en el cuadro de diálogo Archivo/Abrir que es común en la mayoría de aplicaciones estándar de Windows. Sin embargo, debe tener en cuenta que TortoiseSVN está desarrollado con la intención de ser una extensión del Explorador de archivos de Windows. Por este motivo, puede que en otras aplicaciones la integración no sea tan completa, y que por ejemplo, los iconos superpuestos no se muestren.

### Iconos superpuestos

El estado de cada carpeta y archivo versionado se indica por pequeños iconos superpuestos. De esta forma, puede ver fácilmente el estado en el que se encuentra su copia de trabajo.

### Interfaz Gráfica de Usuario

Cuando usted lista los cambios realizados a un archivo o carpeta, puede hacer click en una revisión para ver los comentarios de esa confirmación. Puede también ver una lista de archivos modificados - solamente haga doble click en un archivo para ver exactamente qué ha cambiado.

El cuadro de diálogo de confirmación muestra todos los ítems que se incluirán en una confirmación, y cada ítem tiene una casilla para que pueda escoger que cambios desea incluir. En caso que haya olvidado añadir algún archivo, los archivos que no están versionados pueden ser también listados.

#### Fácil acceso a los comandos de Subversion

Todos los comandos de Subversion están disponibles desde el menú contextual del explorador. TortoiseSVN añade su propio submenú allí.

Dado que TortoiseSVN es un cliente de Subversion, también queremos enseñarle algunas de las características del propio Subversion:

#### Versionado de carpetas

CVS sólo controla la historia de archivos individuales, pero Subversion implementa un sistema “virtual” de archivos versionados que toma en cuenta los cambios en todos los árboles de directorios en el tiempo. Los archivos y los directorios están versionados. Como resultado, hay comandos reales en el lado del cliente como **mover** y **copiar** que operan en archivos y directorios.

#### Confirmaciones atómicas

Una confirmación puede entrar al repositorio completamente, o no entrar en absoluto. Esto permite a los desarrolladores generar y confirmar cambios como unidades lógicas.

#### Metadatos versionados

Cada archivo y carpeta tiene un conjunto invisible de “propiedades” adjuntos. Se puede crear y almacenar cualquier par de clave/valor que se desee. Las propiedades se versionan en el tiempo, igual que el contenido de los archivos.

#### Elección de capas de red

Subversion tiene una noción abstracta del acceso al repositorio, haciendo que se puedan implementar nuevos mecanismos de red fácilmente. El servidor de red “avanzado” de Subversion es un módulo para el servidor web Apache, que habla una variante de HTTP llamada WebDAV/DeltaV. Esto le da a Subversion una gran ventaja en estabilidad e interoperatividad, y provee varias características gratis: autenticación, autorización, compresión de la transmisión y navegación del repositorio, por ejemplo. También está disponible un proceso servidor de Subversion independiente. Este servidor habla un protocolo propio que puede encapsularse fácilmente sobre ssh.

#### Manejo de datos consistente



La superposición *Necesita bloqueo* se utiliza para indicar cuando un archivo tiene el conjunto de propiedades `svn:needs-lock`.

#### Etiquetado y creación de ramas eficiente

*Cerrado* se carga si es posible, pero vuelve a *Normal* si no hay suficientes huecos.

## 3. Licencia

TortoiseSVN es un proyecto de Código Abierto desarrollado bajo la licencia GNU General Public License (GLP). Es libre de descargar y utilizar, ya sea para uso personal o comercial, en cualquier número de PCs.

Aunque mucha gente sólo descarga el instalador, usted también tiene acceso completo de lectura al código fuente de éste programa. Usted puede examinar el código en este link <https://sourceforge.net/p/tortoisesvn/code/HEAD/tree/>. La versión de desarrollo más actual está localizada en `/trunk/`, y las versiones publicadas están localizadas bajo `/tags/`.

## 4. Desarrollo

Tanto TortoiseSVN como Subversion se desarrollan por una comunidad de gente que trabaja en estos proyectos. Proviene de diferentes países alrededor de todo el mundo, trabajando juntos para crear programas estupendos.

### 4.1. Historia de TortoiseSVN

En 2002, Tim Kemp se dio cuenta que Subversion era un sistema de control de versiones muy bueno, pero le faltaba un buen cliente GUI. La idea de tener un cliente de Subversion integrado en la consola de Windows fue

inspirada por un cliente similar para CVS llamado TortoiseCVS. Tim estudió el código fuente de TortoiseCVS y lo utilizó como base de TortoiseSVN. Entonces inició el proyecto, registró el dominio `tortoisesvn.org` y puso el código fuente en línea.

Durante ese tiempo, Stefan Küng estaba buscando un sistema de control de versiones bueno y gratuito, y encontró Subversion y el código fuente de TortoiseSVN. Como TortoiseSVN todavía no estaba listo para usarse, se unió al proyecto y empezó a programar. Pronto reescribió la mayor parte del código existente y empezó a añadir comandos y características, hasta el punto en que no quedó nada del código original.

Según se fue estabilizando Subversion, atrajo más y más usuarios que también empezaron a utilizar TortoiseSVN como su cliente de Subversion. La base de usuarios de TortoiseSVN se incrementó rápidamente (y aún lo hace cada día). Entonces Lübbe Onken se ofreció a ayudar con algunos íconos más vistosos y un logo para TortoiseSVN. Él se encarga ahora de la página web y de administrar las muchas traducciones.

Con el tiempo, otros sistemas de control de versiones tienen su propio cliente Tortoise que causaron un problema de superposiciones con el icono en el navegador: el número de tales superposiciones es limitada e incluso un cliente Tortoise puede fácilmente superar dicho límite. Fue entonces cuando Stefan Küng implementa el componente TortoiseOverlays que permite a todos los clientes Tortoise que utilizan los mismos sobreimpresionados. de icono Ahora todos los clientes, Tortoise e incluso algunos clientes que no son Tortoise usan este componente compartidos.

## 4.2. Reconocimientos

Tim Kemp

por fundar el proyecto TortoiseSVN

Stefan Küng

por el duro trabajo de llevar a TortoiseSVN a lo que es hoy, y el liderazgo del proyecto

Lübbe Onken

por los bonitos íconos, el logo, la caza de errores, por traducir y administrar traducciones

Simon Large

por mantener la documentación

Stefan Fuhrmann

para el caché de registro y el gráfico de revisiones

El libro de Subversion

por la gran introducción a Subversion y su capítulo 2 que hemos copiado aquí

El proyecto Tigris Style

por algunos de los estilos que están siendo reutilizados en esta documentación

Nuestros colaboradores

por los parches, informes de errores y nuevas ideas, y por ayudar a otros respondiendo preguntas de nuestra lista de correo

Nuestros donantes

por la cantidad de horas de entretenimiento con la música que nos enviaron

## 5. Guía de Lectura

Este libro está escrito para usuarios informáticos que quieren usar Subversion para manejar sus datos, prefieren utilizar una GUI en vez de un cliente en la línea de comandos. Dado que TortoiseSVN es una extensión del shell de Windows, se asume que el usuario está familiarizado con el Explorador de windows y sabe cómo usarlo.

Este [Prefacio](#) explica qué es TortoiseSVN, le ofrece una breve explicación sobre el proyecto TortoiseSVN y la comunidad de gente que trabaja en él, y las condiciones de licencia para utilizarlo y distribuirlo.

La sección **Capítulo 1, Empezando** explica como instalar TortoiseSVN en su PC, y como empezar a utilizarlo inmediatamente.

En el **Capítulo 2, Conceptos básicos de control de versiones** le ofrecemos una breve introducción al sistema de control de revisiones *Subversion* que es la base de TortoiseSVN. Está prestado de la documentación del proyecto Subversion y explica las diferentes formas de control de versiones y cómo funciona Subversion.

El capítulo **Capítulo 3, El repositorio** explica cómo se prepara un repositorio local, algo útil para probar Subversion y TortoiseSVN utilizando un único PC. También explica algo sobre la administración de repositorios, que también es relevante para los repositorios que se encuentran en un servidor.

El **Capítulo 4, Guía de uso diario** es la sección más importante, ya que explica las características principales de TortoiseSVN y cómo utilizarlas. Tiene la forma de un tutorial, empezando con obtener una copia de trabajo, modificarla, confirmar los cambios, etc. Luego avanza a temas más avanzados.

The **Capítulo 5, Project Monitor** explains how you can monitor your Subversion projects so you don't miss important commits from your other team members.

El **Capítulo 6, El programa SubWCRev** describe un programa adicional que se incluye con TortoiseSVN, y que puede extraer información de su copia de trabajo y escribirla en un archivo. Esto es útil para incluir información de compilación en sus proyectos.

La sección **Apéndice B, ¿Cómo...?** responde algunas preguntas frecuentes sobre la realización de tareas que no están cubiertas explícitamente en ninguna otra parte.

La sección **Apéndice D, Automatizando TortoiseSVN** muestra cómo puede invocar los diálogos GUI de TortoiseSVN desde la línea de comandos. Esto es útil para scripts donde se necesite la interacción del usuario.

El **Apéndice E, Referencia cruzada del interface de línea de comandos** muestra la correlación entre los comandos de TortoiseSVN y sus equivalentes en el cliente de línea de comandos de Subversion `svn.exe`.

## 6. Terminología usada en este documento

Para hacer más fácil la lectura de la documentación, los nombres de todas las pantallas y menús de TortoiseSVN están remarcados en un tipo de letra diferente. Por ejemplo, el **Diálogo de Registro**.

Las opciones de menú se indican con una flecha. TortoiseSVN → **Mostrar Registro** significa: seleccione *Mostrar Registro* desde el menú contextual *TortoiseSVN*.

Donde aparezca un menú contextual local dentro de uno de los diálogos de TortoiseSVN, se mostrará así: **Menú Contextual** → Grabar como ...

Los botones del interfaz de usuario se indican como este: Pulse **OK** para continuar.

Las acciones del usuario se indican en negrita. **Alt+A**: pulse la tecla **Alt** en su teclado, y mientras la mantiene pulsada, pulse también la tecla **A**. Arrastre con botón derecho: pulse el botón derecho del ratón, y mientras lo mantiene pulsado, *arrastre* los ítems a su nuevo destino.

La salida del sistema y la entrada por teclado se indica con una fuente también diferente.



### Importante

Las notas importantes están marcadas con un ícono.



### Sugerencia

Trucos que le facilitan la vida.



---

## Atención

Lugares donde debe tener cuidado con lo que hace.



---

## Aviso

Donde hay que tener un cuidado extremo. Puede ocurrir una corrupción de datos u otras cosas horribles si se ignoran estas advertencias.



---

# Capítulo 1. Empezando

Esta sección esta destinada a la gente que quiere comprender que es lo que TortoiseSVN hace y experimentar con él. Explica como instalar TortoiseSVN y como configurar un repositorio local, y muestra paso a paso las operaciones más utilizadas.

## 1.1. Instalando TortoiseSVN

### 1.1.1. Requerimientos del sistema

TortoiseSVN se ejecuta en Windows Vista o superior y esta disponible en plataformas de 32-bit y 64-bit. El instalador para 64-bit incluye también la parte de 32-bit. Esto significa que no tienes que instalar la parte 32-bits de manera separada, para conseguir los menus contextuales completos en aplicacione 32-bits.

Soporte para Windows 98, Windows ME y Windows NT4 se desechó en la versión 1.2.0, y para Windows 2000 y XP hasta SP2 el soporte fue desechado en la versión 1.7.0. Soporte para Windows XP con SP3 se desechó en la versión 1.9.0. Puedes todavía bajarte e instalar versiones antiguas si las necesitas.

### 1.1.2. Instalación

TortoiseSVN viene con un instalador fácil de usar. Haga doble click en el programa de instalación y siga las instrucciones. El instalador se encargará del resto. No olvide reiniciar después de la instalación.



#### Importante

Necesita privilegios de Administrador para instalar TortoiseSVN. El instalador le solicitará las credenciales si es necesario.

Están disponibles paquetes de idioma que traducen la interfaz de usuario de TortoiseSVN en varios idiomas. Por favor revise [Apéndice G, Paquetes de Idiomas y Correctores Ortográficos](#) para más información sobre cómo instalarlos.

Si encuentras algún problema durante o después de instalar TortoiseSVN por favor dirígete a nuestro FAQ en <https://tortoisesvn.net/faq.htm> [<https://tortoisesvn.net/faq.html>]

## 1.2. Conceptos básicos de control de versiones

Antes de que nos quedamos atascados en el trabajo con algunos de los archivos reales, es importante tener una visión general de cómo funciona Subversion y los términos que se utilizan.

### El repositorio

Subversion usa una base de datos central que contiene todos los archivos cuyas versiones se controlan y sus respectivas historias. Ésta base de datos se conoce como el *repositorio*. El repositorio normalmente yace en un servidor de archivos ejecutando el servidor de Subversion, que provee a pedido el contenido a los clientes de Subversion (como TortoiseSVN, por ejemplo). Si solo puede hacer una copia de seguridad de una sola cosa, hágala del repositorio, ya que es la copia maestra de toda su información.

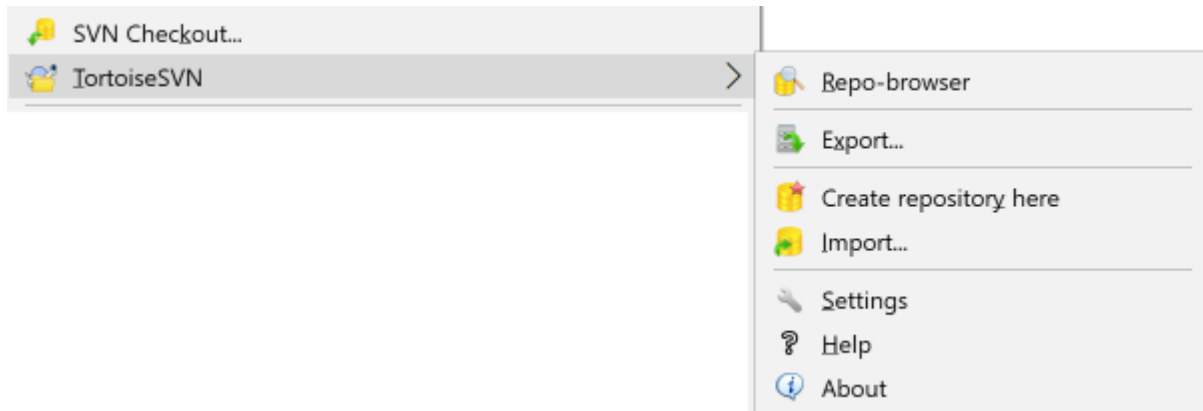
### Copia de trabajo

Acá es donde se realiza el trabajo en serio. Cada desarrollador tiene su propia copia de trabajo, comunemente conocida como caja de arena en su computador local. Usted puede obtener la última versión del repositorio, trabajar en ella localmente sin perjudicar a nadie, y cuando esté feliz con los cambios que ha realizado puede confirmar sus cambios en el repositorio.

Una copia de trabajo de Subversion no contiene la historia de un proyecto, pero sí contiene una copia de los archivos que existían en el repositorio antes que comience a hacer cambios. Esto significa que es fácil verificar qué cambios ha realizado.



También necesita saber donde encontrar TortoiseSVN dado que no hay mucho para ver en el Menú de Inicio. Esto se debe a que TortoiseSVN es una extensión de la Shell, así que primero inicie el Explorador de Windows. Hága click con el botón derecho del mouse en una carpeta del Explorador y debería ver unas entradas nuevas en el menú de contexto como estas:



**Figura 1.1. El menú de TortoiseSVN para carpetas no versionadas**

## 1.3. Prueba del producto

Esta sección demuestra como probar algunas de las características más usadas sobre un pequeño repositorio de pruebas. Naturalmente, no explica absolutamente todo - esta es la Guía de Inicio Rápido, después de todo. Una vez que esté funcionando tomese el tiempo de leer el resto de este manual de usuario, que le muestra las cosas con mucho más detalle. También le muestra las configuraciones correctas para un servidor Subversion.

### 1.3.1. Crear un Repositorio

Para un proyecto real tendrá un repositorio configurado en algún lugar seguro y un servidor Subversion que lo controle. Para este tutorial vamos a usar una característica de repositorio local de Subversion, la que permite acceso directo a un repositorio creado en su disco rígido sin necesidad de un servidor.

Para empezar cree una carpeta vacía en su PC. Puede ser en cualquier lugar, pero en este tutorial vamos a llamarlo C:\svn\_repos. Luego, haga clic derecho sobre la nueva carpeta y del menú de contexto elija TortoiseSVN → Crear un repositorio aquí.... El repositorio queda creado dentro de la carpeta, listo para su uso. También crearemos la estructura interna por defecto de la carpeta haciendo clic en el botón Crear estructura de carpeta.



#### Importante

La característica de repositorio local es muy útil para prueba y evaluación pero, a menos que esté trabajando como desarrollador único en una PC, siempre debería usar un servidor Subversión correctamente configurado. Es tentador, en una compañía pequeña, evitar el trabajo de configurar un servidor y acceder el repositorio en un recurso compartido de la red. Nunca lo haga. Perderá datos. Lea [Sección 3.1.4, “Accediendo a un repositorio en una unidad de red”](#) para enterarse porque es una mala idea hacerlo y para aprender como configurar un servidor.

### 1.3.2. Importar un Proyecto

Ahora ya tenemos un repositorio, pero está completamente vacío. Asumamos que tenemos un conjunto de archivos en C:\Projects\Widget1 que quiero agregar. Navegamos a la carpeta Widget1 en el Explorador y hacemos clic derecho sobre ella. Ahora seleccionamos TortoiseSVN → Importar... la que abre un diálogo



**Figura 1.2. El diálogo Importar**

Un repositorio Subversion se referencia por su URL, lo que nos permite especificar un repositorio en cualquier lado de Internet. En este caso necesitamos apuntar a nuestro propio repositorio local que tiene una URL `file:///c:/svn_repos/trunk`, y al cual le agregamos nuestro propio nombre de proyecto `Widget1`. Note que hay 3 barras luego de `file:` y que siempre se usan esas barras.

La otra característica importante de este diálogo es el cuadro de texto **Mensaje de Importación** que permite ingresar un mensaje que describa lo que se está haciendo. Cuando vaya a mirar la historia del proyecto, esos mensajes de actualización serán una guía valiosa de qué cambios se hicieron y porqué. En este caso podemos poner algo simple como “Importar el proyecto `Widget1`”. Haga clic en el botón **OK** y la carpeta será agregada al repositorio.

### 1.3.3. Obteniendo una Copia de Trabajo

Ahora que tenemos un proyecto en nuestro repositorio, necesitamos crear una copia de trabajo para usar en el trabajo del día a día. Note que la sola acción de importar una carpeta no convierte automáticamente esa carpeta en una copia de trabajo. El término de Subversion para la creación de una nueva copia de trabajo es **Checkout**. Vamos a hacer un checkout de la carpeta `Widget1` de nuestro repositorio a una carpeta de desarrollo en la PC llamada `C:\Projects\Widget1-Dev`. Cree esa carpeta, haga clic derecho en ella y seleccione **TortoiseSVN** → **Checkout...** Ingrese la URL para hacer el checkout, en este caso `file:///c:/svn_repos/trunk/Widget1` y finalmente haga clic en **OK**. Nuestra carpeta de desarrollo será cargada con los archivos de nuestro repositorio.



#### Importante

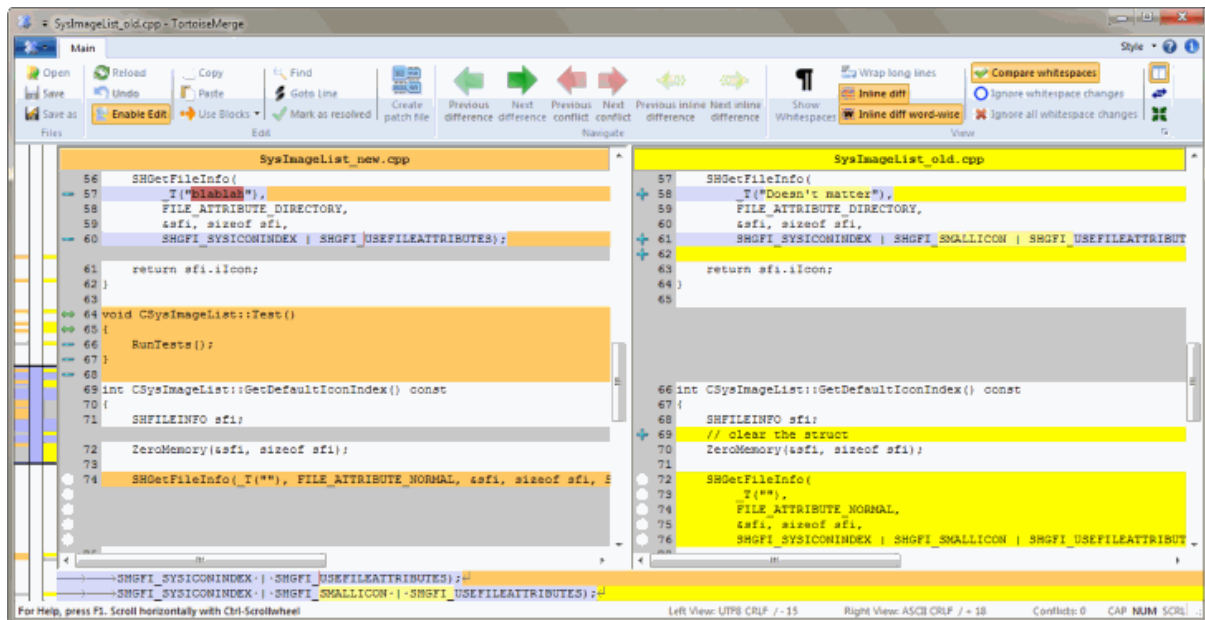
En la configuración por default, la opción salida del menú (checkout) no se encuentra en el submenú TortoiseSVN, pero se muestra en el menú superior del explorador. Los comandos de TortoiseSVN que no están en el submenú tienen **SVN** al principio: **SVN Checkout...**

Debería notarse que la apariencia de esta carpeta es diferente de nuestra carpeta original. Cada carpeta tiene una tilde verde en el vértice inferior izquierdo. Esos son los íconos de estado de TortoiseSVN que están presentes solo en las copias de trabajo. El estado verde indica que el archivo no tiene cambios de la versión del repositorio.

### 1.3.4. Haciendo Cambios

Hora de trabajar, en la carpeta `Widget1-Dev` empezamos editando ficheros - digamos que hacemos cambios en `Widget1.c` y `ReadMe.txt`. Dese cuenta que el icono de dichos ficheros ha cambiado a rojo, indicando que los cambios son locales.

¿Pero cuáles son los cambios? Hága click en uno de los archivos modificados y seleccione `TortoiseSVN → Diff`. El comparador de archivos de TortoiseSVN comienza, mostrando exactamente qué líneas han cambiado.



**Figura 1.3. Examinador de Diferencias de Archivos**

Bien, estamos contentos con los cambios, ahora actualicemos el repositorio. Esta acción se denomina *Actualizar los cambios*. Haga clic derecho sobre la carpeta `Widget1-Dev` y elija `TortoiseSVN → Actualizar`. El diálogo de actualización muestra los archivos cambiados, cada uno con una tilde. Puede elegir solo un subconjunto de esos archivos, pero en este caso vamos a actualizar los cambios en ambos archivos. Ingrese un mensaje que describa para que es el cambio y haga clic en `OK`. El diálogo de progreso muestra los archivos que están siendo actualizados al repositorio y eso es todo.

### 1.3.5. Añadiendo Más Archivos

Mientras el proyecto avanza necesitará agregar nuevos archivos - por ejemplo, porque agrega nuevas características en `Extras.c` y agregar una referencia en el archivo `Makefile` existente. Haga clic derecho en la carpeta y elija `TortoiseSVN → Agregar`. El diálogo muestra todos los archivos que aún no fueron versionados y puede escoger cuales agregar. Otra forma de agregar archivos sería hacer clic derecho sobre el archivo a agregar y seleccionar `TortoiseSVN → Agregar`.

Ahora, cuando va a actualizar la carpeta, el nuevo archivo aparece como *Agregado* y el archivo existente como *Modificado*. Note que puede hacer un doble clic en el archivo modificado para verificar exactamente que cambios se realizaron.

### 1.3.6. Viendo la Historia del Proyecto

Una de las características más útiles de TortoiseSVN es el diálogo de registro. Este muestra una lista de las confirmaciones hechas a un archivo o carpeta, y los mensajes de actualización detallados que fueron ingresados (has *ingresado* los mensajes de confirmación? Si no lo has hecho, ahora puedes darte una idea de lo importante que son).



**Figura 1.4. El Diálogo de Registro**

Aquí hice un poco de trampa y utilicé una captura de pantalla del repositorio TortoiseSVN.

El panel superior muestra una lista de las revisiones de actualización junto con la parte inicial del mensaje de actualización. Si selecciona una de esas revisiones, el panel central mostrará el mensaje completo del registro para esa revisión y el panel inferior una lista de archivos y carpetas que cambiaron.

Cada uno de esos paneles tiene un menú de contexto que permite muchas formas de usar la información. En el panel inferior puede hacer un doble clic en un archivo para ver exactamente qué cambios se hicieron en esa revisión. Lea [Sección 4.10, "Diálogo de Registro de revisiones"](#) para ver la historia completa.

### 1.3.7. Deshaciendo Cambios

Una característica de todos los sistemas de control de revisión es que permiten deshacer los cambios que se hicieron previamente. Como sería de esperar, TortoiseSVN permite realizar esto fácilmente.

Si quiere descartar los cambios que no ha actualizado y volver el archivo a como estaba cuando empezó a editarlo, TortoiseSVN → Revertir es su amigo. Este ítem descarta los cambios (a la Papelera de reciclaje, por si acaso) y revierte a la versión actualizada con la que empezó. Si quiere descartar algunos de los cambios, puede usar TortoiseMerge para ver las diferencias y revertir solo algunas líneas cambiadas.

Si desea deshacer los efectos de una revisión en particular, comience con el diálogo de registro y encuentre la revisión ofensiva. Seleccione Menú Contextual → Revertir los cambios de ésta revisión y los cambios serán revertidos.

## 1.4. Siguiendo adelante...

Esta guía le ha dado un paseo rápido por algunas de las características más importantes y útiles de TortoiseSVN, pero queda claro que hay muchas más que no hemos cubierto. Le recomendamos que se tome el tiempo de leer

el resto de este manual, especialmente **Capítulo 4, *Guía de uso diario*** donde se le da mucho más detalle a las operaciones del día a día.

Hemos trabajado mucho para asegurarnos que ésta guía es informativa y fácil de leer, pero reconocemos que hay mucho material! Tomese su tiempo y no tenga miedo de probar las cosas en un repositorio de pruebas mientras tanto. La mejor forma de aprender es usar el producto.

---

# Capítulo 2. Conceptos básicos de control de versiones

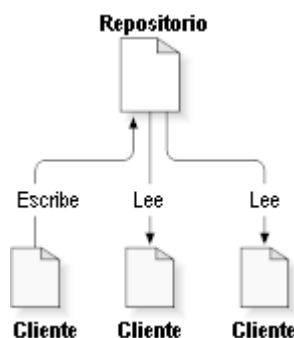
Este capítulo es una versión ligeramente modificada del mismo capítulo en el libro de Subversion. Una versión en línea del libro de Subversion está disponible aquí: <http://svnbook.red-bean.com/>.

Este capítulo es una introducción corta e informal a Subversion. Si usted es nuevo en el control de versiones, este capítulo es definitivamente para usted. Empezamos con una discusión de los conceptos generales de control de versiones, nos adentramos en las ideas específicas que hay tras Subversion, y mostramos algunos ejemplos sencillos del uso de Subversion.

Aún cuando los ejemplos en este capítulo muestran gente compartiendo colecciones de código fuente de programas, tenga en cuenta que Subversion puede manejar cualquier colección de archivos - no está limitado a ayudar a los programadores.

## 2.1. El repositorio

Subversion es un sistema centralizado para compartir información. En su núcleo está un *repositorio*, que es un almacén central de datos. El repositorio almacena información en forma de un *árbol de archivos* - una jerarquía típica de archivos y directorios. Un número de *clientes* se conectan al repositorio, y luego leen o escriben esos archivos. Al escribir datos, el cliente hace que la información esté disponible para los otros; al leer los datos, el cliente recibe información de los demás.



**Figura 2.1. Un sistema típico cliente/servidor**

¿Y esto por qué es interesante? Por ahora, eso suena a la definición típica de un servidor de archivos típico. Y de hecho, el repositorio *es* una clase de servidores de archivos, pero no el habitual. Lo que hace al repositorio de Subversion especial es que *recuerda todos los cambios* que alguna vez se hayan escrito en él: cada cambio en cada archivo, e incluso los cambios en el propio árbol de directorios, como el añadir, borrar o reorganizar archivos y directorios.

Cuando un cliente lee datos de un repositorio, normalmente ve solo la última versión del árbol de archivos. Sin embargo el cliente también tiene la capacidad de ver los estados *previos* del sistema de archivos. Por ejemplo, un cliente podría hacer preguntas como “¿Qué contenía este directorio este miércoles?” o “¿quién fue la última persona en editar este archivo y qué cambios realizó?” Estas son el tipo de preguntas que deberían estar en el epicentro de cualquier *sistema de control de versiones*: sistemas diseñados para guardar y registrar las modificaciones de los datos a lo largo del tiempo.

## 2.2. Modelos de versionado

Todos los sistemas de control de versiones tienen que resolver los mismos problemas fundamentales: ¿cómo permitirá el sistema compartir información entre usuarios, pero evitando que ellos accidentalmente se pisen unos a otros? Es demasiado sencillo que los usuarios accidentalmente sobrescriban los cambios del otro en el repositorio.

### 2.2.1. El problema de compartir archivos

Considere este escenario: suponga que tiene dos compañeros de trabajo, Harry y Sally. Cada uno decide editar el mismo archivo del repositorio a la vez. Si Harry graba sus cambios en el repositorio primero, es posible que (unos momentos después) Sally pueda accidentalmente sobrescribirlos con su propia versión nueva del archivo. Mientras que la versión del archivo de Harry no se ha perdido para siempre (porque el sistema recuerda cada cambio), cualquier cambio que Harry hizo *no estará* en la versión nueva del archivo de Sally, porque para empezar ella nunca vió los cambios de Harry. El trabajo de Harry está aún efectivamente perdido - o al menos falta en la última versión del archivo - y probablemente por accidente. ¡Esta es una situación que definitivamente tenemos que evitar!

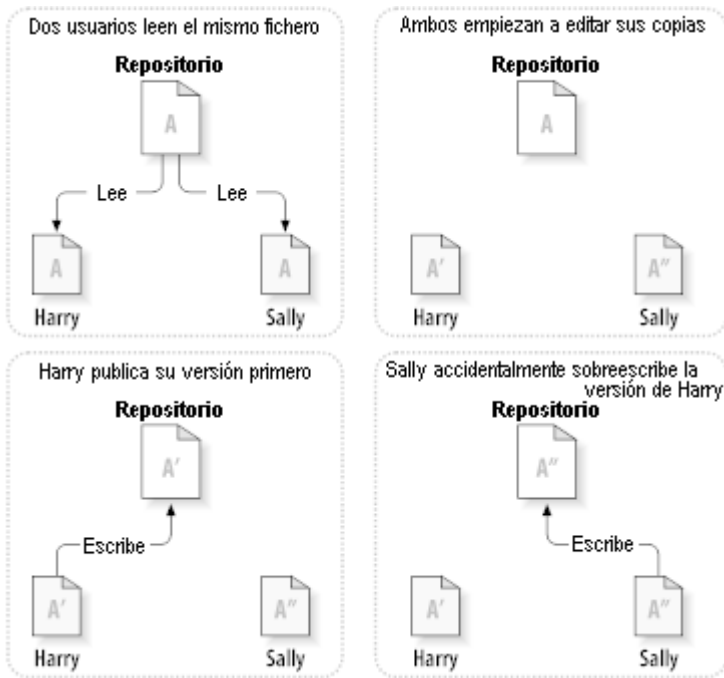
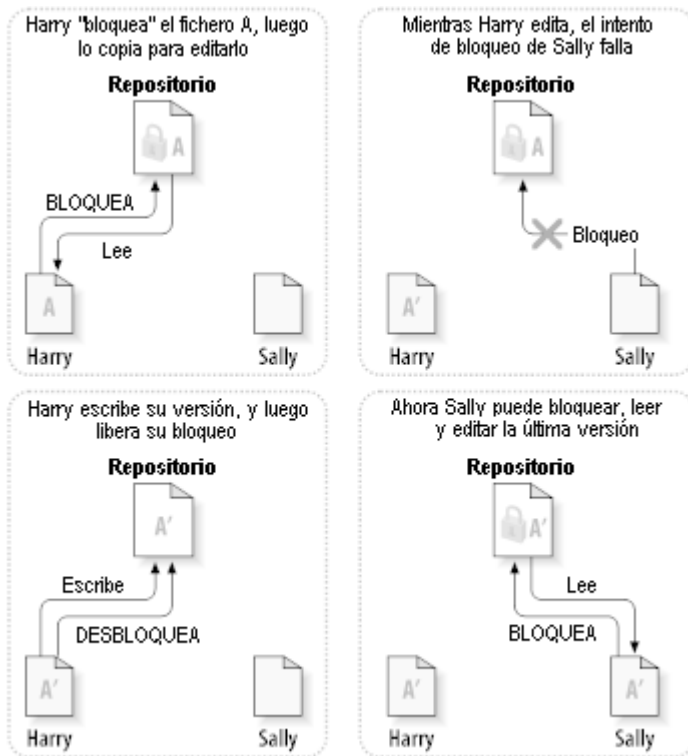


Figura 2.2. El problema a evitar

### 2.2.2. La solución bloquear-modificar-desbloquear

Muchos sistemas de control de versiones utilizan un modelo *bloquear-modificar-desbloquear* para enfrentarse al problema, lo cual es una solución muy simple. En estos sistemas, el repositorio sólo permite que una persona cambie un archivo. Harry primero debe *bloquear* el archivo antes que pueda empezar a hacer cambios en él. Bloquear un archivo se parece mucho a tomar prestado un libro de la biblioteca; si Harry ha bloqueado un archivo, entonces Sally no puede hacer ningún cambio en él. Si ella intenta bloquear el archivo, el repositorio le denegará la petición. Todo lo que ella puede hacer es leer el archivo, y esperar a que Harry termine sus cambios y libere su bloqueo. Después que Harry desbloquee el archivo, se acabó su turno, y ahora es el turno de Sally para bloquear y editar.



**Figura 2.3. La solución bloquear-modificar-desbloquear**

El problema con el modelo bloquear-modificar-desbloquear es que es un poco restrictivo, y a menudo se convierte en una molestia para los usuarios:

- *El bloqueo causa muchos problemas administrativos.* A veces Harry bloqueará un archivo y luego se olvidará de ello. Mientras tanto, dado que Sally está aún esperando para editar el archivo, sus manos están atadas. Y luego Harry se va de vacaciones. Ahora Sally tiene que buscar a un administrador para que libere el bloqueo de Harry. La situación acaba causando un montón de retraso y pérdida de tiempo innecesarios.
- *El bloqueo puede causar procesos en serie innecesarios.* ¿Qué ocurre si Harry está editando el inicio de un archivo de texto, y Sally simplemente quiere cambiar la parte final del mismo archivo? Esos cambios no se superponen en absoluto. Ellos podrían fácilmente editar el archivo de forma simultánea, y no habría ningún daño, asumiendo que los cambios se fusionaran correctamente. No hay necesidad de tomar turnos en esta situación.
- *El bloqueo puede causar una falsa sensación de seguridad.* Imagine que Harry bloquea y edita el archivo A, mientras Sally simultáneamente bloquea y edita el archivo B. Pero suponga que A y B dependen uno del otro, y que los cambios hechos a cada uno son semánticamente incompatibles. De repente A y B ya no funcionan juntos. El sistema de bloqueo no tiene forma de prevenir este problema - sin embargo, de alguna forma dió una sensación de falsa seguridad. Es fácil para Harry y Sally imaginar que al bloquear los archivos, cada uno está empezando una tarea segura y aislada, y por tanto les inhibe de discutir sus cambios incompatibles oportunamente.

### 2.2.3. La solución copiar-modificar-fusionar

Subversion, CVS y otros sistemas de control de versiones usan un modelo *copiar-modificación-fusión* como una alternativa de bloqueo. En este modelo, cada cliente de los usuarios lee el repositorio y crea una *copia de trabajo* personal del fichero o proyecto. Los usuarios trabajan en paralelo, modificando sus copias privadas. Finalmente, las copias privadas son unificadas conjuntamente en una nueva, versión final. El sistema de control de versiones normalmente asesora con la unificación, pero en última instancia un humano es el responsable de hacer esta acción de manera correcta.

Aquí hay un ejemplo. Digamos que tanto Harry como Sally crean copias de trabajo del mismo proyecto, copiado del repositorio. Ellos trabajan simultáneamente, y hacen cambios al mismo archivo A dentro de sus copias. Sally es la primera en grabar sus cambios en el repositorio. Cuando Harry intenta grabar sus cambios más tarde, el



repositorio le informa que su archivo A está *desactualizado*. En otras palabras, que el archivo A en el repositorio ha cambiado de alguna forma desde la última vez que lo copió. Por lo que Harry le pide a su cliente que *fusion*e cualquier nuevo cambio del repositorio dentro de su copia de trabajo del archivo A. Lo más seguro es que los cambios de Sally no se superpongan a los suyos; por lo que una vez que ambos conjuntos de cambios se han integrado, él graba su copia de trabajo de nuevo en el repositorio.

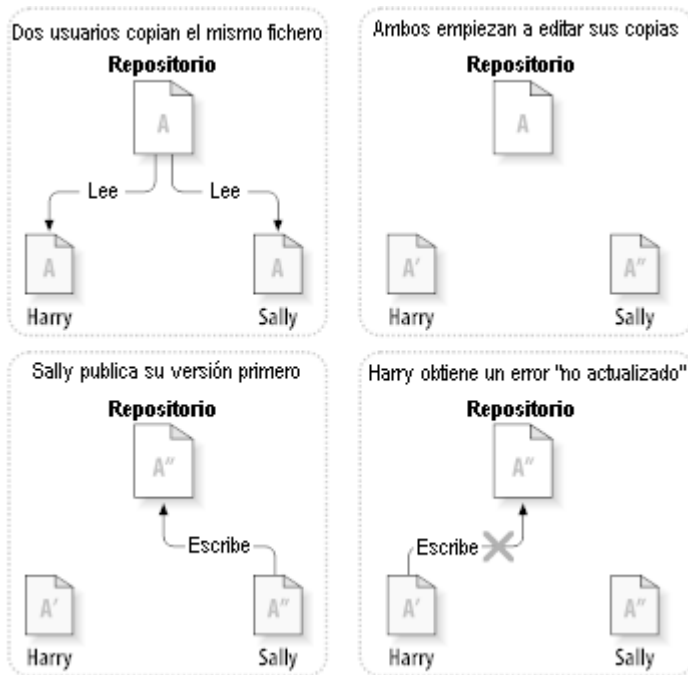


Figura 2.4. La solución copiar-modificar-fusionar

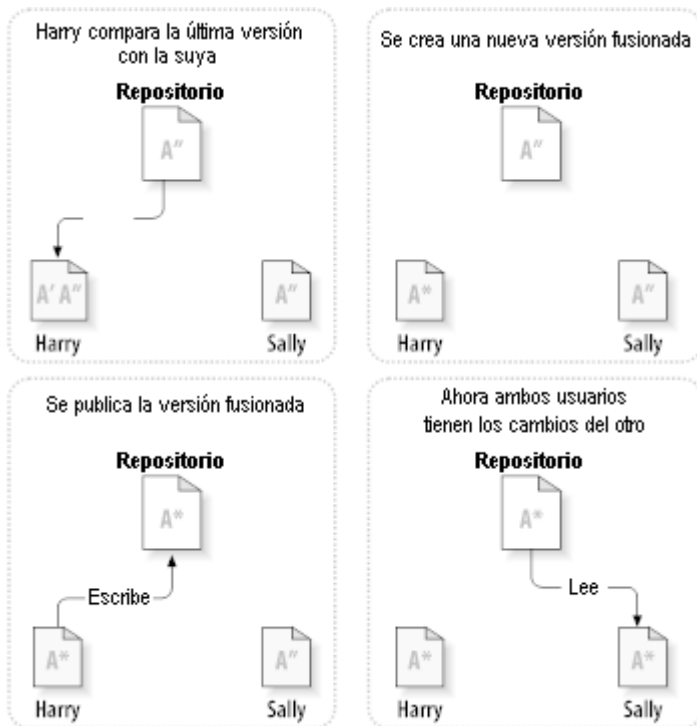


Figura 2.5. ... continuación de Copiar-modificar-fusionar

¿Pero qué ocurre si los cambios de Sally *sí* se superponen a los cambios de Harry? ¿Qué hacemos entonces? La situación se denomina un *conflicto*, y normalmente no es mucho problema. Cuando Harry le pide a su

cliente que fusione los últimos cambios del repositorio en su copia de trabajo, su copia del archivo A se marca de alguna forma como que está en un estado de conflicto: él será capaz de ver ambos conjuntos de cambios conflictivos, y manualmente podrá elegir entre ellos. Tenga en cuenta que el software no puede resolver conflictos automáticamente; sólo los humanos son capaces de entender y hacer las elecciones necesarias de forma inteligente. Una vez que Harry haya resuelto manualmente los cambios que se superponían (¡quizás discutiendo el conflicto con Sally!), puede guardar de forma segura el archivo fusionado al repositorio.

El modelo copiar-modificar-fusionar puede sonar un poco caótico, pero en la práctica, funciona extremadamente bien. Los usuarios pueden trabajar en paralelo, sin que tengan que esperar nunca uno por otro. Cuando trabajan en los mismos archivos, resulta que la mayoría de los cambios concurrentes no se superponen en absoluto; los conflictos no son frecuentes. Y el tiempo que lleva resolver conflictos es mucho menor que el tiempo perdido por un sistema bloqueante.

Al final, todo se reduce a un factor crítico: la comunicación entre usuarios. Cuando los usuarios se comunican de forma pobre, aumentan los conflictos sintácticos y semánticos. No hay sistema capaz de forzar a los usuarios a comunicarse perfectamente, y no hay sistema que pueda detectar conflictos semánticos. Por lo que no hay motivo para que se le prometa falsamente que un sistema con bloqueos prevendrá de alguna forma los conflictos; en la práctica, el bloqueo parece inhibir la productividad más que otra cosa.

Hay una situación común donde el modelo bloquear-modificar-desbloquear resulta mejor, y es cuando tiene archivos no-fusionables. Por ejemplo si su repositorio contiene algunas imágenes gráficas, y dos personas cambian la imagen a la vez, no hay forma de fusionar esos cambios. O Harry o Sally perderán sus cambios.

## 2.2.4. ¿Qué hace Subversion?

Subversion utiliza la solución copiar-modificar-mezclar por defecto, y en muchos casos esto es todo lo que necesitará. Sin embargo, desde la Versión 1.2, Subversion también admite bloqueo de archivos, por lo que si tiene archivos no-fusionables, o si simplemente está forzado a una política de bloqueo por la dirección, Subversion seguirá teniendo las características que necesita.

## 2.3. Subversion en acción

### 2.3.1. Copias de trabajo

Ya ha oído hablar sobre las copias de trabajo; ahora le demostraremos cómo las crea y las utiliza el cliente de Subversion.

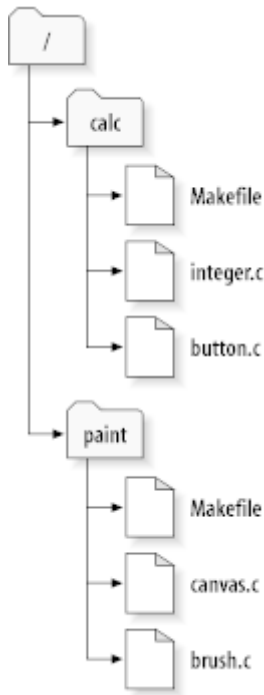
Una copia de trabajo de Subversion es un árbol de directorios ordinario en su sistema local, conteniendo una colección de archivos. Puede editar estos archivos como desee, y si son archivos de código fuente, puede compilar su programa de la forma habitual. Su copia de trabajo es su área de trabajo privada: Subversion nunca incorporará los cambios de otra gente, ni hará que sus cambios estén disponibles para los demás, a menos que se lo pida explícitamente.

Después que haya hecho algunos cambios en los archivos dentro de su copia de trabajo y haya verificado que funcionan correctamente, Subversion le provee de comandos para *publicar* sus cambios para los que trabajan con usted en su proyecto (escribiendo en el repositorio). Si los demás publican sus propios cambios, Subversion le provee de comandos para fusionar esos cambios dentro de su copia de trabajo (leyendo desde el repositorio).

Una copia de trabajo también contiene algunos ficheros extra, creados y mantenidos por Subversion, para ayudar a llevar a cabo estos comandos. En particular, tu copia de trabajo contiene un subdirectorío llamado `.svn`, también conocido como el *directorío administrativo* de la copia de trabajo. Los ficheros en este directorío administrativo ayudan a Subversion a reconocer qué ficheros contienen cambios no publicados, y qué ficheros contienen contenidos desfasados respecto a otros de trabajo. Antes de Subversion 1.7 existían subdirectoríos administrativos `.svn` en cada directorío versionado de tu copia de trabajo. Subversion 1.7 se comporta de manera diferente, usando un único subdirectorío dependiente del root de la copia de trabajo.

Un repositorio típico de Subversion a menudo contiene los archivos (o el código fuente) de varios proyectos; usualmente, cada proyecto es un subdirectorío en el árbol de archivos del repositorio. Con esta disposición, una copia de trabajo de un usuario normalmente corresponderán a un subárbol particular del repositorio.

Por ejemplo, suponga que tiene un repositorio que contiene dos proyectos de software.



**Figura 2.6. El sistema de archivos del repositorio**

En otras palabras, el directorio raíz del repositorio tiene dos subdirectorios, `paint` y `calc`.

Para obtener una copia de trabajo, debes *desproteger* algún subárbol del repositorio. (El término *desproteger* puede sonar como que tuviera algo que hacer con bloqueos o recursos, pero no es necesario; simplemente crea una copia privada del proyecto para tí.)

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 sus cambios para los demás, puede utilizar el comando de Subversion **commit**.

Ahora que sus cambios a `button.c` se han confirmado en el repositorio, si cualquier otro usuario obtiene una copia de trabajo de `/calc`, verá sus cambios en la última versión del archivo.

Suponga que tiene un colaborador, Sally, quien obtuvo una copia de trabajo de `/calc` al mismo tiempo que usted. Cuando ha confirmado sus cambios en `button.c`, la copia de trabajo de Sally se queda sin cambios; Subversion sólo modifica las copias de trabajo cuando lo pide el usuario.

Para poner al día su proyecto, Sally puede pedirle a Subversion *actualizar* su copia de trabajo, utilizando el comando de Subversion **actualizar**. Esto incorporará sus cambios en la copia de trabajo de Sally, junto con cualquier otro que se haya confirmado desde que ella lo obtuvo.

Tenga en cuenta que Sally no necesita especificar qué archivos actualizar; Subversion utiliza la información en el directorio `.svn`, y más información desde el repositorio, para decidir qué archivos deben ponerse al día.

### 2.3.2. URLs de repositorio

Los repositorios de Subversion pueden ser accedidos por muchos métodos diversos - en discos locales, o a través de varios protocolos de red. La ruta de un repositorio es siempre, una URL. El esquema URL indica el método de acceso:

Esquema	Método de acceso
<code>file://</code>	Acceso directo al repositorio en el disco local o de red.
<code>http://</code>	Acceso utilizando el protocolo WebDAV a un servidor Apache configurado para Subversion.
<code>https://</code>	Lo mismo que <code>http://</code> , pero con encriptación SSL.
<code>svn://</code>	Acceso TCP/IP sin autenticación utilizando un protocolo personalizado a un servidor <code>svnserve</code> .
<code>svn+ssh://</code>	Acceso TCP/IP autenticado y encriptado utilizando un protocolo propio a un servidor <code>svnserve</code> .

**Tabla 2.1. URLs de acceso al repositorio**

En su mayoría, las URLs de Subversion utilizan la sintaxis estándar, permitiendo que se especifiquen nombres de servidor y números de puertos como parte de la URL. El método de acceso `file://` se utiliza normalmente para el acceso local, aunque puede utilizarse con rutas UNC a equipos en red. La URL por lo tanto toma la forma `file://nombredeequipo/ruta/al/repositorio`. Para la máquina local, la parte `nombredeequipo` de la URL debe estar o ausente o ser `localhost`. Por esta razón, las rutas locales normalmente aparecen con tres barras, `file:///ruta/al/repositorio`.

Además, los usuarios del esquema `file://` en las plataformas Windows necesitarán utilizar una sintaxis “estándar” no oficial para acceder a los repositorios que están en la misma máquina, pero en una letra de unidad diferente de la unidad actual del cliente. Cualquiera de las siguientes sintaxis de URL funcionarán, donde X es la unidad en la que reside el repositorio:

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

Tenga en cuenta que las URLs utilizan las barras hacia delante (las de dividir) incluso aunque la forma nativa (no-URL) de una ruta en Windows utiliza las barras contrarias.

Puede acceder a un repositorio FSFS utilizando una carpeta compartida de red, pero *no* es recomendable por varias razones:

- Usted está dando permisos de escritura a todos los usuarios, así ellos pueden accidentalmente eliminar o corromper el repositorio del sistema.
- No todos los protocolos de compartición de ficheros soportan el bloqueo que Subversion requiere. Un día se encontrará su repositorio tendrá el subtítulo de *corrupto*.
- Usted tiene que establecer los permisos de acceso del modo correcto. SAMBA es particularmente complicado a este respecto.
- Si una persona instala una versión más moderna del cliente que modifica el formato del repositorio, entonces todo el mundo no podrá acceder al repositorio hasta que se actualicen el cliente a la nueva versión.

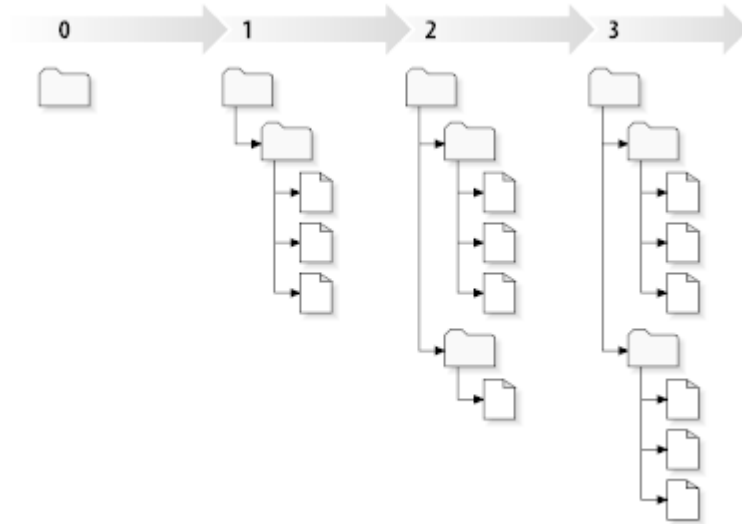
### 2.3.3. Revisiones

Una operación **svn commit** puede publicar los cambios de cualquier número de archivos y carpetas como una única transacción atómica. En su copia de trabajo, puede cambiar el contenido de los archivos, crear, borrar, renombrar y copiar archivos y directorios, y luego confirmar el conjunto completo de cambios como una unidad.

En el repositorio, cada confirmación se trata como una transacción atómica: o bien todos los cambios de la confirmación se llevan a cabo, o bien ninguno de ellos se realiza. Subversion aplica esta atomicidad en caso de errores en el programa, errores del sistema, problemas de red y otras acciones del usuario.

Cada vez que el repositorio acepta una confirmación, crea un nuevo estado del árbol de archivos, llamado *revisión*. A cada revisión se le asigna un número natural único, un número mayor que la revisión anterior. La revisión inicial de un repositorio recién creado se numera como cero, y consiste únicamente en un directorio raíz vacío.

Una buena forma de visualizar el repositorio es como una serie de árboles. Imagine una fila de números de revisiones, empezando en 0, de izquierda a derecha. Cada número de revisión tiene un árbol colgando debajo, y cada árbol es una “foto” de cómo estaba el repositorio tras cada confirmación.



**Figura 2.7. El repositorio**

#### Números globales de revisión

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!

Es importante que tenga en cuenta que las copias de trabajo no siempre se corresponden a una única revisión en el repositorio; pueden contener archivos de varias revisiones. Por ejemplo, suponga que obtiene una copia de trabajo de un repositorio cuya revisión más reciente es la 4:

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

En este momento, esta copia de trabajo corresponde exactamente a la revisión 4 en el repositorio. Sin embargo, suponga que ha hecho cambios al archivo `button.c`, y confirme ese cambio. Asumiendo que no se haya llevado a cabo ningún otro cambio, su confirmación creará la revisión 5 en el repositorio, y su copia de trabajo quedará así:

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

Suponga que, en este punto, Sally confirma un cambio a `integer.c`, creando la revisión 6. Si utiliza **svn update** para actualizar su copia de trabajo, obtendrá ésto:

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

Los cambios de Sally a `integer.c` aparecerán en su copia de trabajo, y su cambio estará aún presente en `button.c`. En este ejemplo, el texto de `Makefile` es idéntico en las revisiones 4, 5, y 6, pero Subversion marcará su copia de trabajo de `Makefile` con la revisión 6 para indicar que aún está actualizado. Por lo que, después que haga una actualización limpia en la parte superior de su copia de trabajo, generalmente corresponderá exactamente una revisión del repositorio.

### 2.3.4. Cómo reconocen las copias de trabajo al repositorio

Por cada archivo en un directorio de trabajo, Subversion grabará dos piezas esenciales de información en el área administrativa `.svn/`:

- qué número de revisión tiene su fichero de trabajo ( esto es llamado la *revisión de trabajo* del fichero), y
- una fecha que indica cuándo se actualizó por última vez la copia local por el repositorio.

Con esta información, hablando con el repositorio, Subversion puede decirle en cuál de los siguientes cuatro estados está un archivo de trabajo:

#### Sin cambios, y actualizado

El archivo no se ha cambiado en el directorio de trabajo, y no se han confirmado cambios a ese archivo en el repositorio desde su revisión de trabajo. Una **confirmación** de ese archivo no hará nada, y una **actualización** de ese archivo no hará nada.

#### Cambiado localmente, y actualizado

El archivo ha sido cambiado en el directorio de trabajo, y no se ha confirmado ningún cambio a ese archivo en el repositorio desde su revisión base. Hay cambios locales que no se han confirmado en el repositorio, por lo que al **confirmar** el archivo se conseguirá publicar sus cambios, y al **actualizar** el archivo no se realizará nada.

#### Sin cambios, y desactualizado

El archivo no ha sido cambiado en el directorio de trabajo, pero ha sido cambiado en el repositorio. El archivo deberá ser actualizado en algún momento, para actualizarlo con la revisión pública. Un comando **confirmar** sobre el archivo no hará nada, y al **actualizar** el archivo se traerán los últimos cambios a su copia de trabajo.

#### Cambiado localmente, y desactualizado

El archivo se ha cambiado tanto en el directorio de trabajo como en el repositorio. Un comando **confirmar** sobre el archivo fallará con un error *desactualizado*. El archivo debería actualizarse primero; al **actualizar** se intentará fusionar los cambios públicos con los cambios locales. Si Subversion no puede completar la fusión de una forma plausible automáticamente, le dejará al usuario la tarea de resolver el conflicto.

## 2.4. Sumario

Hemos cubierto un número de conceptos fundamentales de Subversion en este capítulo:

- Hemos introducido las nociones de un repositorio central, la copia de trabajo del cliente, y la lista de árboles de revisiones del repositorio.
- Hemos visto algunos ejemplos simples sobre cómo dos colaboradores pueden utilizar Subversion para publicar y recibir los cambios de uno a otro, utilizando el modelo 'copiar-modificar-fusionar'.
- Hemos hablado un poco sobre la forma en la que Subversion controla y maneja la información en una copia de trabajo.

---

# Capítulo 3. El repositorio

Sin importar el protocolo que use para acceder a sus repositorios, siempre necesita crear al menos un repositorio. Esto puede hacerse o bien con el cliente de línea de comandos de Subversion o con TortoiseSVN.

Si todavía no ha creado un repositorio de Subversion, ahora es el momento de hacerlo.

## 3.1. Creación de repositorios

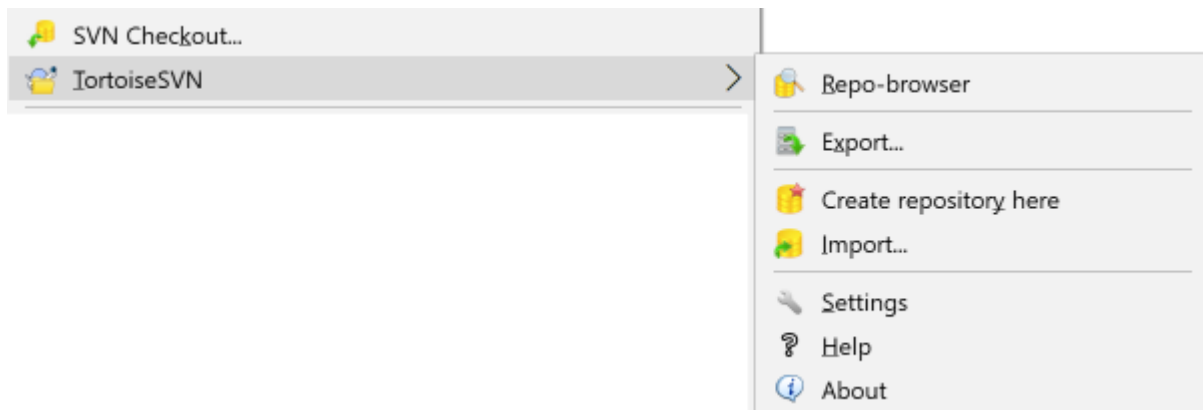
### 3.1.1. Creando un repositorio con el cliente de línea de comandos

1. Cree una carpeta vacía con el nombre SVN (por ejemplo, D:\SVN\), que se usará como la raíz de todos sus repositorios.
2. Cree otra carpeta MiNuevoRepositorio dentro de D:\SVN\
3. Abra el símbolo del sistema (Ventana de comandos DOS), cambie a D:\SVN\ y escriba

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

Ahora ya tiene un nuevo repositorio que se encuentra en D:\SVN\MiNuevoRepositorio.

### 3.1.2. Creando el repositorio con TortoiseSVN



**Figura 3.1. El menú de TortoiseSVN para carpetas no versionadas**

1. Abra el explorador de Windows
2. Cree una nueva carpeta y llámela por ejemplo SVNRepositorio
3. Haga click con el botón derecho sobre la carpeta recién creada y seleccione TortoiseSVN → Crear Repositorio aquí...

Entonces se creará un repositorio dentro de la nueva carpeta. *¡¡¡No edite los archivos!!!*. Si se encuentra con algún error asegúrese que la carpeta esté vacía y que no esté protegida contra escritura.

También se le preguntará si quiere crear una estructura de directorio dentro del repositorio. Encuentre más sobre opciones de disposición en [Sección 3.1.5, “Organización del repositorio”](#).

TortoiseSVN colocará un ícono de carpeta personalizado cuando crea un repositorio para que usted pueda identificar los repositorios locales más fácilmente. Si crea un repositorio usando el cliente de línea de comando oficial este ícono de carpeta no se asigna.



## Sugerencia

También recomendamos que no use el acceso `file://`, a excepción de un uso local. Usando un servidor es más seguro y de confianza para todos los escenarios excepto para un uso monodesarrollador.

### 3.1.3. Acceso local al repositorio

Para acceder a su repositorio local, necesita la ruta a esa carpeta. Recuerde que Subversion espera todas las rutas de repositorios con el formato `file:///C:/RepositorioSVN/`. Tenga en cuenta el uso de las barras de dividir.

Para acceder a un repositorio que se encuentre en una unidad de red compartida puede o bien utilizar mapeado de unidades, o bien usar la ruta UNC. El formato de una ruta UNC es `file://NombreDelServidor/ruta/al/repositorio/`. Observe que sólo hay 2 barras invertidas aquí.

Antes de SVN 1.2, las rutas UNC tenían que estar dadas en la forma más oscura `file:///\\NombreDelServidor/ruta/al/repositorio`. Esta forma aún se puede utilizar, pero no está recomendada.

### 3.1.4. Accediendo a un repositorio en una unidad de red

Aunque es posible poner un repositorio FSFS en una red, compartir y tener acceso concurrente a él usando el protocolo `file://`, decididamente *no* es algo recomendable. De hecho nosotros *especialmente* lo desaconsejamos, y no damos soporte a dicho uso por varios motivos:

- En primer lugar, está dando a todos los usuarios acceso directo de escritura en el repositorio, por lo que cualquier usuario podría accidentalmente borrar el repositorio completo o hacerlo inutilizable de cualquier otra forma.
- En segundo lugar, no todos los protocolos de compartición de archivos de red soportan el bloqueo que Subversion necesita, por lo que puede encontrar que su repositorio se corrompe. Puede que no ocurra al principio, pero un día dos usuarios intentarán acceder al repositorio al mismo tiempo.
- En tercer lugar, también debe establecer los permisos de los archivos. Puede ser sencillo en una unidad de red nativa de Windows, pero en SAMBA esto es particularmente difícil.
- Si una persona instala una versión más moderna del cliente que modifica el formato del repositorio, entonces todo el mundo no podrá acceder al repositorio hasta que se actualicen el cliente a la nueva versión.

El acceso `file://` está pensado únicamente para el acceso local por un único usuario, en particular para tests y depuraciones. Cuando desee compartir el repositorio, *realmente* necesita configurar un servidor de forma apropiada, y realmente no es tan difícil como pueda pensar. Lea [Sección 3.5, “Accediendo al repositorio”](#) para obtener indicaciones sobre cómo elegir y configurar un servidor.

### 3.1.5. Organización del repositorio

Antes de importar sus datos al repositorio, primero debería pensar cómo quiere organizar sus datos. Si utiliza uno de los patrones recomendados lo tendrá luego mucho más fácil.

Hay algunas formas estándar y recomendadas de organizar un repositorio. La mayoría de la gente crea un directorio `trunk` (tronco) para alojar la “línea principal” del desarrollo, un directorio `branches` (ramas) para que contenga las copias/ramas, y un directorio `tags` (etiquetas) para contener las copias/etiquetas. Si un repositorio sólo aloja un proyecto, a menudo la gente crea estos directorios en el primer nivel:

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

Dado que esta disposición es usada tan comúnmente, cuando crea un nuevo repositorio usando TortoiseSVN, se le ofrecerá también crear la estructura de directorio por usted.



Si un repositorio contiene múltiples proyectos, la gente a menudo indexa por ramas:

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

...o por proyecto:

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

Indexar por proyecto tiene sentido si los proyectos no están muy relacionados y cada uno se obtiene de forma individual. Para proyectos relacionados donde puede querer obtener todos los proyectos de golpe, o donde los proyectos están unidos y forman un paquete de distribución único, a menudo es mejor indexar por rama. De esta forma sólo tendrá un tronco para obtener, y las relaciones entre sub-proyectos se ven más fácilmente.

Si adopta una aproximación de primer nivel `/trunk /tags /branches`, no es necesario decir que tendrá que copiar el tronco completo para cada rama y etiqueta, y de alguna forma esta estructura ofrece la mayor flexibilidad.

Para proyectos que no tienen que ver, puede preferir utilizar repositorios separados. Cuando confirma los cambios, lo que cambia es el número de revisión del repositorio completo, no el número de revisión del proyecto. Si tiene 2 proyectos que no tengan que ver compartiendo un repositorio, eso puede llevar a que ocurran grandes lagunas en los números de revisión. Los proyectos Subversion y TortoiseSVN aparecen en el mismo servidor, pero son dos repositorios totalmente separados que permiten un desarrollo independiente, y no hay confusión sobre los números de compilación.

Por supuesto, tiene libertad para ignorar estos patrones comunes. Puede crear cualquier variación, la que mejor le venga a usted o a su equipo. Recuerde que cualquiera que sea la que elija, no es una elección inamovible. Puede reorganizar su repositorio en cualquier momento. Dado que las ramas y las etiquetas son directorios normales, TortoiseSVN puede mover o renombrarlas como desee.

Cambiar de una disposición a otra es sólo una cuestión de ejecutar una serie de movimientos en el lado del servidor; si no le gusta la forma en la que están organizadas las cosas en el repositorio, sólo tiene que ir moviendo los directorios.

Así que si no ha creado todavía una estructura básica de carpetas dentro de su repositorio, debería hacerlo ahora. Hay dos formas de conseguirlo. Si simplemente desea crear una estructura `/trunk /tags /branches`, puede utilizar el visor de repositorios para crear las tres carpetas (en tres confirmaciones distintas). Si desea crear una jerarquía más profunda, es más fácil crear primero una estructura de carpetas en el disco e importarla en una única confirmación, por ejemplo:

1. Cree una nueva carpeta en su disco duro
2. Cree la estructura de carpetas de primer nivel dentro de esa carpeta - ¡no ponga ningún archivo allí todavía!
3. Importe esta estructura en el repositorio via click con el botón derecho en la carpeta y seleccionando TortoiseSVN → Importar.... Esto importará su carpeta temporal a la raíz del repositorio para crear la estructura básica del repositorio.

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. Copia de seguridad del Repositorio

Cualquiera que sea el tipo de repositorio que use, es de vital importancia que mantenga copias de seguridad regulares, y que verifique la copia. Si el servidor falla, puede ser capaz de acceder a la versión más reciente de sus archivos, pero sin el repositorio toda su historia se perderá para siempre.

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.

El método recomendado es ejecutar

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

para crear una copia de tu repositorio de manera segura. Después salvaguarda la copia.

La herramienta `svnadmin` se instala automáticamente cuando instalas el cliente Subversion. La manera más fácil de obtener esto es marcar la opción de la herramienta de línea de comandos cuando instalas TortoiseSVN, pero si prefieres puedes descargar la última versión de la herramienta de línea de comandos directamente del website de [Subversion](https://subversion.apache.org/packages.html#windows) [https://subversion.apache.org/packages.html#windows].

## 3.3. Scripts gancho en el lado del servidor

Un script gancho (N.d.T.: *hook* en inglés) es un programa lanzado desde algún evento de repositorio, como la creación de una nueva revisión o la modificación de una propiedad no versionada. Cada gancho maneja suficiente información para decir de que evento se trata, con que destino(s) está operando, y el nombre de usuario de la persona que lanzó el evento. Dependiendo del resultado del gancho o del estado de la respuesta, el programa gancho puede continuar la acción, detenerla o suspenderla de alguna manera. Por favor diríjase al capítulo [Scripts gancho](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] en el libro de Subversion para ver todos los detalles acerca de los ganchos que están implementados.

Estos scripts gancho se ejecutan por el servidor que hospeda el repositorio. TortoiseSVN también le permite configurar scripts ganchos del lado del cliente que se ejecutan localmente en ciertos eventos. Para más información consulte [Sección 4.31.8, “Scripts gancho del lado del cliente”](#).

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`.



### Ignorando Ganchos

Si un script gancho rechaza su confirmación, entonces ésta es una decisión final. Pero se puede construir un mecanismo en el script mismo utilizando la técnica de la *Palabra Mágica*. Si el script quiere rechazar una operación, buscará primero en el mensaje de registro una frase específica, ya sea ésta una frase fija o quizás el nombre del archivo con un prefijo. Si encuentra la palabra mágica entonces procederá a permitir la confirmación. Si la frase no se encuentra puede bloquear la confirmación con un mensaje como “No dijiste la palabra mágica” :-)

## 3.4. Enlaces de obtener

Si desea hacer disponible su repositorio de Subversion a los demás, puede que desee incluir un vínculo a él desde su sitio web. Una forma de hacer esto más accesible es incluir un *enlace de obtener* para otros usuarios de TortoiseSVN.

Cuando se instala TortoiseSVN, se registra un nuevo protocolo `tsvn:`. Cuando un usuario de TortoiseSVN pulsa en un enlace con ese protocolo, aparece el diálogo obtener automáticamente con la URL del repositorio ya escrita.

Para incluir dicho enlace en su propia página HTML, necesita añadir un código que será similar a este:

```
<a href="tsvn:http://proyecto.dominio.org/svn/trunk">
</a>
```

por supuesto que podría parecer incluso mejor sin cluye es una imagen adecuada. Puedes usar el [logo orteoiseSVNN](https://tortoisesvn.net/images/TortoiseCheckout.png) [https://tortoisesvn.net/images/TortoiseCheckout.png] o puedes proporcionar tu propia imagen.

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

También puede hacer el enlace apuntando a una revisión en concreto, por ejemplo

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

## 3.5. Accediendo al repositorio

Para utilizar TortoiseSVN (o cualquier otro cliente de Subversion), necesita un lugar donde poner sus repositorios. Puede o bien almacenar sus repositorios de forma local y acceder a ellos utilizando el protocolo `file://`, o puede ponerlos en un servidor y acceder a ellos con los protocolos `http://` o `svn://`. Los dos protocolos de servidor también pueden estar encriptados. Utilice `https://` o `svn+ssh://`, o puede utilizar `svn://` con SASL.

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

Si no tiene un servidor y trabaja solo, o si está simplemente evaluando Subversion y TortoiseSVN aisladamente, los repositorios locales son probablemente su mejor elección. Cree sencillamente un repositorio en su propio PC

como se describe más arriba en [Capítulo 3, \*El repositorio\*](#). Puede saltar el resto de este capítulo e ir directamente a [Capítulo 4, \*Guía de uso diario\*](#) para averiguar cómo empezar a utilizarlo.

Si estaba pensando en configurar un repositorio multiusuario en una unidad compartida de red, piénselo de nuevo. Lea [Sección 3.1.4, “Accediendo a un repositorio en una unidad de red”](#) para averiguar por qué creemos que eso es una mala idea. Preparar un servidor no es tan difícil como suena, y le dará mayor confiabilidad y probablemente también mejor velocidad.

Puede encontrar información detallada sobre las opciones del servidor de Subversion, y de cómo elegir la mejor arquitectura para su situación en el libro de Subversion: [Configuración del Servidor](http://svnbook.red-bean.com/en/1.8/svn.serverconfig.html) [<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. Guía de uso diario

Este documento describe el uso diario del cliente TortoiseSVN. *No* es una introducción a los sistemas de control de versiones, y *no* es una introducción a Subversion (SVN). Es más como un lugar donde puede venir cuando sepa qué quiere hacer, pero no recuerde exactamente cómo hacerlo.

Si necesita una introducción al control de versiones con Subversion, le recomendamos que se lea el fantástico libro: *Control de versiones con Subversion* [<http://svnbook.red-bean.com/>].

Este documento es también un trabajo en progreso, igual que lo son TortoiseSVN y Subversion. Si encuentra algún error, por favor háganoslo saber en la lista de correo para que podamos actualizar la documentación. Algunas de las capturas de pantalla en la Guía de Uso Diario (GUD) puede que no reflejen el estado actual del software. Le rogamos nos disculpe. Trabajamos en TortoiseSVN en nuestro tiempo libre.

Para aprovechar al máximo la Guía de uso diario:

- Debe tener ya instalado TortoiseSVN.
- Debe estar familiarizado con los sistemas de control de versiones.
- Debe conocer las bases de Subversion.
- Debe haber preparado un servidor y/o tener acceso a un repositorio de Subversion.

## 4.1. Características generales

Esta sección describe algunas de las características de TortoiseSVN que aplican a todo el manual. Ten en cuenta que muchas de estas características sólo aparecen dentro de una copia de trabajo de Subversion.

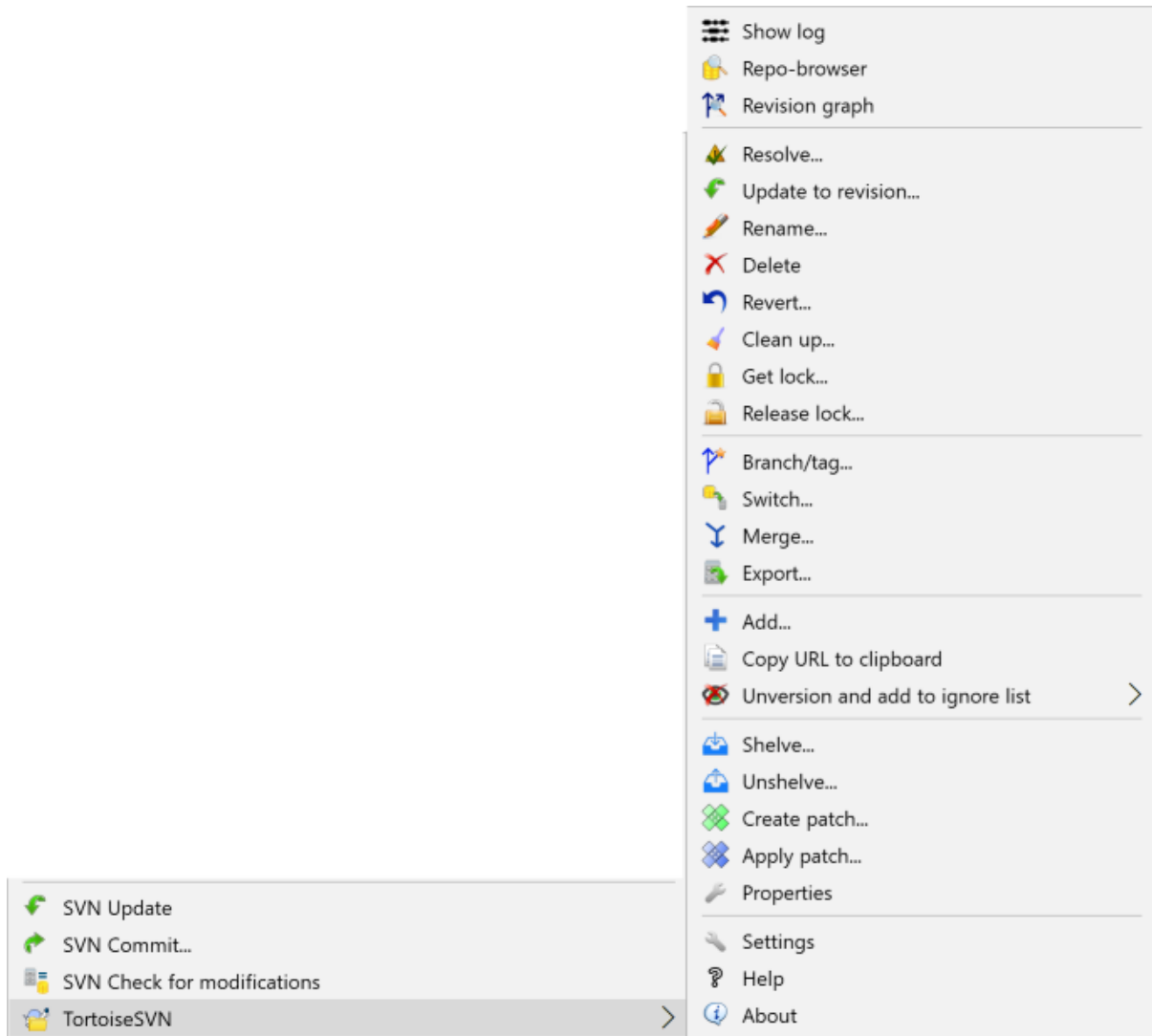
### 4.1.1. Íconos superpuestos



**Figura 4.1. Explorador mostrando íconos superpuestos**

Una de las funciones más visibles de TortoiseSVN son los íconos superpuestos que aparecen en los archivos de su copia de trabajo. Estos le muestran de un vistazo qué archivos han sido modificados. Lea [Sección 4.7.1, “Íconos superpuestos”](#) para averiguar qué representa cada ícono superpuesto.

### 4.1.2. Menús contextuales



**Figura 4.2. Menú contextual para un directorio bajo el control de versiones**

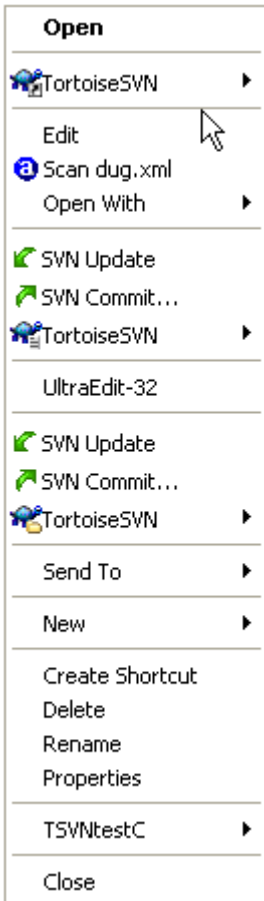
Todos los comandos de TortoiseSVN se invocan desde el menú contextual del explorador de Windows. La mayoría se ven directamente, cuando hace click con el botón derecho en un archivo o una carpeta. Los comandos disponibles dependen de si el archivo o la carpeta o su carpeta padre está bajo el control de versiones o no. También puede ver el menú de TortoiseSVN como parte del menú archivo del explorador.



### Sugerencia

Algunos comandos que se utilizan muy raramente sólo están disponibles en el menú contextual extendido. Para mostrar el menú contextual extendido, mantenga pulsada la tecla **Mayús** mientras hace click con el botón derecho.

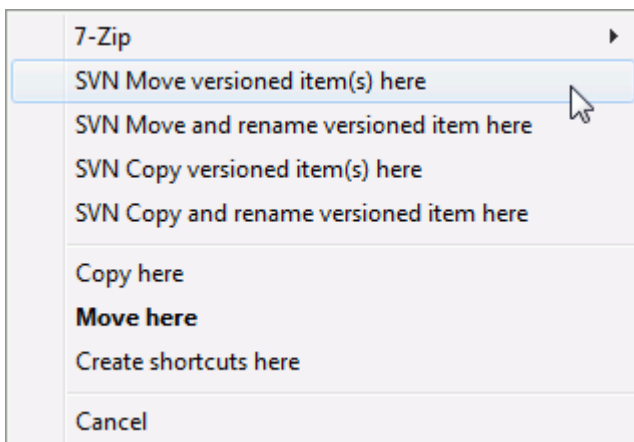
En algunos casos puede ver varias entradas de TortoiseSVN. ¡Esto no es un error!



**Figura 4.3. Menú archivo del explorador para un acceso directo en una carpeta versionada**

Este ejemplo es para un acceso directo sin versionar dentro de una carpeta versionada, y en el menú de archivo del Explorador hay *tres* entradas para TortoiseSVN. Una es para la carpeta, otra para el acceso directo en sí mismo, y otra para el objeto al que apunta el acceso directo. Para ayudarle a distinguir entre ellos, los íconos tienen un indicador en la esquina inferior derecha para mostrarle que la entrada del menú es para un archivo, una carpeta, un acceso directo o para múltiples ítems seleccionados.

### 4.1.3. Arrastrar y soltar



**Figura 4.4. Menú de arrastre con el botón derecho para un directorio bajo el control de versiones**

Otros comandos están disponibles como manejadores de arrastre, cuando arrastra con el botón derecho archivos o carpetas a un nuevo destino dentro de copias de trabajo, o cuando arrastra con el botón derecho un archivo o una carpeta no versionados a un directorio que está bajo el control de versiones.

#### 4.1.4. Atajos comunes

Algunas operaciones comunes tienen atajos de Windows bien conocidos, pero no aparecen en botones o en los menús. Si no puede averiguar cómo hacer algo obvio, como refrescar una vista, mire aquí.

F1

La ayuda, por supuesto.

F5

Refresca la vista actual. Este es quizás el comando de una tecla más útil. Por ejemplo... en el Explorador esto refresca los íconos superpuestos en su copia de trabajo. En el diálogo de confirmación volverá a verificar la copia de trabajo para ver qué puede necesitar ser confirmado. En el diálogo de Mostrar Registro contactará con el repositorio de nuevo buscando los cambios más recientes.

Ctrl-A

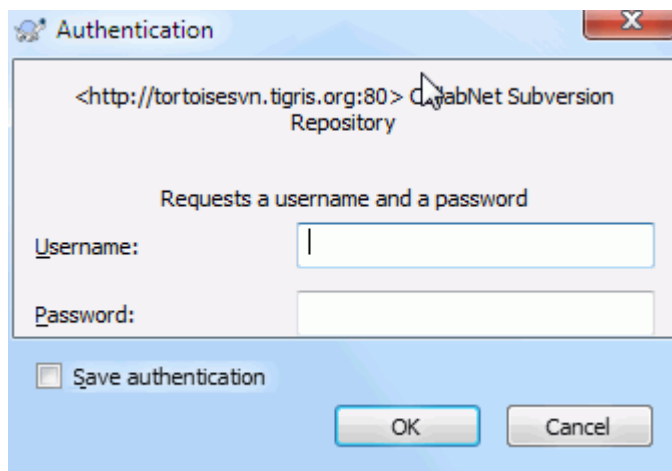
Selecciona todo. Esto puede ser útil si obtiene un mensaje de error y quiere copiar y pegarlo en un email. Utilice Ctrl-A para seleccionar el mensaje de error y luego...

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. Autenticación

Si el repositorio al que intenta acceder está protegido por contraseña, aparecerá un diálogo de autenticación.



**Figura 4.5. Diálogo de autenticación**

Introduzca su usuario y contraseña. La casilla le permite que TortoiseSVN almacene las credenciales en el directorio por defecto de Subversion: %APPDATA%\Subversion\auth, en tres subdirectorios:

- `svn.simple` contiene las credenciales para la autenticación básica (usuario/contraseña). Note que las contraseñas están guardadas usando la WinCrypt API, no como texto sin formato.
- `svn.ssl.server` contiene los certificados SSL de servidor.
- `svn.username` contiene las credenciales para autenticación sólo por usuario (sin necesidad de contraseña).

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 [Sección 4.31.6, “Datos de configuración almacenados”](#).

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 más información sobre cómo preparar su servidor para la autenticación y el control de acceso, vea [Sección 3.5, “Accediendo al repositorio”](#).

### 4.1.6. Maximizando ventanas

Muchos de los diálogos de TortoiseSVN tienen montones de información que mostrar, pero a menudo es más útil maximizar sólo la altura o sólo la anchura, mejor que maximizar para ocupar toda la pantalla. Como ayuda existen atajos para esto en el botón Maximizar. Utilice el botón central del ratón para maximizar verticalmente, y el botón derecho del ratón para maximizar horizontalmente.

## 4.2. Importando datos en un repositorio

### 4.2.1. Importar

Si está importando en un repositorio que ya tiene algunos proyectos, entonces la estructura del repositorio ya estará decidida. Si está importando datos a un nuevo repositorio entonces merece la pena tomar el tiempo para pensar en cómo debería organizarse. Lea [Sección 3.1.5, “Organización del repositorio”](#) para más información.

Esta sección describe el comando Importar de Subversion, que se diseñó para importar una estructura de carpetas en el repositorio de un sólo paso. Aunque realiza su cometido, tiene muchos inconvenientes:

- No hay forma de seleccionar los archivos y carpetas a incluir, salvo si utiliza la configuración del patrón global de ignorar.
- La carpeta importada no se convierte en una copia de trabajo. Tiene que hacer una obtención para copiar los archivos de nuevo desde el servidor.
- Es fácil importar en un nivel de carpetas erróneo en el repositorio.

Por estos motivos recomendamos no utilizar el comando Importar y emplear mejor el método en dos pasos descrito en [Sección 4.2.2, “Importar en el sitio”](#), a menos que esté realizando el paso inicial de crear la estructura de carpetas /trunk /tags /branches en el repositorio. Pero ya que estamos aquí, veamos como funciona la importación básica ...

Antes de importar su proyecto en un repositorio debería:

1. Quitar todos los archivos que no se necesitan para construir el proyecto (archivos temporales, archivos que se generan por un compilador como los \*.obj, binarios compilados, ...)
2. Organizar los archivos en carpetas y subcarpetas. Aunque es posible renombrar/mover los archivos más tarde, ¡es muy recomendable que tenga la estructura del proyecto antes de importarlo!

Ahora seleccione la carpeta superior de la estructura de directorios del proyecto en el explorador de Windows, y haga click con el botón derecho para abrir el menú contextual. Seleccione el comando TortoiseSVN → Importar... y aparecerá un cuadro de diálogo:



**Figura 4.6. El 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.

El mensaje de importación se utiliza como un mensaje de registro.

Por defecto, los archivos y carpetas que concuerden con los patrones globales de ignorar *no* se importan. Para cambiar este comportamiento, puede utilizar la casilla **Incluir archivos ignorados**. Lea [Sección 4.31.1, “Configuración general”](#) para más información sobre cómo establecer un patrón global de ignorar.

Tan pronto como presione **Aceptar**, TortoiseSVN importa el árbol completo de directorios, incluyendo todos los archivos, en el repositorio. El proyecto ahora está almacenado en el repositorio bajo el control de versiones. Por favor tenga en cuenta que la carpeta que ha importado *¡NO* está bajo el control de versiones! Para obtener una *copia de trabajo* bajo el control de versiones necesita Obtener la versión que acaba de importar. O siga leyendo para averiguar cómo importar una carpeta en el sitio.

## 4.2.2. Importar en el sitio

Asumiendo que ya tiene un repositorio, y que quiere añadir una nueva estructura de carpetas e él, sólo tiene que seguir estos pasos:

1. Utilice el navegador de repositorios para crear directamente una carpeta de proyecto en el repositorio. Si está utilizando alguna de las estructuras estándar probablemente querrá crearla como una subcarpeta de trunk en

vez de la raíz del repositorio. El navegador de repositorios muestra la estructura del repositorio igual que el Explorador de archivos de Windows, de forma que puedes ver como está organizado.

2. Obtenga la nueva carpeta encima de la carpeta que quiere importar. Recibirá un aviso indicando que la carpeta local no está vacía. Ignore el aviso. Ahora tendrá una carpeta versionada con contenido no versionado.
3. Utilice TortoiseSVN → Añadir... en esta carpeta versionada para añadir parte o todo su contenido. Puede añadir y eliminar archivos, establecer las propiedades `svn:ignore` en las carpetas y hacer cualquier otro cambio que necesite.
4. Confirme la carpeta de más alto nivel, y ya tiene un nuevo árbol versionado, y una copia de trabajo local, creada desde su carpeta existente.

### 4.2.3. Archivos especiales

A veces necesitará tener un archivo bajo control de versiones que contenga datos específicos del usuario. Esto significa que tiene un archivo que cada desarrollador/usuario necesita modificar para que se ajuste a su configuración local. Pero versionar ese archivo es difícil, porque cada usuario haría confirmaciones de sus cambios cada vez en el repositorio.

En estos casos le sugerimos que utilice archivos *plantilla*. Cree un archivo que contenga todos los datos que sus desarrolladores puedan necesitar, añádale al control de versiones y haga que sus desarrolladores lo obtengan. Luego, cada desarrollador tendrá que *hacer una copia* de ese archivo y renombrar esa copia. Después de eso, modificar la copia no vuelve a ser un problema.

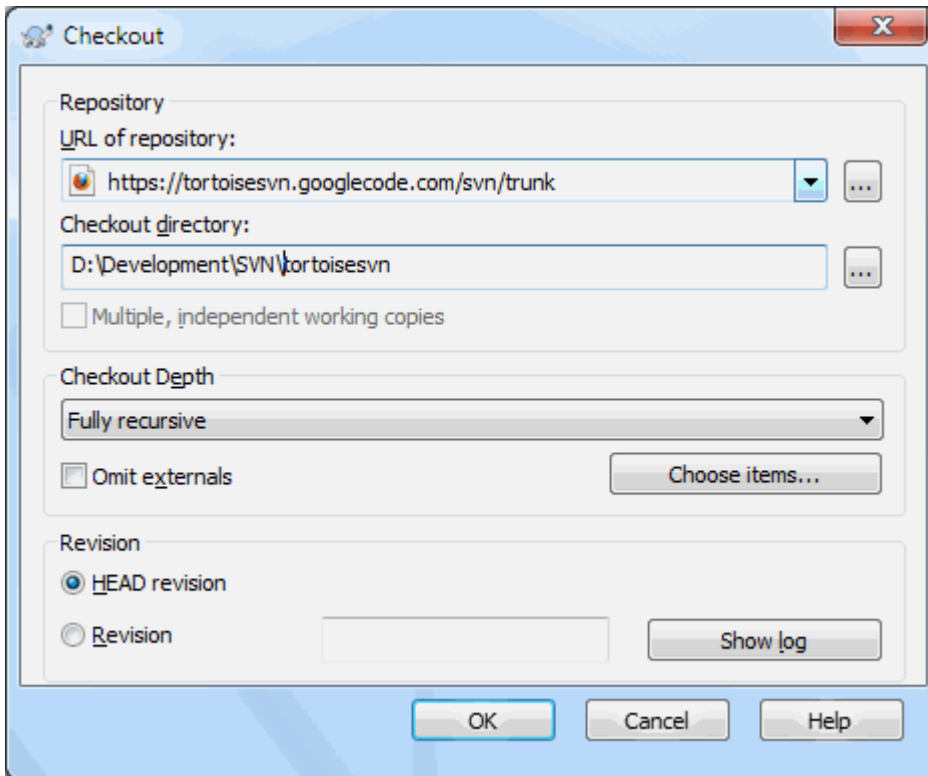
As an example, you can have a look at TortoiseSVN's build script. It calls a file named `default.build.user` which doesn't exist in the repository. Only the file `default.build.user.tmpl`. `default.build.user.tmpl` is the template file which every developer has to create a copy from and rename that file 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. Obteniendo una copia de trabajo

Para tener una copia de trabajo necesita *obtener* una de un repositorio.

Seleccione un directorio en el explorador de Windows donde quiera poner su copia de trabajo. Haga click con el botón derecho para mostrar el menú contextual y seleccione el comando TortoiseSVN → Obtener..., que mostrará el siguiente cuadro de diálogo:



**Figura 4.7. El diálogo Obtener**

Si introduce un nombre de carpeta que no aún no exista, se creará un directorio con ese nombre.



### Importante

El la configuración por default, la opción salida del menú (checkout) no se encuentra en el submenú TortoiseSVN, pero se muestra en el menú superior del explorador. Los comandos de TortoiseSVN que no están en el submenú tienen SVN al principio: SVN Checkout...

#### 4.3.1. Profundidad de obtención

Puede elegir la *profundidad* que desea para la obtención, lo que le permite especificar la profundidad de la recursión en las carpetas hijas. Si sólo desea unas pocas secciones de un árbol grande, puede obtener sólo la carpeta de más alto nivel, y luego actualizar las carpetas seleccionadas de forma recursiva.

##### Totalmente recursivo

Obtener el árbol entero, incluyendo todas las carpetas hijas y subcarpetas.

##### Hijos inmediatos, incluyendo carpetas

Obtener el directorio especificado, incluyendo todos los archivo y carpetas hijas, pero no rellenar las carpetas hijas.

##### Sólo los archivos hijos

Obtener la carpeta especificada, incluyendo todos los archivos pero no obtener ninguna carpeta hija.

##### Sólo este ítem

Obtener sólo el directorio. No rellenarlo con archivos ni carpetas hijas.

##### Copia de trabajo

Retiene la profundidad especificada en la copia de trabajo. Esta opción no se utiliza en el diálogo obtener, pero es el valor por defecto para todos los demás diálogos que tengan opción de profundidad.

#### Excluir

Utilizado para reducir la profundidad de una copia de trabajo después que una carpeta haya sido rellenada. Esta opción sólo está disponible en el diálogo Actualizar a la revisión.

Para seleccionar fácilmente los elementos que quiere obtener y forzar a la copia de trabajo a mantener sólo estos elementos, haga clic en el botón **Seleccione los elementos....** Éste abrirá un cuadro de diálogo donde podrá seleccionar qué elementos quiere en su copia de trabajo y desmarcar los que no quiera. La copia de trabajo resultante se conoce como obtención dispersa. Las actualizaciones de este tipo de copias de trabajo no recuperarán los archivos ni carpetas que falten y sólo actualizará los elementos que ya estaban en la copia de trabajo.

Si obtiene una copia de trabajo dispersa (es decir, eligiendo otra cosa que no sea `totalmente recursivo` para la profundidad de obtención), puede posteriormente añadir o suprimir subcarpetas con facilidad mediante cualquiera de los métodos siguientes.

#### 4.3.1.1. Actualización dispersa utilizando Actualización a la Revisión

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. actualización rala usando el navegador de repositorio.

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. Actualización dispersa usando BuscarModificaciones

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 característica es muy útil si sólo desea obtener partes de un árbol más grande, pero desea la facilidad de poder actualizar una única copia de trabajo. Suponga que tiene un gran árbol que tiene subcarpetas desde `Proyecto01` a `Proyecto99`, y sólo desea obtener las carpetas `Proyecto03`, `Proyecto25` y `Proyecto76/SubProy`. Utilice estos pasos:

1. Obtenga la carpeta padre con profundidad "Sólo este ítem". Ahora tiene una carpeta de nivel superior vacía.
2. Seleccione la nueva carpeta y utilice TortoiseSVN → Navegador de repositorios para ver el contenido del repositorio.
3. Haga click con el botón derecho sobre `Proyecto03` y Menú contextual → Actualizar ítem a la revisión.... Mantenga las configuraciones por defecto y haga click en Aceptar. Ahora tendrá una carpeta totalmente poblada.

Repita el mismo proceso para `Proyecto25`.

4. Navegue a `Proyecto76/SubProy` y haga lo mismo. Esta vez notará que la carpeta `Proyecto76` no tiene otro contenido que `SubProy`, que a su vez está totalmente poblado. Subversion ha creado las carpetas intermedias pero sin rellenarlas.



### Cambiando la profundidad de la copia de trabajo

Una vez que ha obtenido una copia de trabajo a una profundidad en particular, puede cambiar dicha profundidad más tarde para obtener más o menos contenido utilizando Menú contextual

→ Actualizar ítem a la revisión.... En ese diálogo, asegúrese de seleccionar la opción Hacer profundidad pegajosa.



## Utilizando un servidor antiguo

Los servidores anteriores a la versión 1.5 no entienden la petición de profundidad de copia de trabajo, por lo que no siempre pueden manejar las peticiones de forma eficiente. El comando funcionará, pero un servidor antiguo puede que envíe todos los datos, dejando que sea el cliente quien filtre lo que no necesite, lo que puede significar un montón de tráfico de red. Si es posible, debería actualizar su servidor al menos a la versión 1.5.

Si el proyecto contiene referencias a proyectos externos que *no* desea que se obtengan al mismo tiempo, utilice la casilla Omitir externos.



## Importante

Si está marcado Omitir externos, o si desea incrementar el valor de la profundidad, deberá realizar actualizaciones a su copia de trabajo utilizando TortoiseSVN → Actualizar a la Revisión... en vez de TortoiseSVN → Actualizar. La actualización estándar incluirá todos los externos y mantendrá la profundidad actual.

Le recomendamos que obtenga únicamente la parte `trunk` del árbol de directorios, o desde más abajo. Si especifica la ruta padre del árbol de directorios en la URL, al final puede acabar con un disco duro lleno ¡porque obtendrá una copia del árbol completo del repositorio, incluyendo cada rama y etiqueta de su proyecto!



## Exportando

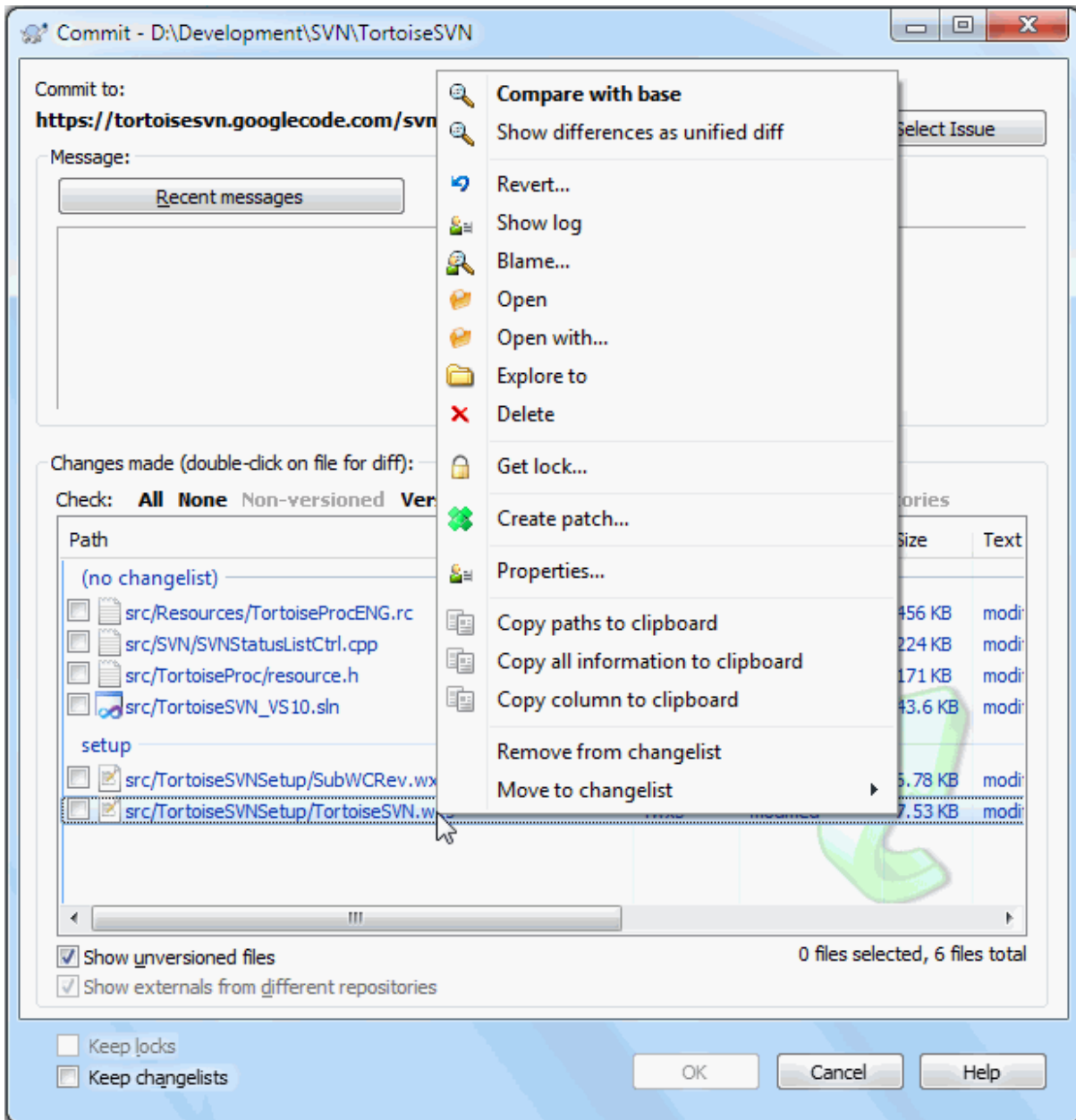
A veces puede querer crear una copia local sin ninguno de esos directorios `.svn`, por ejemplo para crear un archivo comprimido de sus fuentes. Lea [Sección 4.27, “Exportando una copia de trabajo de Subversion”](#) para averiguar cómo hacerlo.

## 4.4. Confirmando sus cambios en el repositorio

Enviar los cambios que ha hecho al repositorio se conoce como *confirmar* los cambios. Pero antes de confirmar tiene que estar seguro que su copia de trabajo está actualizada. Puede o bien ejecutar TortoiseSVN → Actualizar directamente, o bien ejecutar TortoiseSVN → Comprobar Modificaciones primero, para ver qué se ha cambiado localmente o en el servidor.

### 4.4.1. El diálogo de Confirmación

Si su copia de trabajo está actualizada y no hay conflictos, ya está preparado para confirmar sus cambios. Seleccione los archivos y/o carpetas que desee confirmar y seleccione TortoiseSVN → Confirmar....



**Figura 4.8. El diálogo de Confirmación**

El diálogo de confirmación le mostrará todos los archivos cambiados, incluso los archivos añadidos, borrados o no versionados. Si no desea que un archivo cambiado se confirme, simplemente desmarque ese archivo. Si desea incluir un archivo no versionado, márkelo para añadirlo a la confirmación.

Para rápidamente marcar o desmarcar tipos de ficheros como todos los ficheros versionados o todos los ficheros modificados, pulsa el enlace justo encima de la lista de los elementos mostrados

Para información sobre el colores y leyendas de los elementos de acuerdo a su status, por favor mira [Sección 4.7.3, “Estado local y remoto”](#).

Los ítems que han sido cambiados a una ruta de repositorio diferente también se indican utilizando un marcador (s). Puede haber cambiado algo mientras trabaja en una rama y habérsele olvidado volver a cambiarlo al tronco. ¡Este es su signo de advertencia!



## ¿Confirmar archivos o carpetas?

Cuando confirma archivos, el diálogo de confirmación sólo le enseña los archivos que ha seleccionado. Cuando confirma una carpeta el diálogo de confirmación seleccionará los archivos que han cambiado de forma automática. Si se olvidó un archivo nuevo que haya creado, al confirmar la carpeta lo encontrará. Confirmar una carpeta *no* significa que todos los archivos se marquen como cambiados; sólo le hace la vida más fácil haciendo más trabajo por usted.



## Muchos archivos sin versionar en el diálogo de confirmar

Si cree que el diálogo de confirmación de TSVN le muestra demasiados archivos no versionados (por ejemplo, archivos generados por el compilador o copias de seguridad hechas por su editor), hay varias formas de manejar esta situación. Puede:

- añadir el archivo (o una extensión con máscara) a la lista de archivos a ignorar en la página de configuración. Esto afectará a todas las copias de trabajo que tenga.
- añadir el archivo a la lista de `svn:ignore` utilizando TortoiseSVN → Añadir a la lista de ignorados. Esto únicamente afectará al directorio en el que establezca la propiedad `svn:ignore`. Puede cambiar la propiedad `svn:ignore` de un directorio utilizando el Diálogo de Propiedades SVN.
- 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.

Para más información, lea [Sección 4.14, “Ignorando archivos y directorios”](#).

Haciendo doble click en cualquier archivo modificado en el diálogo de confirmación, se lanzará la herramienta externa de diferenciar para mostrarle sus cambios. El menú contextual le proporciona más opciones, como se ve en la captura de pantalla. También puede arrastrar archivos desde aquí a otra aplicación, como un editor de textos o un IDE.

Puede seleccionar o deseleccionar ítems haciendo click en la casilla a la izquierda del ítem. Para los directorios puede utilizar **Mayúsculas**-Seleccionar para ejecutar la acción de forma recursiva.

Las columnas que se muestran en el panel inferior son personalizables. Si hace click con el botón derecho en cualquier cabecera de columna verá un menú contextual que le permite seleccionar qué columnas se muestran. También puede cambiar el ancho de la columna utilizando el manejador de arrastre que aparece cuando mueve el cursor sobre el límite de una columna. Estas personalizaciones se mantienen, por lo que verá los mismos encabezados la siguiente vez.

Por defecto cuando confirma los cambios, cualquier bloqueo que tenga en los archivos se libera automáticamente cuando la confirmación tiene éxito. Si desea mantener esos bloqueos, asegúrese que la casilla **Mantener bloqueos** está marcada. El estado por defecto de esta casilla se toma de la opción `no_unlock` del archivo de configuración de Subversion. Lea [Sección 4.31.1, “Configuración general”](#) para más información sobre cómo editar el archivo de configuración de 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: [Sección 4.31.2, “Configuración del gráfico de revisión”](#)



## Arrastrar y soltar

Puede arrastrar archivos hasta el diálogo de confirmación desde cualquier otro lugar, siempre y cuando las copias de trabajo sean del mismo repositorio. Por ejemplo, puede tener una copia de trabajo enorme con diferentes ventanas del explorador abiertas en carpetas distantes de la jerarquía. Si quiere evitar confirmar desde la carpeta más alta (lo que implica una lenta operación de búsqueda de cambios) puede abrir el diálogo de confirmar para una carpeta y arrastrar desde las otras ventanas para incluir archivos dentro de la misma confirmación atómica.

Puede arrastrar archivos no versionados que residan dentro de una copia de trabajo al diálogo de confirmación, y automáticamente serán SVN añadidos.

Arrastrar archivos desde la lista al pie del diálogo de confirmación al cuadro de edición del registro de mensajes insertará las rutas como texto plano en ese cuadro de edición. Esto es útil si desea escribir mensajes de registro de confirmación que incluyan las rutas que están siendo afectadas por la confirmación.



## Reparando renombrados externos

A veces los archivos se renombran fuera de Subversion, y se muestran en la lista de archivos como un archivo faltante y un archivo no versionado. Para evitar perder la historia necesita notificar a Subversion su conexión. Simplemente seleccione tanto el nombre antiguo (faltante) como el nombre nuevo (sin versionar) y utilice **Menú contextual** → **Reparar movimiento** para emparejar los dos archivos como un renombrado.



## Reparando copias externas

Si hizo una copia de un archivo pero olvidó usar el comando de Subversion para hacerlo, puede reparar esa copia para que el nuevo archivo no pierda su historial. Simplemente seleccione tanto el nombre antiguo (normal o modificado) como el nuevo (no versionado) y utilice **Menú contextual** → **Reparar Copia** para emparejar los dos archivos como una copia.

### 4.4.2. Listas de cambios

El diálogo de confirmación da soporte a las listas de cambios de Subversion para ayudar a agrupar archivos relacionados. Averigue más sobre esta característica en [Sección 4.8, “Listas de cambios”](#).

### 4.4.3. Confirmar ficheros parcialmente

Quizá quiera confirmar únicamente parte de las modificaciones que ha hecho en un fichero. Esta situación normalmente sucede cuando se trabaja en algo y aparece la necesidad de una corrección urgente, y esa corrección resulta estar en el mismo fichero en el que se está trabajando.

Haga click derecho en el fichero y elija **Menú Contextual** → **Restaurar después de confirmar**. Esto creará una copia del archivo tal cual está. En ese momento se puede editar el fichero y deshacer todos los cambios que no desee confirmar. Tras guardar los cambios ya puede confirmar el fichero.



## Usando 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.

Cuando la confirmación se haya hecho, la copia del fichero se restaurará automáticamente, y su fichero contendrá de nuevo las modificaciones que no fueron confirmadas.

### 4.4.4. Excluyendo ítems de la lista de confirmación

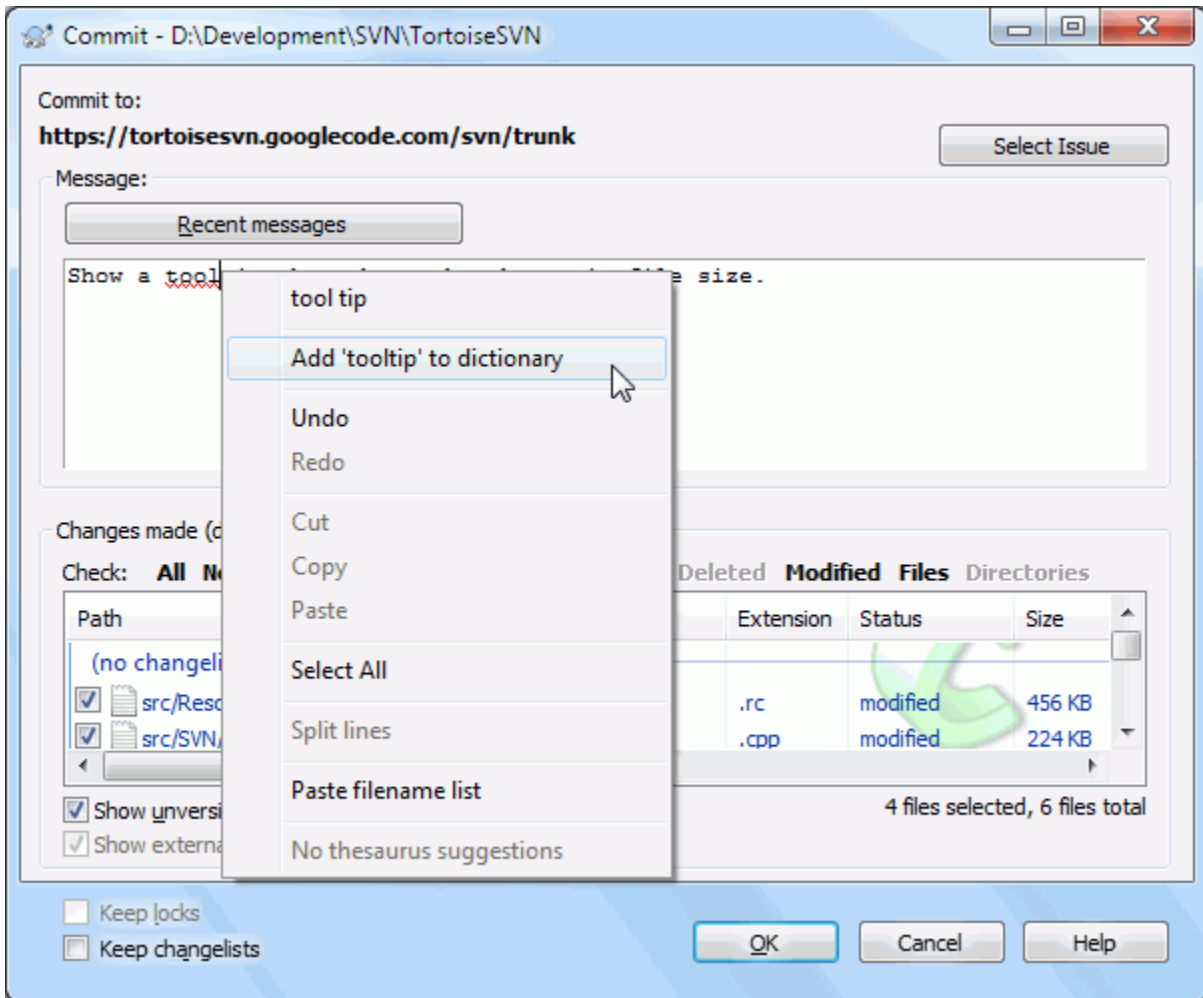
A veces tiene archivos versionados que cambian con frecuencia pero que realmente no desea confirmar. En ocasiones esto indica un fallo en su sistema de compilación - ¿por qué están esos archivos versionados? ¿debería utilizar archivos de plantilla? Pero ocasionalmente es inevitable. Una razón clásica es que su IDE cambie una fecha en el archivo de proyecto cada vez que lo compile. El archivo de proyecto debe estar versionado ya que contiene todas las configuraciones de la compilación, pero no necesita confirmarse sólo porque la fecha haya cambiado.

Para ayudarle en casos tan extraños como estos, hemos reservado una lista de cambios llamada `ignore-on-commit`. Cualquier archivo añadido a esta lista de cambios se desmarcará automáticamente en el diálogo de confirmación. Aún puede confirmar los cambios, pero tendrá que seleccionarlo manualmente en el diálogo de confirmación.

### 4.4.5. Mensajes de registro de confirmación

Asegúrese de introducir un mensaje de registro que describa los cambios que está confirmando. Esto le ayudará a saber qué ocurrió y cuando según navegue por los mensajes de registro del proyecto en el futuro. El mensaje puede ser tan extenso o escueto como desee; muchos proyectos tienen directrices sobre qué debe incluirse en ellos, el idioma que debe utilizarse, y a veces incluso un formato estricto.

Puede aplicar formatos sencillos en sus mensajes de registro utilizando una convención similar a la usada en los emails. Para aplicar un estilo a un texto, utilice `*texto*` para la negrita, `_texto_` para el subrayado, y `^texto^` para la cursiva.



**Figura 4.9. El corrector ortográfico del diálogo de Confirmación**

TortoiseSVN incluye un corrector ortográfico para ayudarle a escribir sus mensajes de registro correctamente. Este corrector señalará cualquier palabra mal escrita. Utilice el menú contextual para acceder a las correcciones sugeridas. Por supuesto, el corrector no conoce *todos* los términos técnicos que utiliza, así que a veces palabras bien escritas aparecerán como errores. Pero no se preocupe. Puede simplemente añadirlas a su diccionario personal utilizando el menú contextual.

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/>.

Como crear la expresión regular correctamente puede ser un poco laborioso, para ayudarle a crear una expresión adecuada existe un diálogo de testeo que le permite introducir una expresión y posteriormente introducir nombres de archivo contra los que la expresión es comprobada. Para iniciar el diálogo de testeo, introduzca el comando `TortoiseProc.exe /command:autotexttest` en el símbolo de sistema.

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: \\.

Puede reutilizar mensajes de registro que haya introducido anteriormente. Tan sólo debe pulsar en **Mensajes recientes** para ver una lista de los últimos mensajes que ha introducido para esta copia de trabajo. El número de mensajes almacenados se puede personalizar en el diálogo de configuración de TortoiseSVN.

Puede limpiar todos los mensajes de confirmación almacenados desde la página **Datos Almacenados** de la configuración de TortoiseSVN, o puede eliminar mensajes individuales dentro del diálogo **Mensajes recientes** utilizando la tecla **Supr**.

Si desea incluir las rutas marcadas en su mensaje de registro, puede utilizar el comando **Menú contextual** → **Pegar lista de nombres de archivos** en el control de edición.

Otra forma de insertar rutas en el mensaje de registro es simplemente arrastrar los archivos desde la lista de archivos al control de edición.



### Propiedades especiales de carpetas

Hay diversas propiedades especiales de carpeta que pueden usarse para darle mayor control sobre el formato de los mensajes de registro de las confirmaciones y el idioma que utiliza el módulo del corrector ortográfico. Para más información, lea [Sección 4.18, “Configuración del proyecto”](#).

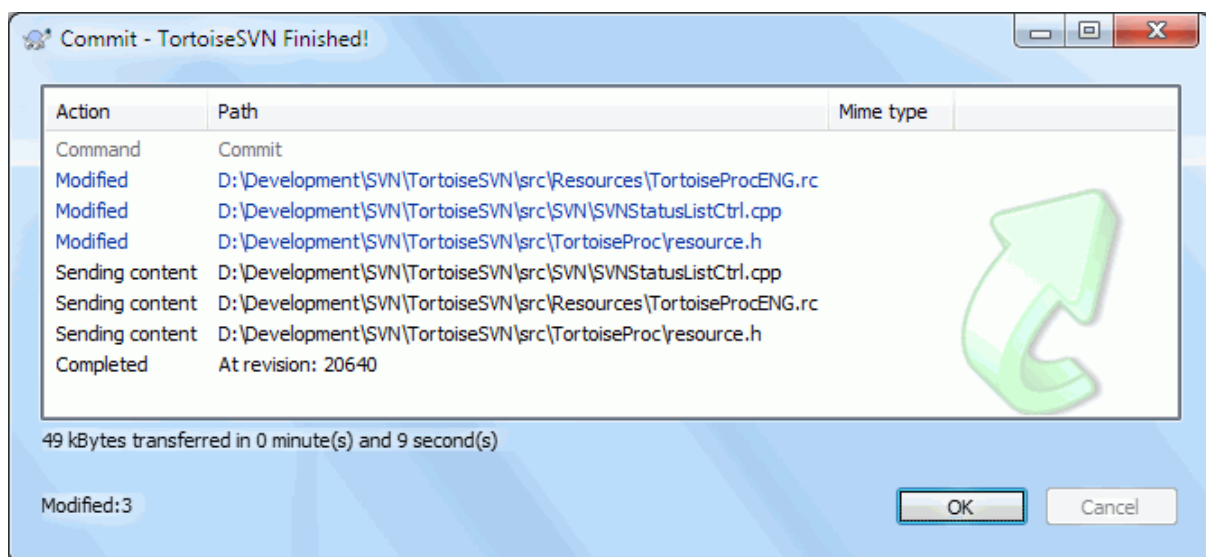


### Integración con herramientas de control de errores

Si ha activado el sistema de control de errores, puede poner una o más incidencias en el cuadro de texto Bug-ID / N<sup>o</sup>-Incid.:. Si quiere introducir múltiples incidencias, sepárelas por comas. Alternativamente, si está utilizando el soporte de control de errores basado en expresiones regulares, simplemente añada las referencias a sus incidencias como parte del mensaje de registro. Si desea saber más, lea [Sección 4.29, “Integración con sistemas de control de errores / seguimiento de incidencias”](#).

## 4.4.6. Progreso de confirmación

Tras pulsar **Aceptar** aparece un diálogo mostrando el progreso de la confirmación.



**Figura 4.10. El diálogo Progreso mostrando el progreso de una confirmación**

El diálogo de progreso utiliza una codificación de colores para resaltar las diferentes acciones de confirmación:

Azul

Confirmando una modificación.

Púrpura

Confirmando un ítem añadido.

Rojo oscuro

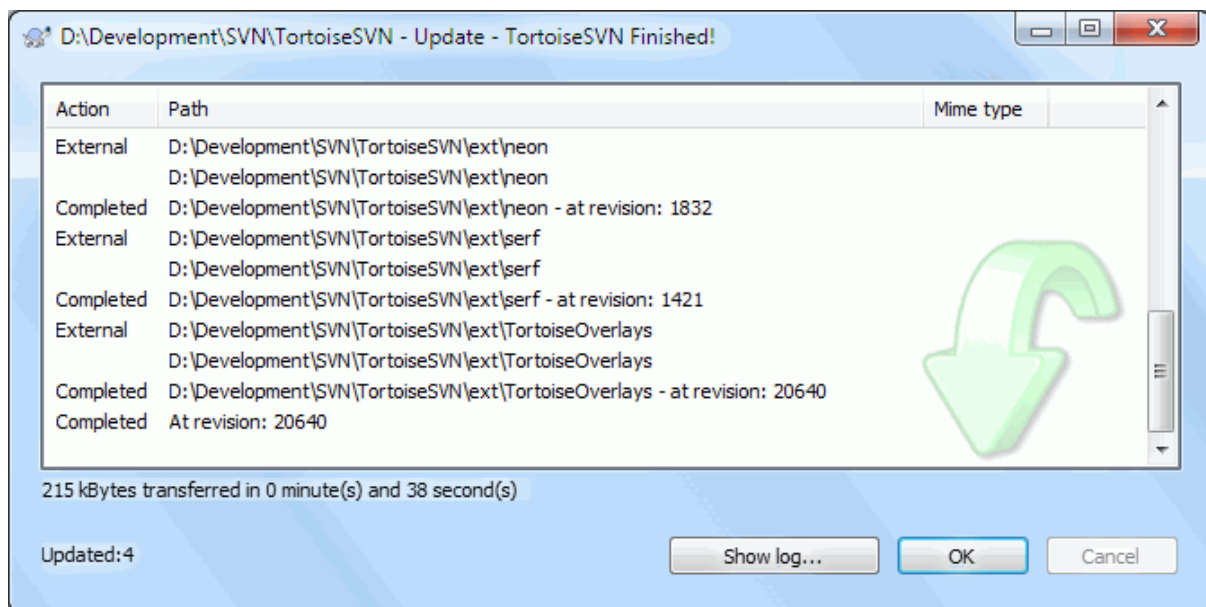
Confirmando un borrado o un reemplazo.

Negro

Todos los demás ítems.

Este es el esquema de colores por defecto, pero puede personalizar dichos colores utilizando el diálogo de configuración. Para más información, lea [Sección 4.31.1.5, “Configuración de colores de TortoiseSVN”](#).

## 4.5. Actualice su copia de trabajo con los cambios de otros



**Figura 4.11. Diálogo de progreso mostrando una actualización terminada**

Periódicamente, debería asegurarse que los cambios que hacen los demás se incorporen en su copia de trabajo local. El proceso de incorporar los cambios desde el servidor a su copia de trabajo local se conoce como *actualización*. La actualización puede hacerse en archivos sueltos, en un conjunto de archivos, o recursivamente en jerarquías completas de directorios. Para actualizar, seleccione los archivos y/o directorios que desee, haga click con el botón derecho y seleccione TortoiseSVN → Actualizar en el menú contextual del explorador. Aparecerá una ventana con el progreso de la actualización según se ejecuta. Los cambios que los demás hayan hecho se fusionarán con sus archivos, manteniendo cualquier cambio que haya hecho en los mismos archivos. El repositorio *no* se ve afectado por una actualización.

El diálogo de progreso utiliza un código de colores para resaltar diferentes acciones de actualización:

Púrpura

Nuevo ítem añadido a su copia de trabajo

Rojo oscuro

Ítem redundante borrado de su copia de trabajo, o ítem faltante reemplazado en su copia de trabajo.

Verde

Cambios del repositorio que se han fusionado satisfactoriamente con sus cambios locales.

### Rojo brillante

Cambios del repositorio fusionados con sus cambios locales, pero que han dado lugar a conflictos que debe resolver.

### Negro

Items sin cambios en su copia de trabajo actualizados con una versión más nueva desde el repositorio.

Este es el esquema de colores por defecto, pero puede personalizar dichos colores utilizando el diálogo de configuración. Para más información, lea [Sección 4.31.1.5, “Configuración de colores de TortoiseSVN”](#).

Si obtiene algún *conflicto* durante una actualización (esto puede suceder si los demás han cambiado las mismas líneas del mismo archivo a la vez que usted y esos cambios no concuerdan), el diálogo muestra esos conflictos en rojo. Puede hacer doble click en esas líneas para iniciar la herramienta de fusión externa para resolver los conflictos.

Cuando se completa la actualización, el diálogo de progreso le muestra un resumen con el número de ítems actualizados, añadidos, eliminados, en conflicto, etc. bajo la lista de archivos. Esta información de resumen puede copiarse al portapapeles utilizando Ctrl+C.

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

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

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

También puedes escoger si deseas ignorar alguno de los proyectos externos en la actualización (en otras palabras, proyectos referenciados usando `svn:externals`).



### Atención

Si actualiza un archivo o una carpeta a una revisión en concreto, no debería hacer cambios en esos archivos. ¡Obtendrá mensajes `out of date` (desactualizado) cuando intente confirmarlos! Si desea deshacer los cambios de un archivo y empezar de nuevo desde una revisión anterior, puede revertir a una revisión previa desde el diálogo de registro de revisiones. Lea [Sección B.4, “Deshacer revisiones en el repositorio”](#) si desea más instrucciones, y métodos alternativos.

Actualizar a la revisión puede ser útil a veces para ver cómo estaba su proyecto en un momento anterior en su historia. Pero en general, actualizar archivos individuales a una revisión anterior no es una buena idea, ya que deja su copia de trabajo en un estado inconsistente. Si el archivo que está actualizando ha cambiado de nombre, incluso puede encontrar que ese archivo ha desaparecido de su copia de trabajo porque en esa revisión no había ningún archivo con ese nombre. También debe tener en cuenta que el ítem mostrará un ícono superpuesto normal verde, por lo que no se puede distinguir del resto de archivos que están actualizados.

Si desea simplemente una copia local de una versión antigua de un archivo, es mejor utilizar el comando Menú contextual → Guardar revisión en... desde el diálogo de registro para dicho archivo.



### Múltiples archivos/carpetas

Si selecciona múltiples archivos y carpetas en el explorador y luego selecciona Actualizar, todos esos archivos/carpetas se actualizan uno a uno. ¡TortoiseSVN se asegura que todos los archivos/

carpetas del mismo repositorio se actualicen exactamente a la misma revisión! Incluso si entre esas actualizaciones ocurrió alguna confirmación.

## 4.6. Resolviendo conflictos

De vez en cuando, obtendrá un *conflicto* cuando actualice/fusione sus archivos desde el repositorio o cuando cambie su copia de trabajo a una URL diferente. Hay dos tipos de conflictos:

conflictos de archivo

Un conflicto de archivo ocurre si dos (o más) desarrolladores han cambiado las mismas líneas de un archivo.

conflictos de árboles

Un conflicto de árbol ocurre cuando un desarrollador mueve/renombra/elimina un archivo o una carpeta, que otro desarrollador también ha movido/renombrado/borrado, o quizás lo haya modificado.

### 4.6.1. Conflictos de archivos

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:

nombre-del-archivo.ext.mine

Este es su archivo tal y como estaba en su copia de trabajo antes que actualizara su copia de trabajo - esto es, sin marcadores de conflicto. Este archivo tiene sus últimos cambios en él y nada más.

nombre-del-archivo.ext.rREV-ANTIGUA

Este es el archivo que era la revisión BASE antes que actualizara su copia de trabajo. Esto es, el archivo que obtuvo antes de empezar a hacer sus últimos cambios.

nombre-del-archivo.ext.rREV-NUEVA

Este es el archivo que su cliente de Subversion acaba de recibir desde el servidor del que actualizó su copia de trabajo. Este archivo corresponde a la revisión HEAD del repositorio.

Puede lanzar una herramienta de fusión / editor de conflictos externo con TortoiseSVN → Editar Conflictos o puede usar cualquier editor de texto para resolver el conflicto manualmente. Debe decidir cómo debe quedar finalmente el código, hacer los cambios necesarios y guardar el fichero. Usando una herramienta de fusión como TortoiseMerge o cualquier otra herramienta popular es normalmente la mejor opción ya que generalmente presentan los ficheros en una vista de 3 paneles y no tiene que preocuparse de las marcas de conflicto. Si usa un editor de texto entonces debe buscar las líneas que comiencen con el literal <<<<<<.

Después, ejecute el comando TortoiseSVN → Resolver y confirme sus modificaciones al repositorio. Tome nota que el comando Resolver realmente no resuelve el conflicto. Simplemente elimina los archivos `filename.ext.mine` y `filename.ext.r*`, dejándole confirmar sus cambios.

Si tiene conflictos con archivos binarios, Subversion no intentará mezclar dichos archivos por si mismo. El archivo local se mantendrá sin cambios (exactamente tal y como lo había cambiado usted) y obtendrá unos archivos `nombrefichero.ext.r*`. Si desea descartar sus cambios y quedarse con la versión del repositorio, utilice el comando Revertir. Si desea mantener su versión y sobrescribir la versión del repositorio, utilice el comando Resuelto y luego confirme su versión.

Puede utilizar el comando Resuelto para múltiples archivos si pulsa con el botón derecho en la carpeta padre y selecciona TortoiseSVN → Resuelto... Esto mostrará un diálogo con todos los archivos en conflicto dentro de esa carpeta, y le permitirá seleccionar cuáles marcar como resueltos.

#### 4.6.2. Conflictos de propiedad

Un conflicto de propiedades ocurre si dos (o más) desarrolladores han cambiado la misma propiedad. Como con los conflictos de archivos, resolver el conflicto solo puede ser hecho por los desarrolladores.

Si uno de los cambios debe sobrescribir al otro entonces seleccione la opción **Resolver usando la propiedad local** o **Resolver usando la propiedad remota**. Si los cambios deben ser fusionados entonces seleccione **Editar propiedad** manualmente, resuelva cuál debiera ser el valor de la propiedad y marque como resuelto.

#### 4.6.3. Conflictos de árbol

Un conflicto de árbol ocurre cuando un desarrollador mueve/renombra/elimina un archivo o una carpeta, que otro desarrollador también ha movido/renombrado/borrado, o quizás lo haya modificado. Hay diferentes situaciones que puede resultar en un conflicto de árbol, y todas ellas requiere pasos diferentes para resolver el conflicto.

Cuando se elimina un archivo de forma local en Subversion, el archivo también se elimina del sistema local de archivos, por lo que incluso si es parte de un conflicto de árbol no se puede mostrar un ícono superpuesto de conflicto y no puede hacer clic con el botón derecho sobre él para resolver el conflicto. En este caso, utilice el diálogo **Comprobar modificaciones** para acceder a la opción **Editar conflictos**.

TortoiseSVN puede ayudarle a encontrar el lugar correcto para fusionar los cambios, pero puede que necesite realizar un trabajo adicional para arreglar los conflictos. Recuerde que tras una actualización la BASE de trabajo siempre contendrá la revisión de cada ítem tal y como estaba en el repositorio en el momento de la actualización. Si revierte un cambio tras la actualización, se revierte a su estado del repositorio, no a como estaba cuando empezó a hacer sus propios cambios locales.

##### 4.6.3.1. Borrado local, llega un cambio en la actualización

1. El desarrollador A modifica `Foo.c` y lo confirma en el repositorio.
2. El desarrollador B al mismo tiempo ha movido `Foo.c` a `Bar.c` en su propia copia de trabajo, o simplemente ha borrado `Foo.c` o su carpeta padre.

Una actualización de la copia de trabajo del desarrollador B acaba con un conflicto de árbol:

- `Foo.c` ha sido borrado de la copia de trabajo, pero está marcado como un conflicto de árbol.
- Si el conflicto aparece después de un renombrado en vez de un borrado, entonces `Bar.c` está marcado como añadido, pero no contiene las modificaciones del desarrollador A.

El desarrollador B ahora tiene que elegir si mantiene los cambios realizados por el desarrollador A. En el caso de un renombrado, puede fusionar los cambios de `Foo.c` dentro del archivo renombrado `Bar.c`. Para simples borrados de archivos o directorios puede elegir quedarse con los cambios del ítem del desarrollador A y descartar el borrado. O, si marca el conflicto como resuelto sin hacer nada más, estará descartando los cambios del desarrollador 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. Edición local, entra un borrado en la actualización

1. El desarrollador A mueve `Foo.c` a `Bar.c` y lo confirma en el repositorio.
2. El desarrollador B modifica `Foo.c` en su copia de trabajo.



O en el caso de mover una carpeta...

1. El desarrollador A mueve la carpeta padre FooFolder a BarFolder y lo confirma en el repositorio.
2. El desarrollador B modifica Foo.c en su copia de trabajo.

Una actualización de la copia de trabajo de B acaba con un conflicto de árbol. Para un conflicto simple de archivo:

- Bar.c se añade a la copia de trabajo como un archivo normal.
- Foo.c se marca como añadido (con historia) y tiene un conflicto de árbol.

Para un conflicto de carpeta:

- BarFolder se añade a la copia de trabajo como una carpeta normal.
- FooFolder se marca como añadida (con historia) y tiene un conflicto de árbol.

Foo.c se marca como modificado.

El desarrollador B tiene ahora que decidir si desea continuar con la reorganización del desarrollador A y fusionar sus cambios en los archivos correspondientes de la nueva estructura, o simplemente revertir los cambios de A y mantener el archivo 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. Eliminación local, entra una eliminación en la actualización**

1. El desarrollador A mueve Foo.c a Bar.c y lo confirma en el repositorio.
2. El desarrollador B mueve Foo.c a Bix.c.

Una actualización de la copia de trabajo del desarrollador B acaba con un conflicto de árbol:

- Bix.c se marca como añadido con historia.
- Bar.c se añade a la copia de trabajo con el estado 'normal'.
- Foo.c se marca como borrado y tiene un conflicto de árbol.

Para resolver este conflicto, el desarrollador B tiene que averiguar qué nombre de archivo tiene ahora el archivo en conflicto Foo.c que fue renombrado/movido en el repositorio. Esto puede hacerse utilizando el diálogo de registro.

Luego el desarrollador B tiene que decidir qué nuevo nombre de archivo de Foo.c mantiene - el del desarrollador A o el renombrado que hizo él mismo.

Después que el desarrollador B haya resuelto manualmente el conflicto, el conflicto de árbol debe marcarse como resuelto mediante el botón del diálogo de editar conflictos.

#### **4.6.3.4. Falta en local, entra un cambio en la fusión**

1. El desarrollador A, que está trabajando en el tronco, modifica Foo.c y lo confirma en el repositorio

2. El desarrollador B, que está trabajando en una rama, mueve `Foo.c` a `Bar.c` y lo confirma en el repositorio

Una fusión de los cambios en el tronco del desarrollador A con los cambios de la copia de trabajo de la rama del desarrollador B acaba con un conflicto de árbol:

- `Bar.c` ya está en la copia de trabajo con estado 'normal'.
- `Foo.c` se marca como faltante con un conflicto de árbol.

Para resolver este conflicto, el desarrollador B tiene que marcar el archivo como resuelto en el diálogo de edición de conflictos, lo que lo eliminará de la lista de conflictos. Luego tendrá que decidir si copia el archivo faltante `Foo.c` desde el repositorio a la copia de trabajo, si fusiona los cambios del desarrollador A hechos a `Foo.c` en el archivo renombrado `Bar.c`, o si desea ignorar los cambios marcando el conflicto como resuelto y no haciendo nada más.

Tenga en cuenta que si copia el archivo faltante desde el repositorio y luego marca como resuelto, su copia de eliminará de nuevo. Tiene que resolver el conflicto antes.

#### 4.6.3.5. Edición local, entra un borrado en la fusión

1. El desarrollador A, que está trabajando en el tronco, mueve `Foo.c` a `Bar.c` y lo confirma en el repositorio.
2. El desarrollador B, que está trabajando en una rama, modifica el archivo `Foo.c` y lo confirma en el repositorio.
1. El desarrollador A, que está trabajando en el tronco, mueve la carpeta padre `FooFolder` a `BarFolder` y lo confirma en el repositorio.
2. El desarrollador B, que está trabajando en un rama, modifica `Foo.c` en su copia de trabajo.

Una fusión de los cambios en el tronco del desarrollador A con los cambios de la copia de trabajo de la rama del desarrollador B acaba con un conflicto de árbol:

- `Bar.c` se marca como añadido.
- `Foo.c` se marca como modificado con un conflicto de árbol.

El desarrollador B tiene ahora que decidir si desea continuar con la reorganización del desarrollador A y fusionar sus cambios en los archivos correspondientes de la nueva estructura, o simplemente revertir los cambios de A y mantener el archivo 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. Eliminación local, entra un borrado en la fusión

1. El desarrollador A, que está trabajando en el tronco, mueve `Foo.c` a `Bar.c` y lo confirma en el repositorio.
2. El desarrollador B, que está trabajando en una rama, mueve `Foo.c` a `Bix.c` y lo confirma en el repositorio.

Una fusión de los cambios en el tronco del desarrollador A con los cambios de la copia de trabajo de la rama del desarrollador B acaba con un conflicto de árbol:

- `Bix.c` se marca con el estado normal (no modificado).

- Bar.c se marca como añadido con historia.
- Foo.c se marca como faltante con un conflicto de árbol.

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.

Luego el desarrollador B tiene que decidir qué nuevo nombre de archivo de Foo.c mantiene - el del desarrollador A o el renombrado que hizo él mismo.

Después que el desarrollador B haya resuelto manualmente el conflicto, el conflicto de árbol debe marcarse como resuelto mediante el botón del diálogo de editar conflictos.

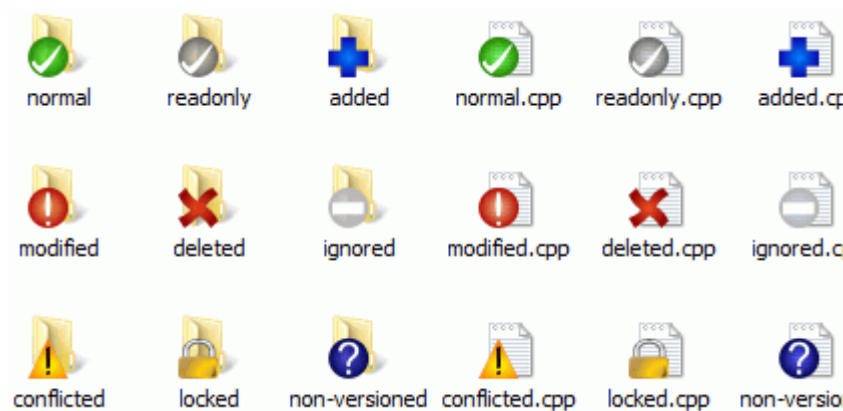
#### 4.6.3.7. Otros conflictos de árboles

Hay otros casos etiquetados como conflictos de árboles simplemente porque el conflicto implica un directorio en vez de un archivo. Por ejemplo, si añade un directorio con el mismo nombre tanto al tronco como a la rama y trata de fusionarlos obtendrá un conflicto de árbol. Si quiere mantener el directorio del objetivo de la fusión, marque el conflicto como resuelto. Si quiere usar el de la fuente de fusión, entonces necesita primero hacer un SVN delete para borrar el directorio del objetivo y volver a lanzar la fusión. Si necesita algo más complejo deberá hacerlo manualmente.

### 4.7. Obteniendo información del estado

Mientras está trabajando en su copia de trabajo a menudo necesitará saber qué archivos ha cambiado/añadido/borrado o renombrado, o incluso qué archivos han sido cambiados y confirmados por los demás.

#### 4.7.1. Íconos superpuestos



**Figura 4.12. Explorador mostrando íconos superpuestos**

Ahora que ha obtenido una copia de trabajo desde un repositorio de Subversion, puede ver sus archivos en el explorador de Windows con los íconos cambiados. Ésta es una de las razones por las que TortoiseSVN es tan popular. TortoiseSVN añade lo que se llama un ícono superpuesto al ícono de cada archivo que se superpone al ícono original del archivo. Dependiendo del estado en Subversion del archivo, el ícono superpuesto es diferente.



Una copia de trabajo recién obtenida tiene una marca verde como ícono superpuesto. Esto significa que el estado de Subversion es normal.



En cuanto empiece a editar un archivo, el estado cambia a `modificado` y el ícono superpuesto cambia entonces a una marca de exclamación roja. De esta forma puede ver fácilmente qué archivos se han cambiado desde la última vez que actualizó su copia de trabajo, y que necesitan ser confirmados.



Si durante una actualización ocurre un `conflicto`, el ícono cambia a un signo de exclamación amarillo.



Si ha establecido la propiedad `svn:needs-lock` en un archivo, Subversion establece ese archivo como de sólo-lectura hasta que obtenga un bloqueo en él. Estos archivos tienen este ícono superpuesto para indicarle que debe obtener un bloqueo antes que pueda editarlo.



Si ha obtenido un bloqueo sobre un archivo, y el estado de Subversion es `normal`, este ícono superpuesto le recordará que debería liberar el bloqueo si no lo está utilizando para permitir a los demás que puedan confirmar sus cambios en el archivo.



Este ícono le muestra que algunos archivos o carpetas dentro de la carpeta actual se han marcado para ser eliminados del control de versiones, o bien que falta un archivo que está bajo el control de versiones dentro de una carpeta.



El signo más le indica que el archivo o carpeta está programado para ser añadido al control de versiones.



La barra le indica que el archivo o carpeta está *ignorado* para los asuntos de control de versiones. Esta superposición es opcional.



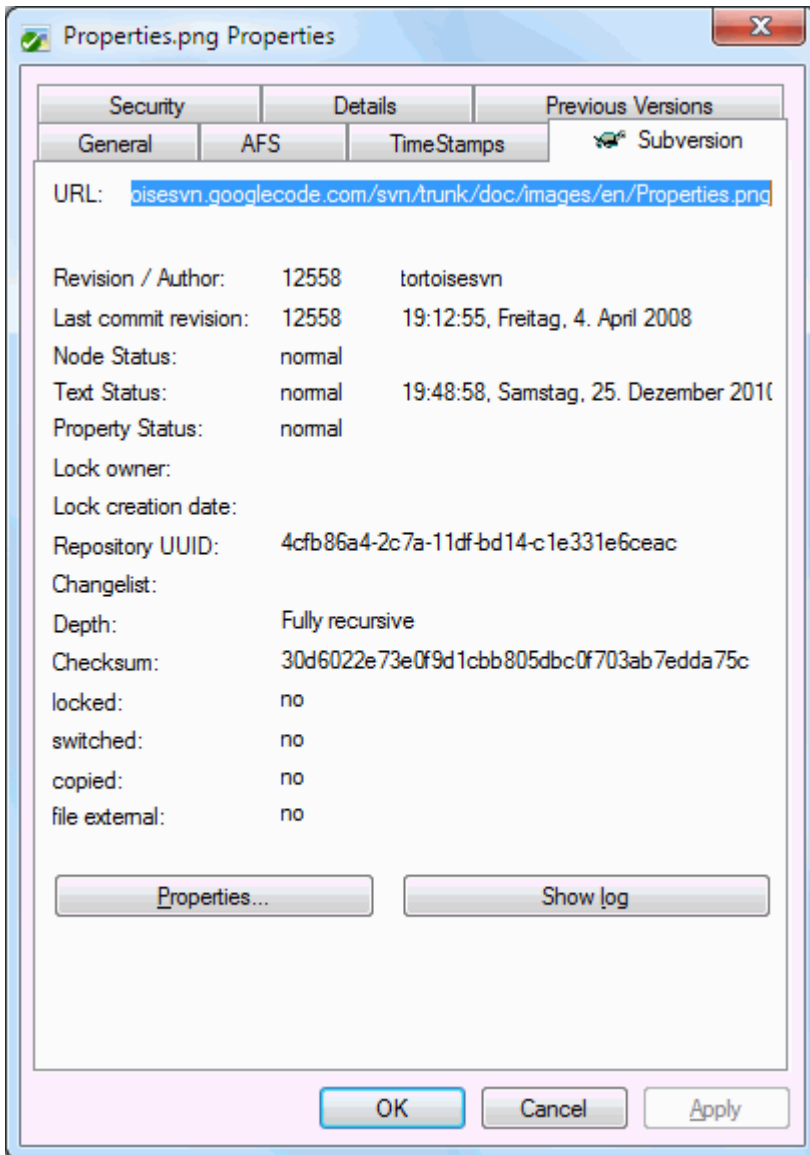
Este ícono muestra los archivos y carpetas que no están bajo el control de versiones pero tampoco han sido ignorados. Este ícono superpuesto es opcional.

De hecho, puede que se encuentre con que no todos estos íconos se utilizan en su sistema. Esto se debe a que el número de íconos superpuestos permitidos por Windows es muy limitado y si está utilizando también una versión antigua de TortoiseCVS, no hay suficientes lugares disponibles para íconos superpuestos. TortoiseSVN intenta ser un “Buen Ciudadano ®” y limita su uso de superposiciones para darles una oportunidad al resto de las aplicaciones.

Ahora que hay muchos otros clientes Tortoise (TortoiseCVS, TortoiseHg, ...) el límite de iconos comienza a ser un problema real. Para evitar esto, el proyecto TortoiseSVN introdujo un set de iconos compartidos, cargados como una DLL, que pueden ser usados por todos los clientes Tortoise. Compruebe con el proveedor de su cliente si esto ya se ha integrado :-)

Si desea una descripción de cómo se corresponden los íconos superpuestos con los estados de Subversion y otros detalles técnicos, lea [Sección F.1, “Íconos superpuestos”](#).

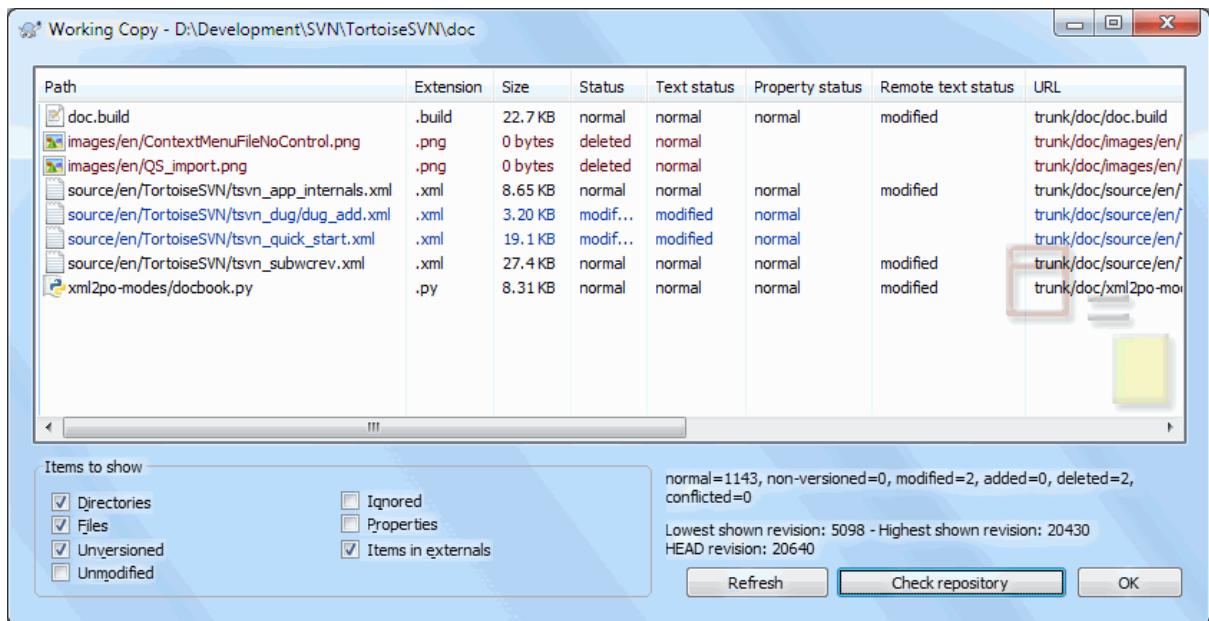
## 4.7.2. Estado detallado



**Figura 4.13. Página de propiedades del Explorador, pestaña Subversion**

A veces desea obtener información más detallada sobre un archivo o directorio que la que proporcionan los íconos superpuestos. Puede obtener toda la información que Subversion proporciona en el diálogo de propiedades del explorador. Simplemente seleccione el archivo o directorio y seleccione Menú de Windows → Propiedades en el menú contextual (atención: ésta es la entrada normal del menú de propiedades que el explorador proporciona, ¡no la que está en el submenú de TortoiseSVN!). En el cuadro de diálogo de propiedades, TortoiseSVN ha añadido una nueva página de propiedades para los archivos y carpetas bajo control de Subversion, donde puede ver toda la información relevante sobre el archivo/directorio seleccionado.

### 4.7.3. Estado local y remoto



**Figura 4.14. Comprobar modificaciones**

A menudo es muy útil saber qué archivos ha cambiado y también qué archivos han cambiado y confirmado los demás. Ahí es donde viene bien el comando TortoiseSVN → Comprobar Modificaciones.... Este diálogo le muestra todos los archivos que ha cambiado de alguna forma en su copia de trabajo, y además todos los archivos no versionados que pueda tener.

Si pulsa en el botón Comprobar Repositorio también puede comprobar los cambios en el repositorio. De esa forma puede comprobar antes de hacer una actualización si es posible que haya un conflicto. También puede actualizar archivos concretos desde el repositorio sin actualizar la carpeta completa. Por defecto, el botón Comprobar Repositorio sólo obtiene el estado remoto con la profundidad obtenida de la copia de trabajo. Si desea ver todos los archivos y carpetas del repositorio, incluso aquellas que no ha obtenido, entonces tendrá que pulsar la tecla **M**ays mientras hace click en el botón Comprobar Repositorio.

El diálogo utiliza un código de colores para resaltar el estado.

Azul

Ítems modificados localmente.

Púrpura

Ítems añadidos. Los ítems que han sido añadidos con historia tienen un signo + en la columna Estado del texto, y un texto de ayuda que le muestra de dónde ha sido copiado.

Rojo oscuro

Ítems faltantes o borrados.

Verde

Ítems modificados localmente y en el repositorio. Los cambios se fusionarán al actualizar. Ésto *puede* producir conflictos al actualizar.

Rojo brillante

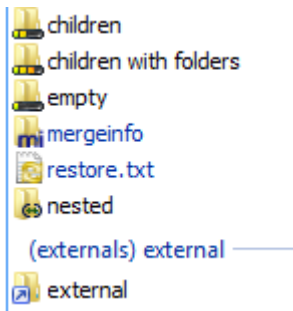
Ítems modificados localmente y borrados en el repositorio, o modificados en el repositorio y borrados localmente. Esto *producirá* conflictos al actualizar.

Negro

Ítems sin cambios y sin versionar.

Este es el esquema de colores por defecto, pero puede personalizar dichos colores utilizando el diálogo de configuración. Para más información, lea [Sección 4.31.1.5, “Configuración de colores de TortoiseSVN”](#).

Se usan íconos superpuestos para indicar también otros estados. La captura de pantalla siguiente muestra todas las posibles superposiciones que se pueden llegar a ver.



Los íconos superpuestos se muestran para los siguientes estados:

- Profundidad de obtención *vacía*, significando solo el propio ítem.
- Profundidad de obtención *archivos*, significando solo el propio ítem y todos los archivos hijo sin carpetas hijas.
- Profundidad de obtención *inmediata*, significando solo el propio ítem y todos los archivos y carpetas hijos, pero sin los hijos de las carpetas hijas.
- Ítems anidados, como copias de trabajo dentro de la copia de trabajo.
- Ítems externos, como todos los ítems añadidos usando la propiedad `svn:externals`.
- Ítems restaurados tras una confirmación. Para más información, lea [Sección 4.4.3, “Confirmar ficheros parcialmente”](#).
- Ítems con modificación de propiedades, pero solo en la propiedad `svn:mergeinfo`. Si tiene cualquier otra propiedad modificada, el icono superpuesto no se usa.

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

Desde el menú contextual del diálogo puede mostrar un resumen de los cambios. Compruebe los cambios locales que *usted* ha hecho utilizando Menú Contextual → Comparar con Base. Compruebe los cambios en el repositorio hechos por los demás utilizando Menú Contextual → Mostrar Diferencias como Diff Unificado.

También puede revertir cambios en archivos individuales. Si ha borrado un archivo de forma accidental, se mostrará como *Falta* y puede utilizar

Los archivos sin versionar y los ignorados se pueden enviar a la papelera de reciclaje desde aquí utilizando Menú Contextual → Eliminar. Si quiere eliminar los archivos de forma definitiva (sin utilizar la papelera de reciclaje) pulse la tecla **Mayúsculas** mientras hace click en Eliminar.

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

Las columnas son personalizables. Si hace click con el botón derecho en cualquier cabecera de columna verá un menú contextual que le permite seleccionar qué columnas se muestran. También puede cambiar el ancho de la columna utilizando el manejador de arrastre que aparece cuando mueve el cursor sobre el límite de una columna. Estas personalizaciones se mantienen, por lo que verá los mismos encabezados la próxima vez.

Si está trabajando al mismo tiempo en varias tareas sin relación entre ellas, también puede agrupar los archivos juntos en listas de cambios. Para más información, lea [Sección 4.4.2, “Listas de cambios”](#).

En la parte inferior del diálogo puede ver un resumen del rango de revisiones del repositorio en uso en su copia de trabajo. Estas son revisiones *confirmadas*, no las revisiones *actualizadas*; representan el rango de revisiones donde estos archivos fueron confirmados por última vez, no las revisiones a las que se han actualizado. Tenga en

cuenta que el rango de revisiones mostrado se aplica sólo a los ítems mostrados, no a la copia de trabajo completa. Si quiere ver esa información para la copia de trabajo completa debe seleccionar la casilla **Mostrar archivos no modificados**.



### Sugerencia

Si quiere una vista llana de su copia de trabajo, por ejemplo mostrando todos los archivos y carpetas en todos los niveles de su jerarquía de carpetas, entonces el diálogo **Comprobar modificaciones** es la forma más sencilla de conseguirlo. Sólomente seleccione la casilla **Mostrar archivos no modificados** para ver todos los archivos de su copia de trabajo.



### Reparando renombrados externos

A veces los archivos se renombran fuera de Subversion, y se muestran en la lista de archivos como un archivo faltante y un archivo no versionado. Para evitar perder la historia necesita notificar a Subversion su conexión. Simplemente seleccione tanto el nombre antiguo (faltante) como el nombre nuevo (sin versionar) y utilice **Menú contextual** → **Reparar movimiento** para emparejar los dos archivos como un renombrado.



### Reparando copias externas

Si hizo una copia de un archivo pero olvidó usar el comando de Subversion para hacerlo, puede reparar esa copia para que el nuevo archivo no pierda su historial. Simplemente seleccione tanto el nombre antiguo (normal o modificado) como el nuevo (no versionado) y utilice **Menú contextual** → **Reparar Copia** para emparejar los dos archivos como una copia.

## 4.7.4. Viendo diferencias

A menudo querrá mirar dentro de sus archivos, para echar un vistazo a lo que ha cambiado. Puede llevar esto a cabo seleccionando un archivo que haya cambiado, y seleccionando **Diferenciar** desde el menú contextual de TortoiseSVN. Esto inicia el visor externo de diferencias, que comparará el archivo actual con la copia prístina (revisión **BASE**), que se guardó tras su obtención o tras la última actualización.



### Sugerencia

Puede mostrar diferencias incluso cuando no está dentro de una copia de trabajo, o cuando tiene múltiples versiones del archivo alrededor:

Seleccione los dos archivos que desea comparar en el explorador (por ejemplo, utilizando la tecla **Ctrl** y el ratón) y seleccione **Diferenciar** del menú contextual de TortoiseSVN. El archivo que haya pulsado en último lugar (el que tiene el foco, es decir, el rectángulo con puntos) será tomado como más nuevo.

## 4.8. Listas de cambios

En un mundo ideal, sólo trabajará en una cosa cada vez, y su copia de trabajo sólo contendrá un conjunto de cambios lógicos. Vale, de vuelta al mundo real. A menudo ocurre que tiene que trabajar en varias tareas sin relación entre sí a la vez, y cuando mira en el diálogo de confirmar, todos los cambios están juntos y mezclados. La característica *lista de cambios* le ayuda a hacer agrupaciones de archivos, facilitando ver qué se está haciendo. Por supuesto esto sólo funciona si los cambios no se superponen. Si dos tareas diferentes afectan al mismo archivo, no hay forma de separar los cambios.

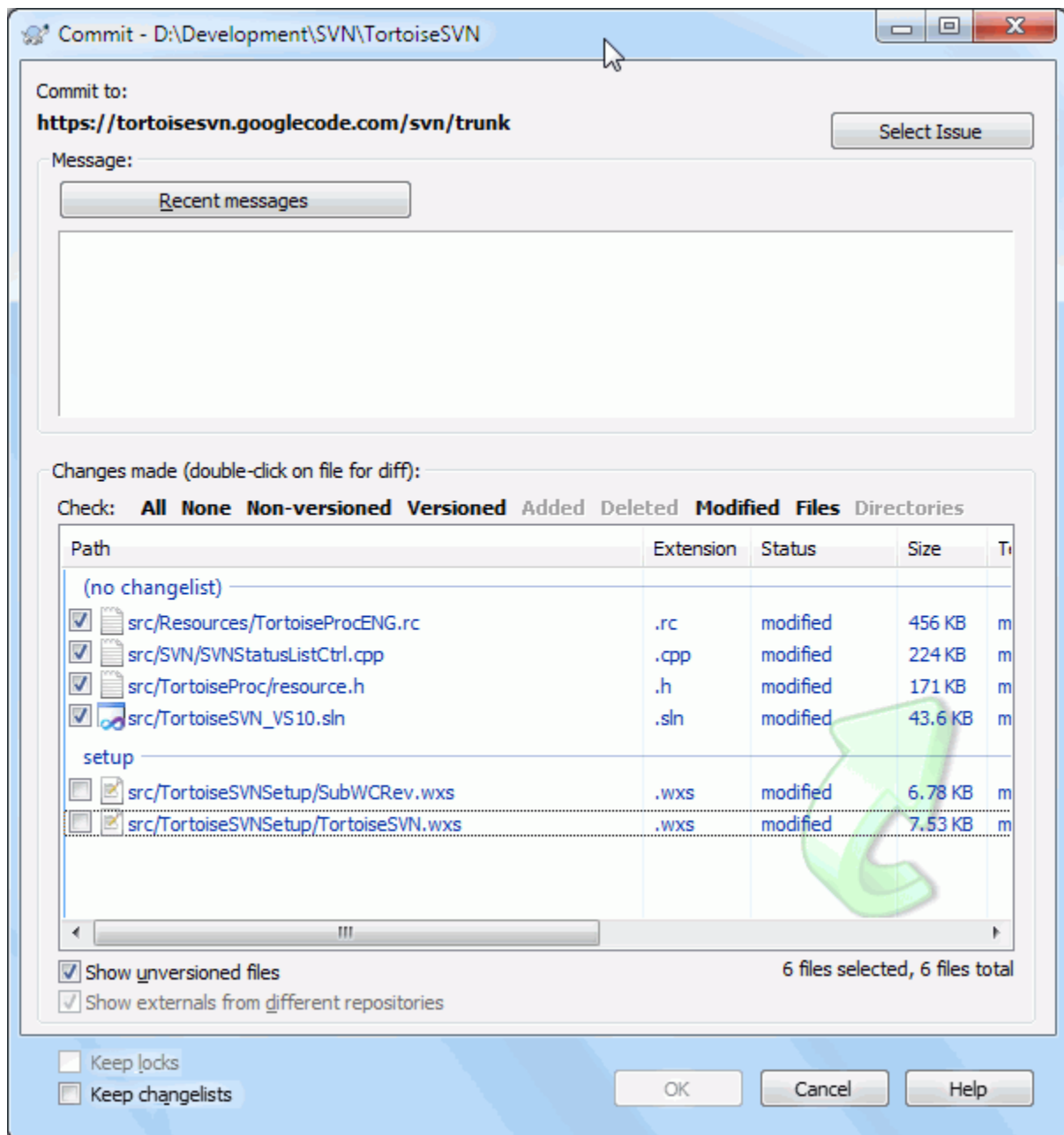
Puede ver las listas de cambios en varios lugares, pero los más importantes son el diálogo de confirmación y el de comprobar modificaciones. Empecemos en el diálogo comprobar modificaciones después que haya estado



trabajando en varias características y varios archivos. La primera vez que abra el diálogo, todos los archivos modificados se muestran juntos. Supongamos que ahora quiere organizar las cosas y agrupar esos archivos según la característica.

Seleccione uno o más archivos y utilice **Menú contextual** → **Mover a la lista de cambios** para añadir un ítem a una lista de cambios. Inicialmente no habrá listas de cambios, por lo que la primera vez que ejecute esto tendrá que crear una nueva lista de cambios. Déle un nombre que describa para qué la está utilizando, y pulse **Aceptar**. El diálogo de confirmación cambiará para mostrar agrupaciones de ítems.

Una vez que haya creado una lista de cambios puede arrastrar y soltar ítems en ella, tanto desde otra lista de cambios como desde el Explorador de Windows. Arrastrar desde el Explorador puede ser útil ya que le permite añadir ítems a una lista de cambios antes que el archivo sea modificado. Puede hacer eso desde el diálogo comprobar modificaciones, pero sólo si muestra todos los archivos no modificados.



**Figura 4.15. Diálogo de confirmación con listas de cambios**

En el diálogo de confirmación puede ver esos mismos archivos, agrupados por listas de cambios. Además de dar una indicación visual inmediata de las agrupaciones, también puede utilizar los encabezados de grupo para seleccionar qué archivos confirmar.

TortoiseSVN reserva un nombre de lista de cambios para su propio uso, llamada `ignore-on-commit`. Se utiliza para marcar los archivos versionados que casi nunca querrá confirmar, incluso aunque tengan cambios locales. Esta característica se describe en [Sección 4.4.4, “Excluyendo ítems de la lista de confirmación”](#).

Cuando confirme archivos que pertenezcan a una lista de cambios, normalmente no es necesario que dichos archivos sigan perteneciendo a la lista de cambios. Por este motivo, y por defecto, los archivos se eliminan de las listas de cambios automáticamente al ser confirmados. Si desea mantener el archivo en su lista de cambios, utilice la casilla **Mantener listas de cambios** en la parte inferior del diálogo de confirmación.



### Sugerencia

Las listas de cambios son una característica únicamente del cliente local. La creación y eliminación de listas de cambios no afectan ni al repositorio ni a la copia de trabajo de ningún otro cliente. Simplemente son una forma conveniente para organizar sus archivos.



### Aviso

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.

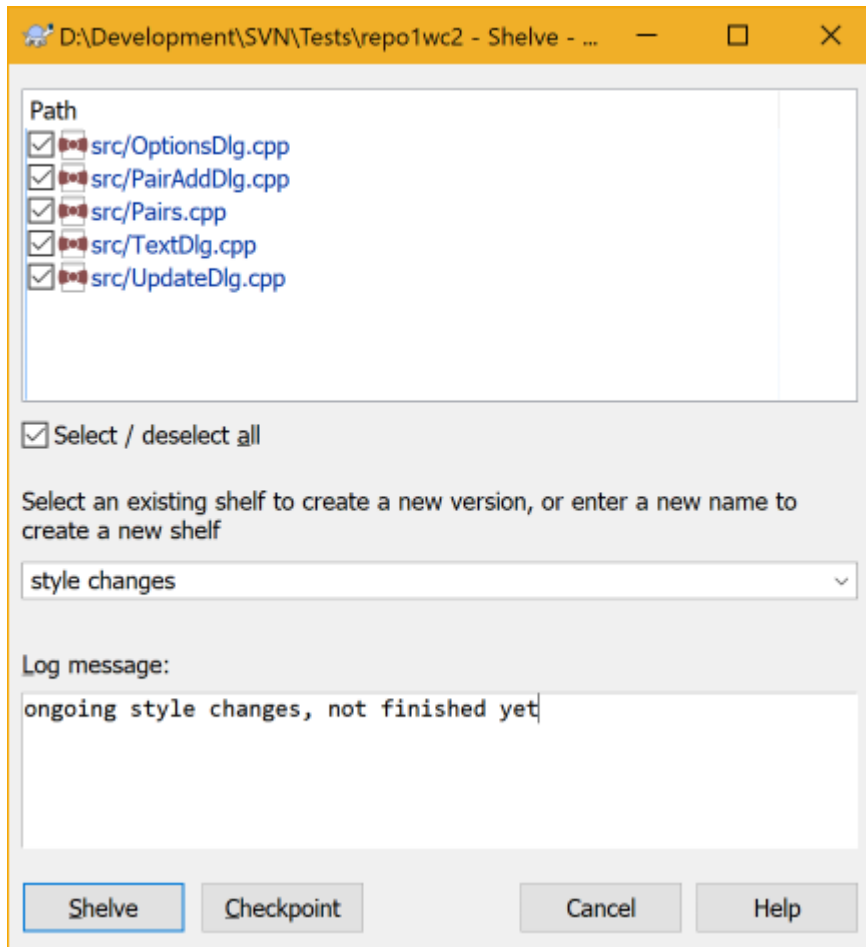
## 4.9. Shelving

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.

Dos nuevos comandos son implementados para esto. Uno para archivar y uno para desarchivar.

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. Diálogo de archivado**

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**.

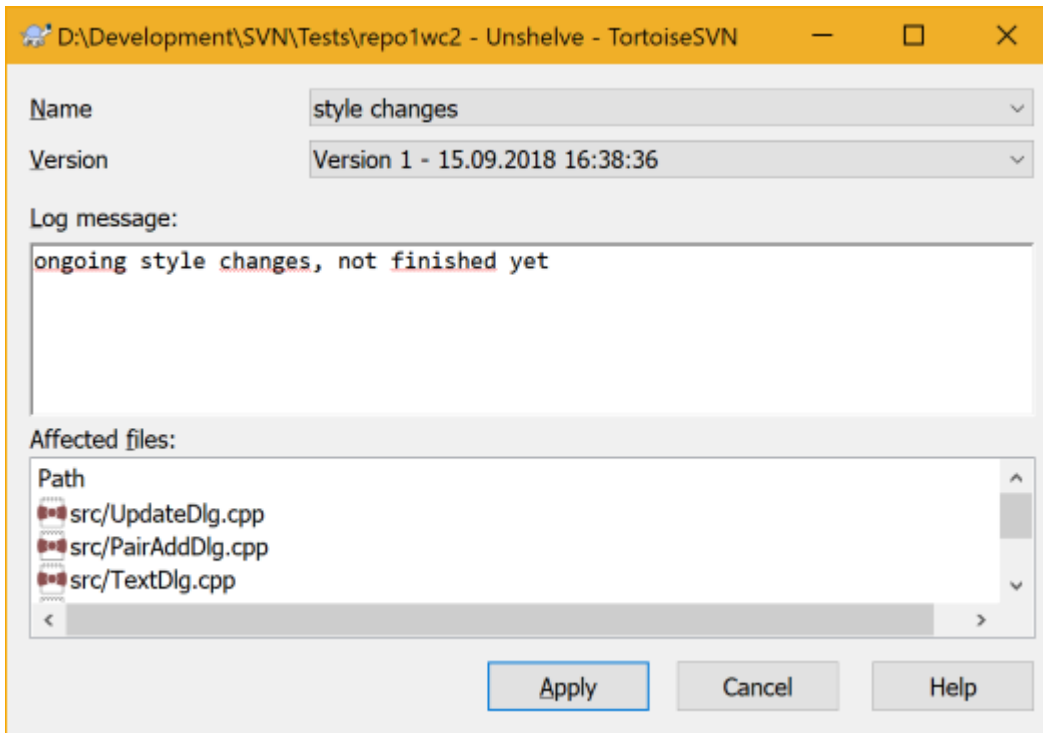


Figura 4.17. Unshelve dialog



### Sugerencia

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. Diálogo de Registro de revisiones

Para cada cambio que haga y confirme, debería proporcionar un mensaje de registro de ese cambio. Así podrá averiguar después qué cambios hizo y por qué, y tendrá un registro detallado para su proceso de desarrollo.

El diálogo de Registro de revisiones recopila todos esos mensajes de registro y se los enseña. La pantalla se divide en tres paneles.

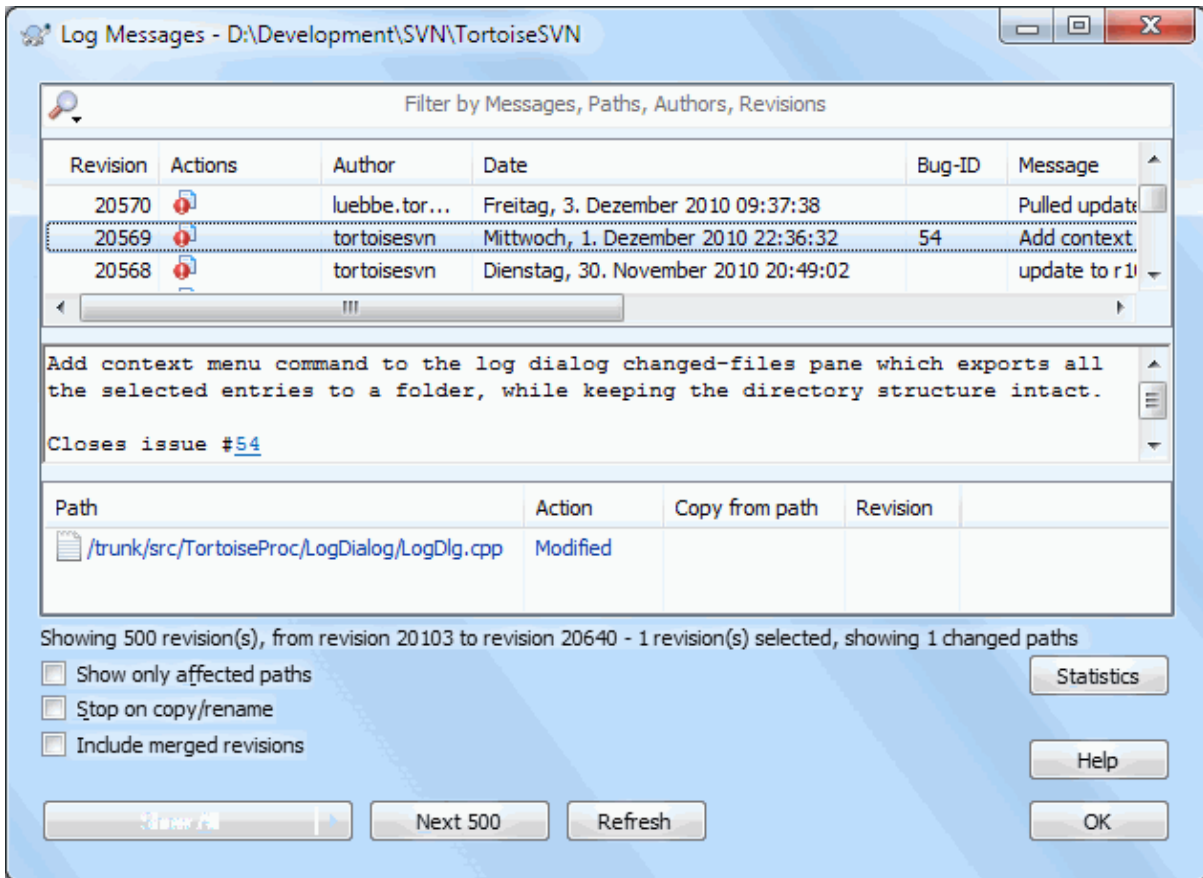
- El panel superior le muestra una lista de revisiones donde se confirmaron cambios a los archivos/carpetas. Este sumario incluye la fecha y la hora, la persona que confirmó la revisión y el inicio del mensaje de registro.

Las líneas azules indican que algo se ha copiado a esta línea de desarrollo (quizás desde una rama).

- El panel medio le muestra el mensaje de registro completo para la revisión seleccionada.
- El panel inferior le muestra una lista de todos los archivos y carpetas que se cambiaron como parte de la revisión seleccionada.

Pero hace mucho más que eso - le proporciona comandos del menú contextual que puede utilizar para obtener aún más información de la historia del proyecto.

#### 4.10.1. Invocando el diálogo de Registro de revisiones



**Figura 4.18. El diálogo de Registro de revisiones**

Hay varios lugares desde los que puede mostrar el diálogo de Registro:

- Desde el submenú contextual de TortoiseSVN
- Desde la página de propiedades
- Desde el diálogo de Progreso después que termine una actualización. En ese caso el diálogo de Registro sólo le mostrará aquellas revisiones que cambiaron desde su última actualización
- Desde el navegador del repositorio.

Si el repositorio no está disponible verá el diálogo ¿Desea trabajar sin conexión?, descrito en [Sección 4.10.10, "Modo sin conexión"](#).

#### 4.10.2. Acciones del registro de revisiones

El panel superior tiene una columna Acciones que contiene íconos que resumen qué se ha hecho en esa revisión. Hay cuatro íconos diferentes, cada uno mostrado en su propia columna.



Si una revisión modificó un archivo o un directorio, se muestra el ícono *modificado* en la primera columna.



Si en una revisión se añadió un archivo o directorio, se muestra el ícono *añadido* en la segunda columna.



Si en una revisión se eliminó un archivo o directorio, se muestra el ícono *eliminado* en la tercera columna.



Si una revisión reemplazó un archivo o un directorio, se muestra el ícono *reemplazado* en la cuarta columna.



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.

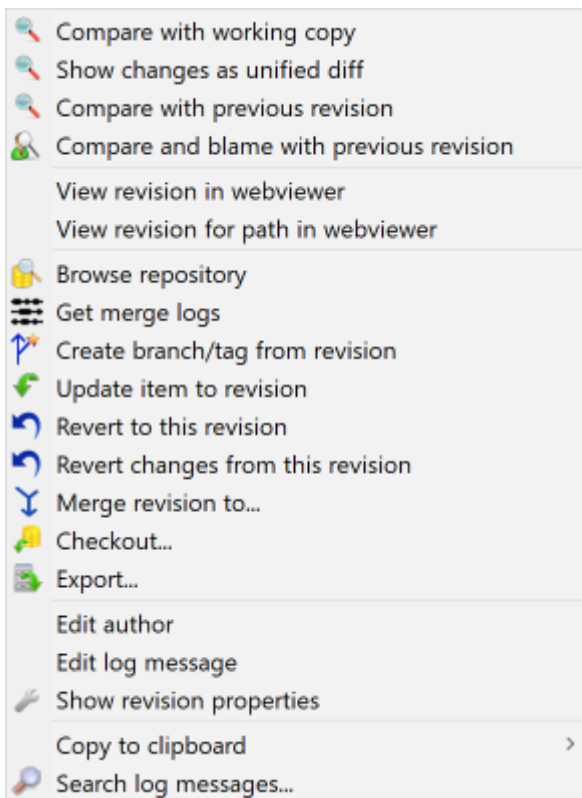


If a revision merged a file or directory, the *merged* icon is shown in the fourth column.



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

### 4.10.3. Obteniendo información adicional



**Figura 4.19. El panel superior del diálogo de Registro de revisiones con el menú contextual**

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 con la copia de trabajo

Comparar las revisiones seleccionadas con su copia de trabajo. La Herramienta de Diferencias por defecto es TortoiseMerge que se proporciona con TortoiseSVN. Si el diálogo de registro es de una carpeta, esto mostrará una lista de sus archivos cambiados, y le permitirá revisar los cambios hechos a cada archivo individualmente.

#### Comparar y mostrar autoría con la BASE de trabajo

Obtener la información de autoría de la revisión seleccionada, y del archivo en su BASE de trabajo, y comparar los resultados utilizando una herramienta visual de diferencias. Lea [Sección 4.24.2, “Autoría de las diferencias”](#) para más detalles. (Sólo para archivos).

#### Mostrar cambios como diff unificado

Ver los cambios hechos en la revisión seleccionada como archivo de diff unificado (formato de parche GNU). Esto le muestra sólo las diferencias con unas pocas líneas de contexto. Es más difícil de leer que una comparación visual de archivos, pero le mostrará todos los cambios juntos en un formato compacto.

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

#### Comparar con la revisión anterior

Compara la revisión seleccionada con la revisión anterior. Esto funciona de forma similar a comparar con su copia de trabajo. Para carpetas esta opción mostrará primero el diálogo de archivos cambiados permitiéndole seleccionar los archivos a comparar.

#### Comparar y mostrar autoría con la revisión anterior

Mostrar el diálogo de archivos modificados permitiéndole seleccionar archivos. Obtiene la información de autoría de la revisión seleccionada, y de la revisión anterior, y compara los resultados utilizando una herramienta visual de diferencias (sólo para carpetas).

#### Guardar revisión en...

Almacenar la revisión seleccionada en un archivo para tener una versión antigua de ese archivo. (Sólo para archivos)

#### Abrir / Abrir con...

Abrir el archivo seleccionado, bien con el visor por defecto para ese tipo de archivo, o bien con el programa que elija. (Sólo para archivos)

#### Autoría...

Muestra la autoría del archivo hasta la revisión seleccionada. (sólo para archivos)

#### Explorar el repositorio

Abrir el navegador de repositorios para examinar el archivo o la carpeta seleccionados en el repositorio tal y como estaban en la revisión seleccionada.

#### Crear rama/etiqueta desde la revisión

Crea una rama o una etiqueta desde la revisión seleccionada. Esto es útil por ejemplo si se le olvidó crear una etiqueta y ya ha confirmado algunos cambios que se supone que no deben ir en esa versión.

#### Actualizar ítem a la revisión

Actualizar su copia de trabajo a la revisión seleccionada. Útil si quiere hacer que su copia de trabajo refleje un momento en el pasado, o si ha realizado confirmaciones posteriores en el repositorio y quiere actualizar su copia de trabajo un paso cada vez. Es mejor actualizar un directorio completo en su copia de trabajo, no sólo un archivo, ya que si no su copia de trabajo podría ser inconsistente.

Si desea deshacer un cambio anterior de forma permanente, utilice **Revertir a esta revisión**

#### Revertir a esta revisión

Revierte a una revisión anterior. Si ha hecho varios cambios, y luego decide que realmente desea volver a dejar las cosas como estaban en la revisión N, este es el comando que necesita. Los cambios se revierten en su copia de trabajo, por lo que esta operación *no* afecta al repositorio hasta que confirme los cambios. Tenga en cuenta que esto desharrá *todos* los cambios que se hayan realizado tras la revisión seleccionada, reemplazando el archivo o la carpeta con la versión anterior.

Si su copia de trabajo está en un estado no modificado, después de realizar esta acción su copia de trabajo se mostrará como modificada. Si ya tiene cambios locales, este comando fusionará los cambios *de deshacer* en su copia de trabajo.

Lo que ocurre internamente es que Subversion realiza una fusión inversa de todos los cambios realizados después de la revisión seleccionada, deshaciendo el efecto de esas confirmaciones sobre las previas.

Si después de realizar esta acción decide que desea *deshacer lo deshecho* y volver a obtener su copia de trabajo a su estado previo sin modificaciones, deberá utilizar TortoiseSVN → Revertir desde el Explorador de Windows, lo que descartará los cambios locales hechos por esta acción de fusión inversa.

Si desea simplemente ver cómo estaba un archivo o una carpeta en una revisión anterior, utilice Actualizar a la revisión o Guardar revisión como....

#### Revertir los cambios hechos en esta revisión

Deshace los cambios que se hicieron en la revisión seleccionada. Estos cambios se deshacen en su copia de trabajo, ¡por lo que esta operación *no* afecta al repositorio en absoluto! Tenga en cuenta que esto deshacerá únicamente los cambios hechos en esa revisión; no reemplaza su copia de trabajo con el archivo completo tal y como estaba en la revisión anterior. Esto es muy útil para deshacer un cambio anterior cuando ya se han hecho otros cambios que no tienen que ver con él.

Si su copia de trabajo está en un estado no modificado, después de realizar esta acción su copia de trabajo se mostrará como modificada. Si ya tiene cambios locales, este comando fusionará los cambios *de deshacer* en su copia de trabajo.

Lo que ocurre internamente es que Subversion realiza una fusión inversa de esa única revisión, deshaciendo sus efectos sobre la confirmación anterior.

Puede *deshacer lo deshecho* tal y como se describe arriba en Revertir a esta revisión.

#### Fusionar revisión en...

Fusiona la(s) revisión(es) seleccionada(s) en una copia de trabajo diferente. Un diálogo de selección de carpeta le permitirá elegir la copia de trabajo donde desea fusionar, pero después de eso no hay diálogo de confirmación, ni oportunidad de probar la fusión sin ejecutarla realmente. Es una buena idea fusionar en una copia de trabajo sin cambios, ¡y así poder revertir los cambios si no funcionan! Esta es una funcionalidad útil si desea fusionar las revisiones seleccionadas de una rama en otra.

#### Obtener...

Hace una obtención nueva de la carpeta seleccionada en la revisión seleccionada. Esto muestra un diálogo para que confirme la URL y la revisión, y seleccione un lugar para la obtención.

#### Exportar...

Exporta la carpeta o el archivo seleccionado en la revisión seleccionada. Esto muestra un diálogo para que confirme la URL y la revisión, y para que seleccione un lugar para la exportación.

#### Editar autor / mensaje de registro

Editar el mensaje de registro o el autor adjunto a una confirmación anterior. Lea [Sección 4.10.7, “Cambiando el mensaje de registro y el autor”](#) para averiguar cómo funciona esto.

#### Mostrar propiedades de la revisión

Ver y editar cualquier propiedad de revisión, no sólo el mensaje de registro y el autor. Lea [Sección 4.10.7, “Cambiando el mensaje de registro y el autor”](#).

#### Copiar al portapapeles

Copia los detalles de registro de las revisiones seleccionadas al portapapeles. Esto copia el número de revisión, el autor, la fecha, el mensaje de registro y la lista de ítems cambiados para cada revisión.

#### Buscar mensaje de registro...

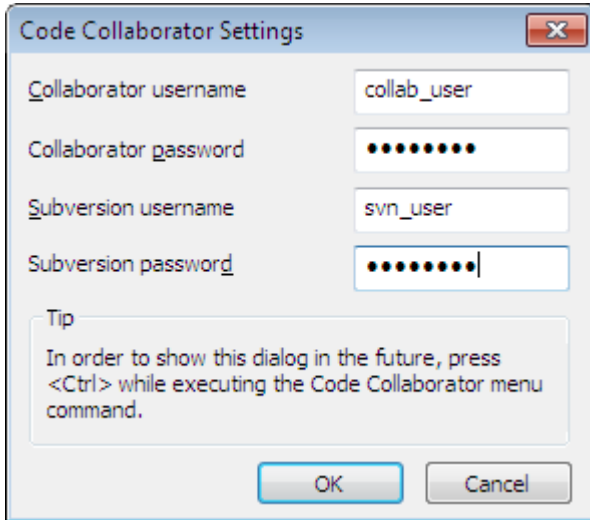
Buscar en los mensajes de registro el texto que desee. Esto busca en los mensajes de registro que ha introducido, y también en los sumarios de acción creados por Subversion (mostrados en el panel inferior). La búsqueda no distingue mayúsculas y minúsculas.

#### Create code collaborator review...

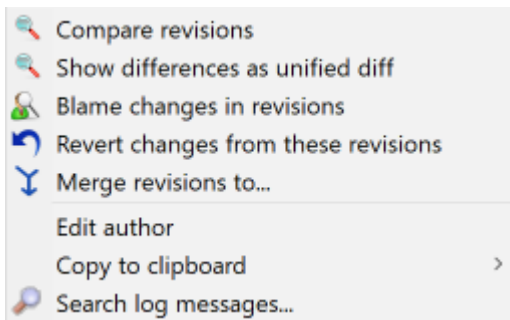
This menu is shown only if the SmartBear code collaborator tool is installed. When invoked for the first time, a dialog is shown prompting the user to enter user credentials for both code collaborator and SVN. Once the



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. Menú contextual del panel superior para 2 revisiones seleccionadas**

Si selecciona dos revisiones a la vez (utilizando el modificador habitual **Ctrl**), el menú contextual cambia y le ofrece menos opciones:

#### Comparar revisiones

Compara las dos revisiones seleccionadas utilizando una herramienta de diferencias visual. La herramienta de diferencias por defecto es TortoiseMerge que se proporciona con TortoiseSVN.

Si selecciona esta opción para una carpeta, aparecerá un diálogo posterior mostrando los archivos cambiados y ofreciéndole más opciones de diferenciación. Lea más sobre el diálogo Comparar Revisiones en [Sección 4.11.3, “Comparando carpetas”](#).

#### Autoría de las revisiones

Obtener la información de autoría de las dos revisiones y comparar los resultados utilizando una herramienta visual de diferencias. Lea [Sección 4.24.2, “Autoría de las diferencias”](#) para más detalles.

#### Mostrar diferencias como diff unificado

Ver las diferencias entre las dos revisiones seleccionadas como un archivo diff unificado. Esto funciona para archivos y carpetas.

#### Copiar al portapapeles

Copie los mensajes de registro al portapapeles tal y como se describió anteriormente.

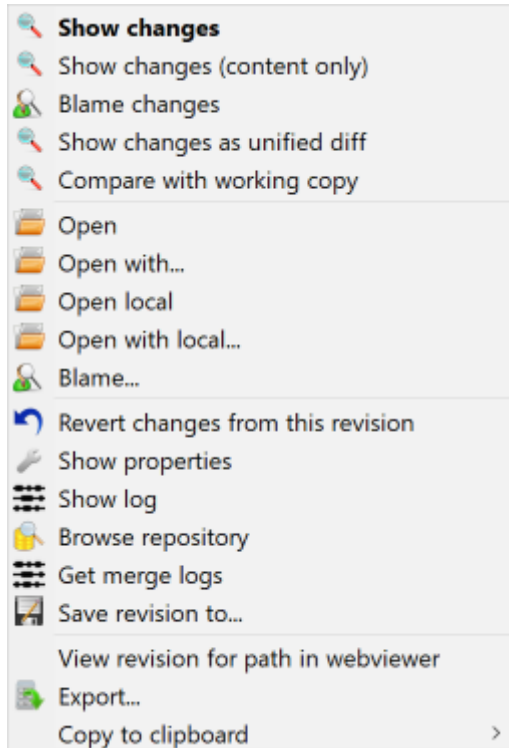
Buscar mensaje de registro...

Buscar mensajes de registro como se describe más arriba.

Si selecciona dos o más revisiones (utilizando los modificadores habituales **Ctrl** o **Mayúsculas**), el menú contextual incluirá una entrada para revertir todos los cambios que se hicieron en ese rango de revisiones. Ésta es la forma más sencilla para deshacer un grupo de revisiones de golpe.

También puede elegir fusionar en otra copia de trabajo las revisiones seleccionadas, como se describió más arriba.

Si todas las revisiones seleccionadas tienen el mismo autor, puede editar el autor de todas esas revisiones de un golpe.



**Figura 4.22. El panel inferior del diálogo de Registro con el menú contextual**

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

Mostrar cambios

Mostrar el registro de revisiones para el archivo seleccionado.

Autoría de los cambios

Obtener la información de autoría de la revisión seleccionada y de la revisión anterior del archivo seleccionado, y comparar los resultados utilizando una herramienta visual de diferencias. Lea [Sección 4.24.2, “Autoría de las diferencias”](#) para más detalles.

Mostrar como diff unificado

Mostrar los cambios del archivo en formato diff unificado. Este menú contextual sólo está disponible para los archivos que se muestran como *modificados*.

Abrir / Abrir con...

Abrir el archivo seleccionado, bien con el visor por defecto para ese tipo de archivo, o bien con el programa que elija.

Autoría...

Abre el diálogo Autoría, permitiéndole ver la información de autoría hasta la revisión seleccionada.

Revertir los cambios hechos en esta revisión

Revertir los cambios hechos al archivo seleccionado en esa revisión.

Mostrar propiedades

Ver las propiedades de Subversion del ítem seleccionado.

Mostrar registro

Mostrar el registro de revisiones para ese archivo seleccionado.

Obtener registros de fusiones

Muestra el registro de revisiones para el único archivo seleccionado, incluyendo los cambios fusionados. Más información en [Sección 4.10.6, “Características de registro de fusión”](#).

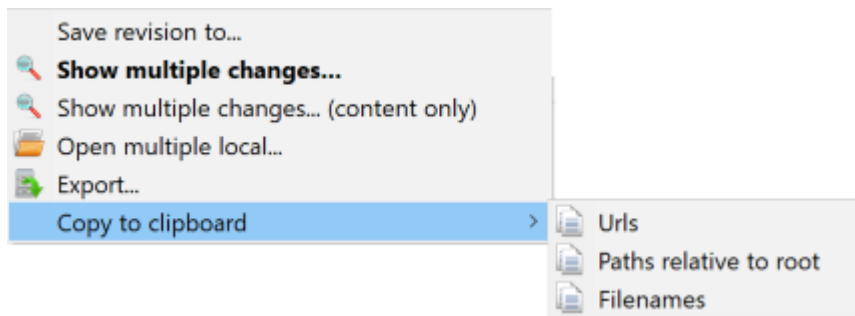
Guardar revisión en...

Grabar la revisión seleccionada a un archivo, para que pueda tener una versión antigua de ese archivo.

Exportar...

Exportar los elementos seleccionados en esta revisión a una carpeta, preservando la jerarquía de archivos.

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 revisión en...

Grabar la revisión seleccionada a un archivo, para que pueda tener una versión antigua de ese archivo.

Show multiple changes...

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

Abrir multiples locales...

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.



## Sugerencia

Puede haberse dado cuenta que a veces nos referimos a cambios y otras veces a diferencias. ¿Cuál es la diferencia?

Subversion utiliza números de revisión para dos cosas diferentes. Una revisión generalmente representa el estado de un repositorio en un momento en el tiempo, pero también puede utilizarse para representar el conjunto de cambios que generaron esa revisión; por ejemplo, “Hecho en la r1234” significa que los cambios confirmados en la revisión 1234 implementan la característica X. Para dejar claro en qué sentido se utilizan, empleamos dos términos distintos.

Si selecciona dos revisiones N y M, el menú contextual le ofrecerá mostrar las *diferencias* entre estas dos revisiones. En términos de Subversion esto es `diff -r M:N`.

Si selecciona una única revisión N, el menú contextual le ofrecerá mostrar los *cambios* realizados en esa revisión. En términos de Subversion esto es `diff -r N-1:N` o `diff -c N`.

El panel inferior le muestra los archivos cambiados en todas las revisiones seleccionadas, por lo que el menú contextual siempre ofrece mostrar *cambios*.

### 4.10.4. Obteniendo más mensajes de registro

El diálogo Registro no siempre le muestra todos los cambios que se hayan hecho alguna vez por unos cuantos motivos:

- En un repositorio grande, puede haber cientos o incluso miles de cambios, y obtenerlos todos puede llevar mucho tiempo. Normalmente estará interesado sólo en los cambios más recientes. Por defecto, el número de mensajes de registro obtenidos se limita a 100, pero puede cambiar este valor en TortoiseSVN → Configuración (Sección 4.31.1.2, “Configuración de diálogos de TortoiseSVN 1”),
- Cuando se marca la casilla Parar en copia/renombrado, Mostrar Registro se parará en el punto en el que el archivo o carpeta seleccionado se copió de algún otro lugar en el repositorio. Esto puede ser útil para buscar ramas (o etiquetas) porque se para en la raíz de esa rama, y le da una indicación rápida de los cambios hechos únicamente en esa rama.

Normalmente querrá dejar esta opción sin marcar. TortoiseSVN recuerda el estado de la casilla, por lo que respetará su preferencia.

Cuando se invoca el diálogo Mostrar Registro desde el diálogo Fusionar, la casilla siempre se marca por defecto. Esto es porque al fusionar lo más probable es buscar cambios en las ramas, y retroceder a la raíz de la rama no tiene sentido en ese caso.

Tenga en cuenta que Subversion actualmente implementa el renombrado como un par de copia/borrado, por lo que renombrar un archivo o carpeta también provocará que el diálogo de registro se pare si se marca esta opción.

Si desea ver más mensajes de registro, pulse **Siguientes 100** para obtener los siguientes 100 mensajes de registro. Puede repetir ésto tantas veces como sea necesario.

Al lado de este botón hay un botón multifunción que recuerda la última opción que utilizó. Pulse en la flecha para ver las otras opciones ofrecidas.

Utilice **Mostrar rango ...** si desea ver un rango específico de revisiones. Aparecerá un diálogo que le preguntará por la revisión de inicio y de fin.

Utilice **Mostrar Todos** si desea ver *todos* los mensajes de registro desde HEAD hasta la revisión 1.

Para refrescar la última revisión en caso de que haya habido otras confirmaciones mientras el diálogo estuvo abierto, presione la tecla **F5**.

Para refrescar el cache del log, presione la tecla **Ctrl-F5**.

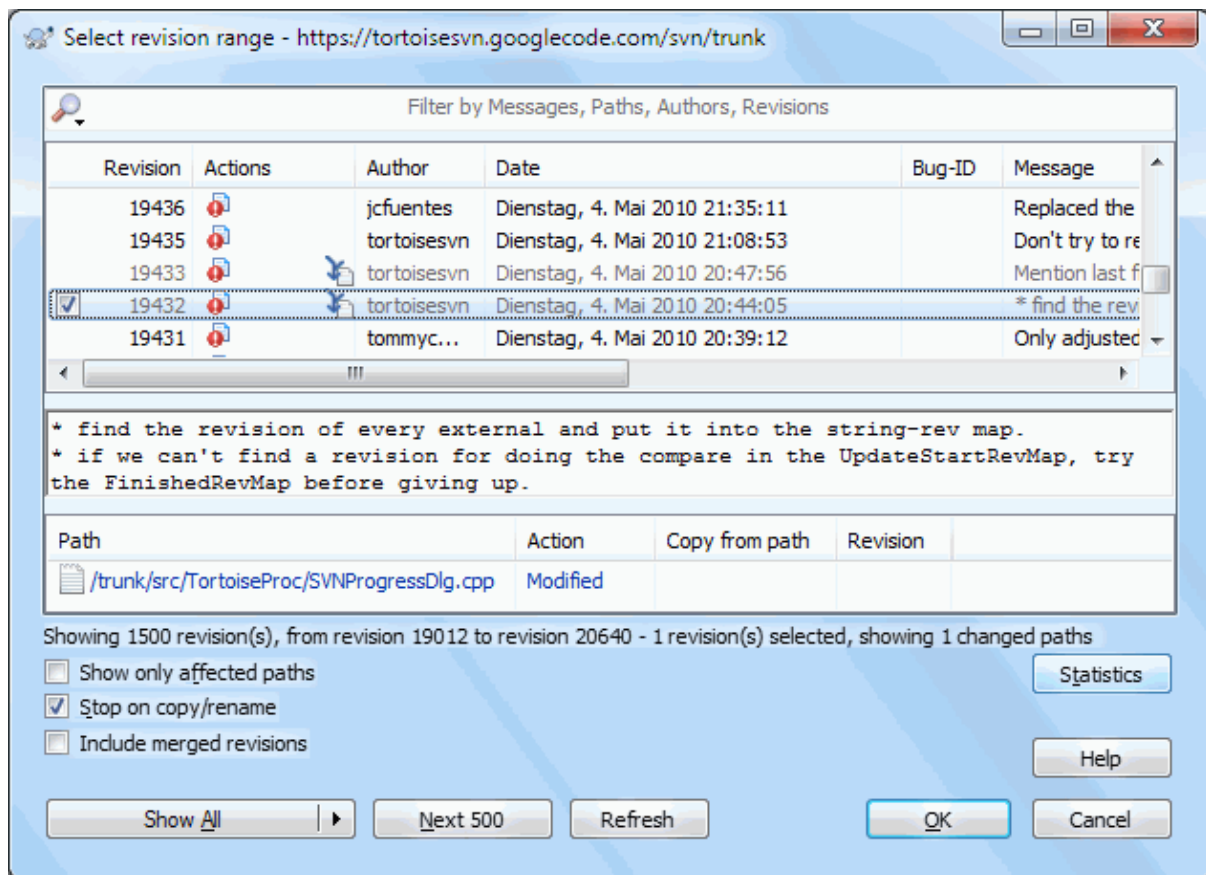
#### 4.10.5. Revisión actual de la copia de trabajo

Dado que el diálogo de registro le muestra el registro desde HEAD, no desde la revisión actual de la copia de trabajo, a veces ocurre que se muestran mensajes de registro para contenidos que aún no han sido actualizados en su copia de trabajo. Para ayudar a mostrar esto más claro, el mensaje de confirmación que corresponde a la revisión que tiene en su copia de trabajo se muestra en negrita.

Cuando se le muestra el registro de un directorio, la revisión señalada es la más alta encontrada dentro de ese directorio, lo cual requiere una operación de búsqueda de la copia de trabajo. Esta operación de búsqueda tiene lugar en un hilo separado para evitar retrasar el mostrado del registro, aunque esto puede hacer que la señalación de revisiones para directorios no aparezca inmediatamente.

#### 4.10.6. Características de registro de fusión

Subversion 1.5 y posteriores almacenan un registro de las fusiones utilizando propiedades. Esto nos permite obtener una historia más detallada de los cambios fusionados. Por ejemplo, si desarrolla una nueva característica en una rama y luego la fusiona en el tronco, la característica desarrollada aparecerá en el historial de registro del tronco como una única confirmación para la fusión, incluso aunque haya habido 1000 confirmaciones durante el desarrollo de la rama.



**Figura 4.24. El diálogo de registro mostrando revisiones con registro de fusión**

Si desea ver el detalle de qué revisiones se fusionaron como parte de esa confirmación, utilice la casilla **Incluir revisiones fusionadas**. Esto obtendrá los mensajes de registro de nuevo, pero también intercalará los mensajes de registro desde las revisiones que se fusionaron. Las revisiones fusionadas se muestran en gris porque representan cambios hechos en una parte diferente del árbol.

¡Por supuesto que fusionar nunca es sencillo! Durante el desarrollo de la rama habrá probablemente fusiones ocasionales desde el tronco para mantener la rama en sincronía con la línea principal del código. Por eso la historia de fusiones de la rama también incluirá otra capa de historia de fusiones. Estas capas se muestran en el diálogo de registro utilizando niveles de indentación.

#### 4.10.7. Cambiando el mensaje de registro y el autor

Las propiedades de revisión son completamente diferentes de las propiedades Subversion de cada ítem. Las propiedades de revisión (revprops) son ítems descriptivos que están asociados con un número de revisión en concreto en el repositorio, como el mensaje de registro, la fecha de confirmación o el nombre del confirmador (autor).

A veces querrá cambiar un mensaje de registro que introdujo en su día, quizás porque hay un error ortográfico en él o porque quiere mejorar el mensaje o cambiarlo por otras razones. O quizás quiera cambiar el autor de una confirmación porque se le olvidó preparar la autenticación, o ...

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 [Sección 3.3, “Scripts gancho en el lado del servidor”](#) to find some further notes on implementing hooks on a Windows machine.

Una vez que haya preparado su servidor con los ganchos necesarios, puede cambiar tanto el autor como el mensaje de registro (o cualquier otra propiedad de revisión) de cualquier revisión, utilizando el menú contextual del panel superior del diálogo Registro. También puede editar un mensaje de registro utilizando el menú contextual del panel del medio.



#### Aviso

Dado que las propiedades de revisión de Subversion no se versionan, al hacer modificaciones a estas propiedades (por ejemplo, la propiedad del mensaje `svn:log`) se sobrescribirá el valor anterior de esa propiedad *para siempre*.



#### 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 [Sección 4.10.11, “Refrescando la vista”](#)

#### 4.10.8. Filtrando los mensajes de registro

Si desea restringir los mensajes de registro para mostrar sólo en los que está interesado en vez de tener que navegar en una lista de cientos, puede utilizar los controles de filtro en la parte superior del Diálogo de Registro. Los controles de fecha de inicio y de fin le permite restringir la salida a un rango de fechas conocido. La caja de texto de búsqueda le permite mostrar sólo los mensajes que contengan una frase en particular.

Pulse en el ícono de búsqueda para seleccionar sobre qué información desea buscar, y para seleccionar el modo *regex*. Normalmente sólo necesitará una búsqueda de texto simple, pero si necesita utilizar términos de búsqueda más flexibles, puede utilizar expresiones regulares. Si pasa el ratón por encima de la caja, aparecerá un texto de ayuda que le proporcionará pistas sobre cómo utilizar las funciones regex y las funciones de subcadena. El filtro comprueba si su cadena de búsqueda coincide con alguna entrada de registro, y sólo entonces muestra aquellas entradas que *coinciden*.

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".

Describir el uso de expresiones regulares en búsquedas está fuera del ámbito de este manual. Puede encontrar documentarión en línea y un tutorial en <http://www.regular-expressions.info/>

Tenga en cuenta que estos filtros actúan sobre los mensajes ya obtenidos. Ellos no controlan la descarga de mensajes desde el repositorio.

También puede filtrar los nombres de las rutas en el panel inferior utilizando la casilla **Mostrar sólo las rutas relacionadas**. Las rutas relacionadas son aquellas que contienen la ruta utilizada para mostrar el registro. Si obtiene el registro de una carpeta, eso significa todo lo que esté en esa carpeta o debajo de ella. Para un archivo,

significa sólo ese archivo. Normalmente la lista de rutas muestra cualquier otra ruta relacionada con la misma confirmación, pero en gris. Si la casilla está activada, esas rutas no serán mostradas.

A veces sus prácticas de trabajo requerirán que los mensajes de registro sigan un formato particular, lo que significa que el texto que describe los cambios no es visible desde el resumen abreviado mostrado en el panel superior. La propiedad `tsvn:logsummary` puede utilizarse para extraer una porción del mensaje de registro que se mostrará en el panel superior. Lea [Sección 4.18.2, “Propiedades de proyecto TortoiseSVN”](#) para saber cómo se utiliza esta propiedad.



### Ningún formato de registro desde el navegador de repositorios

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. Información estadística

El botón Estadísticas lanza un cuadro de diálogo que muestra algunas informaciones interesantes sobre las revisiones que se muestran en el diálogo Registro. Le muestra cuántos autores han estado trabajando, cuántas confirmaciones han hecho, el progreso por semanas, y mucho más. Ahora puede ver de un vistazo quién ha trabajado duro y quién se ha tocado la barriga ;-)

### 4.10.9.1. Página de estadísticas

Esta página le proporciona todas las cifras que pueda necesitar, en particular el período y el número de revisiones cubiertas, y algunos valores mínimos/máximos/medios.

### 4.10.9.2. Página de confirmaciones por autor

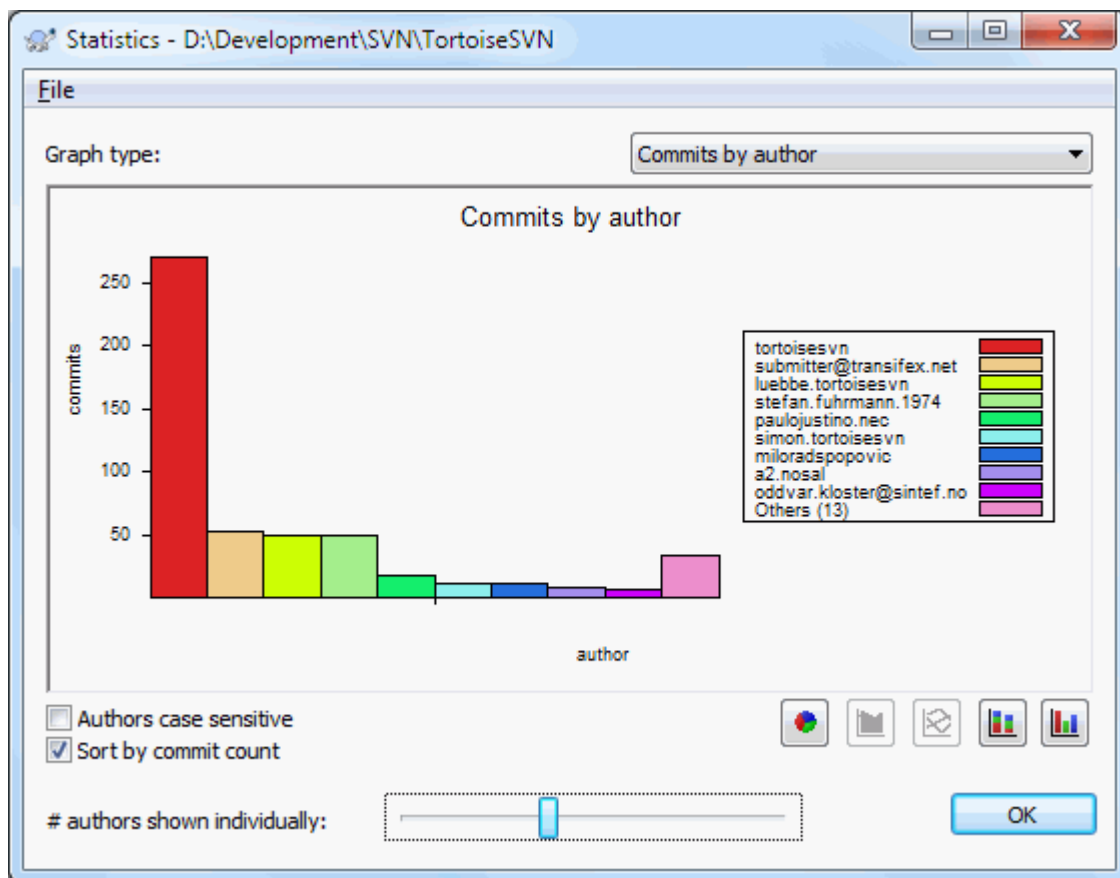
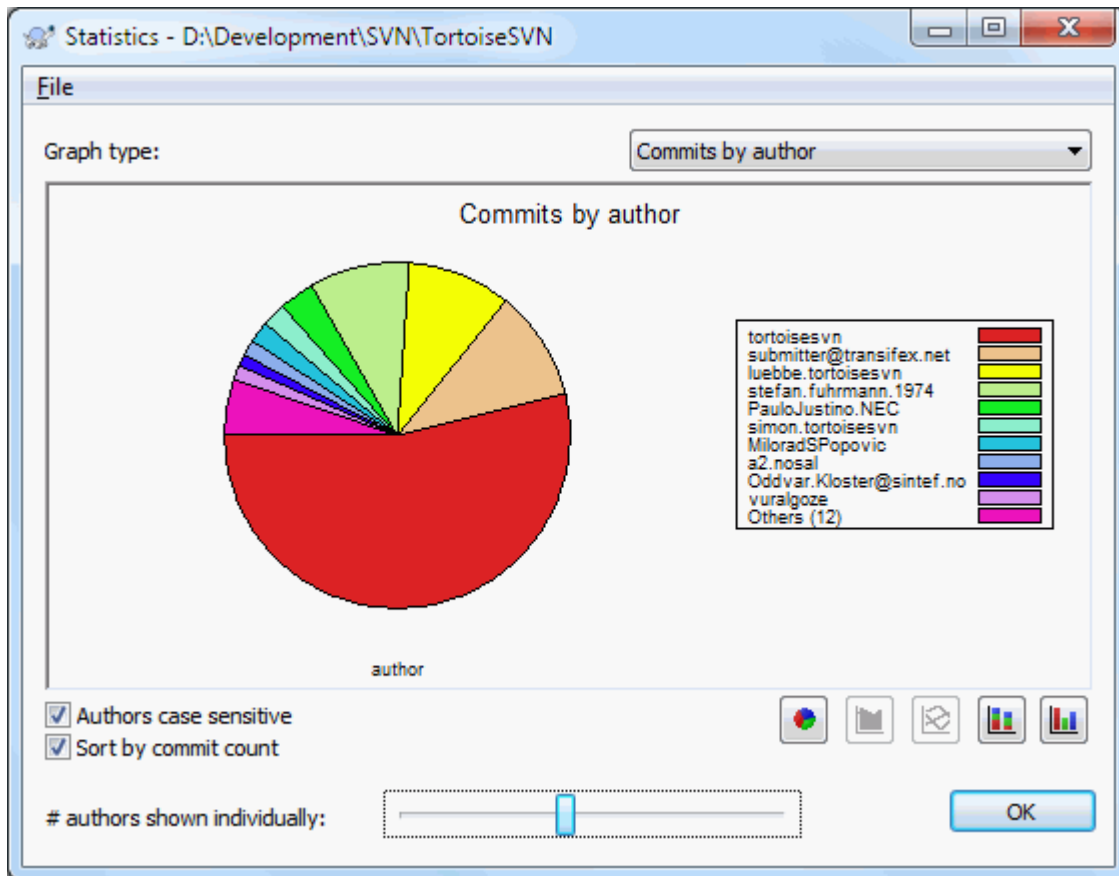


Figura 4.25. Histograma de confirmaciones por autor



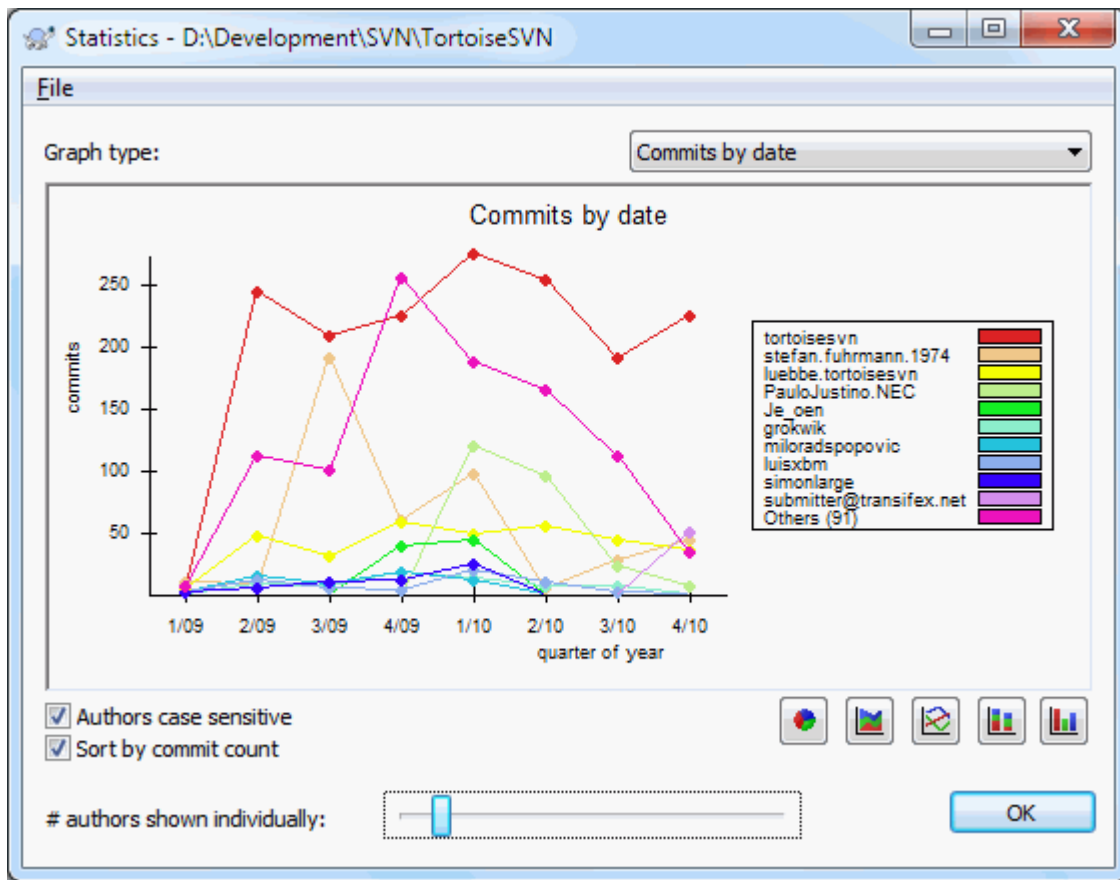
Este gráfico le muestra qué autores han estado activos en el proyecto como un simple histograma, un histograma apilado o un gráfico de tarta.



**Figura 4.26. Gráfico de tarta de confirmaciones por autor**

Cuando hay unos pocos autores muy activos y muchos pequeños contribuyentes, el número de segmentos pequeños puede hacer que el gráfico sea más difícil de leer. El selector deslizante en la parte inferior le permite establecer un límite (el porcentaje sobre el total de confirmaciones) bajo el cual cualquier actividad se agrupa en una categoría *Otros*.

### 4.10.9.3. Página de confirmaciones por fecha



**Figura 4.27. Gráfico de confirmaciones por fecha**

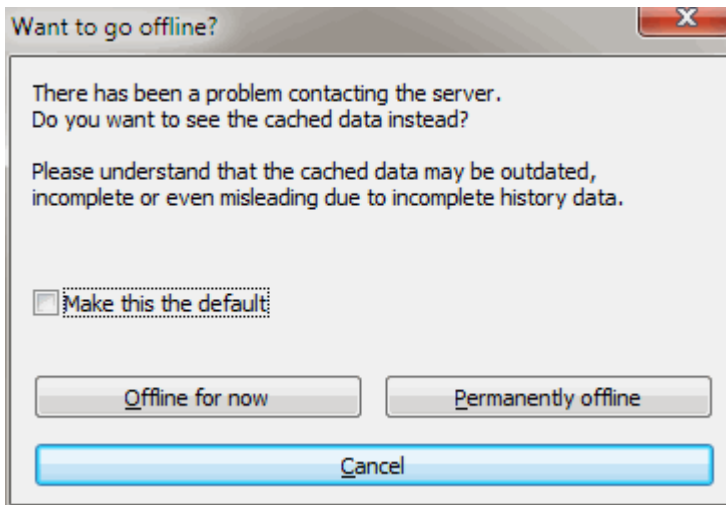
Esta página proporciona una representación gráfica de la actividad del proyecto en términos del número de confirmaciones y autor. Esto da una idea de cuándo se ha trabajado en un proyecto, y quién estaba trabajando en cada momento.

Cuando hay varios autores, puede obtener muchas líneas en el gráfico. Hay dos vistas disponibles aquí: *normal*, en la que la actividad de cada autor se refiere a la línea base, y *apilado*, donde la actividad de cada autor se refiere a la línea subyacente. La última opción evita que se crucen las líneas, lo que puede permitir un gráfico más sencillo de leer, pero es menos fácil ver la salida de cada autor.

Por defecto el análisis distingue mayúsculas y minúsculas, por lo que los usuarios PeterEgan y PeteRegan se tratan como autores diferentes. Sin embargo, en muchos casos los nombres de usuario no distinguen mayúsculas y minúsculas, y a veces se introducen de forma inconsistente, por lo que puede querer que DavidMorgan y davidmorgan se traten como la misma persona. Utilice la casilla Autores sin importar mayús/minús para controlar este comportamiento.

Tenga en cuenta que las estadísticas cubren el mismo período que el diálogo Registro. Si sólo se está mostrando una revisión, las estadísticas no le dirán mucho.

#### 4.10.10. Modo sin conexión



**Figura 4.28. Diálogo para terminar la conexión**

Si no se puede conectar al servidor y tiene habilitado el caché de registro, puede utilizar el diálogo de registro y el gráfico de revisiones en modo sin conexión. Esto utiliza la información de la caché, lo que le permite continuar trabajando aunque la información puede no estar actualizada o incluso completa.

En este caso tiene tres opciones:

Sin conexión por el momento

Completar la operación actual en modo sin conexión, pero reintentar el repositorio la próxima vez que se pida la información de registro.

Permanente sin conexión

Permanece en el modo sin conexión hasta que se pida específicamente una comprobación de repositorio. Vea [Sección 4.10.11, “Refrescando la vista”](#).

Cancelar

Si no desea continuar la operación con datos posiblemente antiguos, simplemente cancele.

La casilla de opción **Hacer este el default** evita que este diálogo reaparezca y siempre elegirá la opción que usted escoja a continuación. En cualquier momento, usted puede cambiar (o remover) el default después de llevar a cabo esta acción desde TortoiseSVN → Settings.

#### 4.10.11. Refrescando la vista

Si desea comprobar el servidor de nuevo para obtener los mensajes de registro más recientes, puede simplemente refrescar la vista utilizando **F5**. Si está utilizando la caché de registro (habilitada por defecto), esto buscará en el repositorio los mensajes más recientes y traerá sólo los nuevos. Si la caché de registro estaba en modo desconectado, esto intentará volver a ponerla en línea.

Si está utilizando la caché de registros y cree que el contenido del mensaje o el autor han podido cambiar, puede utilizar **Mays-F5** o **Ctrl-F5** para re-obtener los mensajes mostrados desde el servidor y actualizar la caché de registro. Tenga en cuenta que esto sólo afecta a los mensajes que se están mostrando actualmente y que por tanto no invalida la caché completa de ese repositorio.

### 4.11. Viendo diferencias

Uno de los requisitos más comunes en el desarrollo de proyectos es ver qué ha cambiado. Puede querer ver las diferencias entre dos revisiones del mismo archivo, o las diferencias entre dos archivos separados. TortoiseSVN

provee una herramienta integrada llamada TortoiseMerge para ver las diferencias entre archivos de texto. Para ver las diferencias entre archivos de imagen, TortoiseSVN también tiene una herramienta llamada TortoiseIDiff. Por supuesto, puede utilizar su herramienta de diferencias favorita si lo desea.

### 4.11.1. Diferencias de archivos

#### Cambios locales

Si desea ver qué cambios ha hecho *usted* en su copia de trabajo, simplemente utilice el menú contextual del explorador y seleccione TortoiseSVN → Diferenciar.

#### Diferenciar con otra rama/etiqueta

Si desea ver qué ha cambiado en el tronco (si está trabajando en una rama) o en una rama específica (si está trabajando en el tronco), puede utilizar el menú contextual del explorador. Simplemente sostenga la tecla **Mayúsculas** mientras hace click con el botón derecho en el archivo. Luego, seleccione TortoiseSVN → Diferenciar con URL. En el siguiente diálogo, especifique la URL del repositorio con la que quiere comparar su archivo local.

También puede utilizar el navegador de repositorios y seleccionar dos árboles para diferenciar, quizás dos ramas, o una rama/etiqueta y el tronco. Ahí, el menú contextual le permite compararlos utilizando Comparar revisiones. Lea más en [Sección 4.11.3, “Comparando carpetas”](#).

#### Diferenciar desde una revisión anterior

Si desea ver las diferencias entre una revisión en concreto y su copia de trabajo, utilice el diálogo Registro de Revisiones, seleccione la revisión de interés, y luego seleccione Comparar con la copia de trabajo desde el menú contextual.

Si desea ver la diferencia entre la última revisión confirmada y su copia de trabajo, asumiendo que la copia de trabajo no se haya modificado, simplemente haga click con el botón derecho sobre el archivo. Luego, seleccione TortoiseSVN → Comparar con la revisión anterior. Esto realizará una diferenciación entre la revisión anterior a la fecha-de-la-última-confirmación (tal y como se registró en su copia de trabajo) y la BASE de la copia de trabajo. Esto le muestra el último cambio hecho al archivo que lo llevó al estado que ahora está viendo en su copia de trabajo. No mostrará cambios más nuevos que su copia de trabajo.

#### Diferenciar entre dos revisiones antiguas

Si desea ver las diferencias entre dos revisiones que ya se confirmaron, utilice el diálogo Registro de Revisiones y seleccione las dos revisiones que desea comparar (utilizando el modificador habitual **Ctrl**). Luego seleccione Comparar revisiones desde el menú contextual.

Si hizo esto desde el historial de revisiones de una carpeta, aparece un diálogo Comparar Revisiones, mostrando una lista de archivos cambiados en esa carpeta. Lea más en [Sección 4.11.3, “Comparando carpetas”](#).

#### Todos los cambios hechos en una confirmación

Si desea ver los cambios hechos a todos los archivos en una revisión en particular de una vez, puede utilizar la salida diff unificado (formato de parche GNU). Esto le muestra sólo las diferencias con unas pocas líneas de contexto. Es más difícil de leer que una comparación visual de archivos, pero le mostrará todos los cambios juntos. Desde el diálogo Registro de Revisiones, seleccione la revisión de interés, y luego seleccione Mostrar Diferencias como Diff Unificado desde el menú contextual.

#### Diferencias entre archivos

Si desea ver las diferencias entre dos archivos diferentes, puede hacerlo directamente en el explorador seleccionando ambos archivos (utilizando el modificador habitual **Ctrl**). Luego desde el menú contextual del explorador seleccione TortoiseSVN → Diferenciar.

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.

#### Diferencias entre un archivo/carpeta en la copia de trabajo y una URL

Si desea ver las diferencias entre un archivo en su copia de trabajo, y un archivo en cualquier repositorio de Subversion, puede hacerlo directamente en el explorador seleccionando el archivo y pulsando la tecla **Mayúsculas** mientras hace click con el botón derecho para obtener el menú contextual. Seleccione TortoiseSVN → Diferenciar con URL. Puede hacer lo mismo para una carpeta de copia de trabajo. TortoiseMerge muestra esas diferencias de la misma forma que muestra un archivo de parche - una lista de archivos cambiados que puede ver de uno en uno.

#### Diferencias con información de autoría

Si desea ver no sólo las diferencias sino también el autor, revisión y la fecha en la que se hicieron los cambios, puede combinar los informes de diferencias y de autoría desde dentro del diálogo del historial de revisiones. Lea [Sección 4.24.2, "Autoría de las diferencias"](#) para más detalles.

#### Diferencias entre carpetas

Las herramientas proporcionadas con TortoiseSVN no le permiten ver las diferencias entre jerarquías de directorios. Pero si tiene una herramienta externa que soporte esta funcionalidad, puede utilizarla. En [Sección 4.11.6, "Herramientas externas de diferencias/fusión"](#) le contamos algunas de las herramientas que hemos utilizado.

Si ha configurado una herramienta de diferenciado de terceros, puede utilizar **Mayúsculas** cuando seleccione el comando Diferenciar para utilizar la herramienta alternativa. Lea [Sección 4.31.5, "Configuración de programas externos"](#) para saber más sobre configurar otras herramientas de diferenciación.

### 4.11.2. Opciones de fin de línea y espacios en blanco

A veces en la vida de un proyecto puede querer cambiar los fines de línea de CRLF a LF, o puede querer cambiar la indentación de una sección. Desafortunadamente esto marcará un gran número de líneas como modificadas, incluso aunque no haya cambios en el significado del código. Estas opciones le ayudarán a administrar estos cambios cuando se comparan y aplican diferencias. Verá estas opciones en los diálogos Fusionar y Autoría, además de en la configuración de TortoiseMerge.

Ignorar finales de línea excluye los cambios que sólo se deban a diferencias en el estilo de fin de línea.

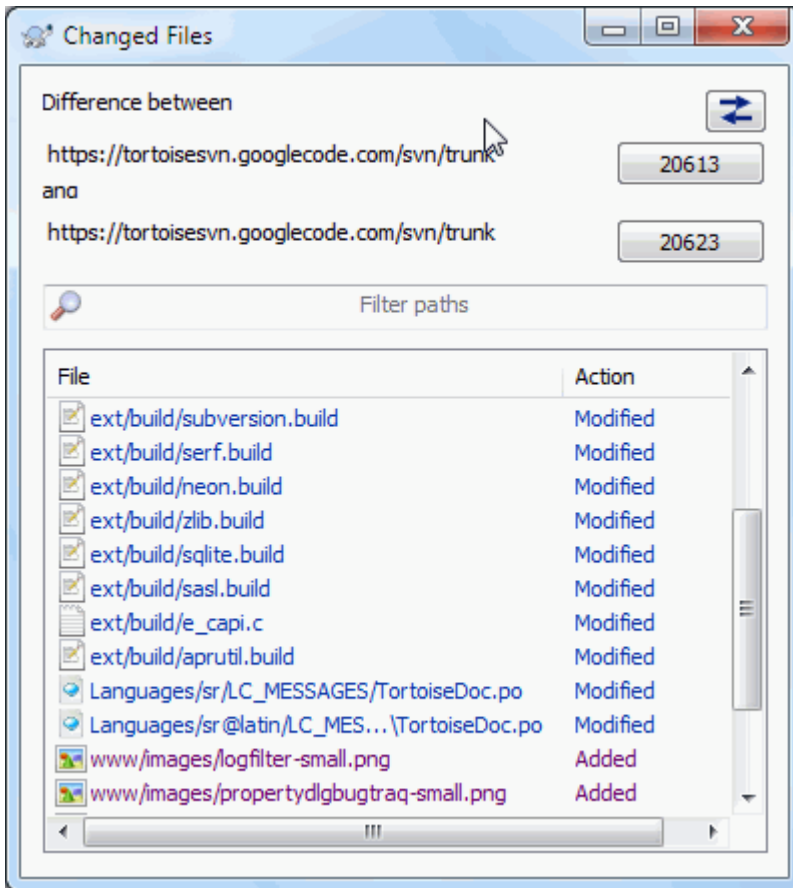
Comparar espacios en blanco incluye todos los cambios en la indentación y en los espacios en blanco interiores como líneas añadidas/eliminadas.

Ignorar cambios en espacios en blanco excluye los cambios que sólo se deben a un cambio en la cantidad o tipo de espacios en blanco, como por ejemplo cambios en la indentación o cambio de tabuladores a espacios. Añadir espacios en blanco donde antes no había, o eliminar completamente espacios en blanco aún se mostrará como un cambio.

Ignorar todos los espacios en blanco excluye todos los cambios que sólo se deban a espacios en blanco.

Naturalmente, cualquier línea cuyo contenido haya cambiado se incluye siempre en la diferenciación.

### 4.11.3. Comparando carpetas



**Figura 4.29. El diálogo Comparar Revisiones**

Cuando seleccione dos árboles dentro del navegador de repositorios, o cuando seleccione dos revisiones de una carpeta en el diálogo de registro, puede Menú contextual → Comparar revisiones.

Este diálogo muestra una lista de todos los archivos que han cambiado y le permite comparar o ver la autoría de cada uno individualmente utilizando el menú contextual.

Puede exportar un *árbol de cambios*, lo que es útil si necesita enviar a alguien la estructura de su árbol de proyecto, pero conteniendo sólo los archivos que han cambiado. Esta operación trabaja sólo sobre los archivos seleccionados, por lo que necesitará seleccionar los archivos de interés - normalmente eso significa todos - y luego Menú contextual → Exportar selección a.... Se le preguntará por una ruta donde guardar el árbol de cambios.

También puede exportar la *lista* de archivos cambiados a un archivo de texto utilizando Menú contextual → Guardar lista de archivos seleccionados a....

Si desea exportar la lista de archivos y *también* las acciones (modificado, añadido, borrado), puede hacerlo utilizando Menú contextual → Copiar la selección al portapapeles.

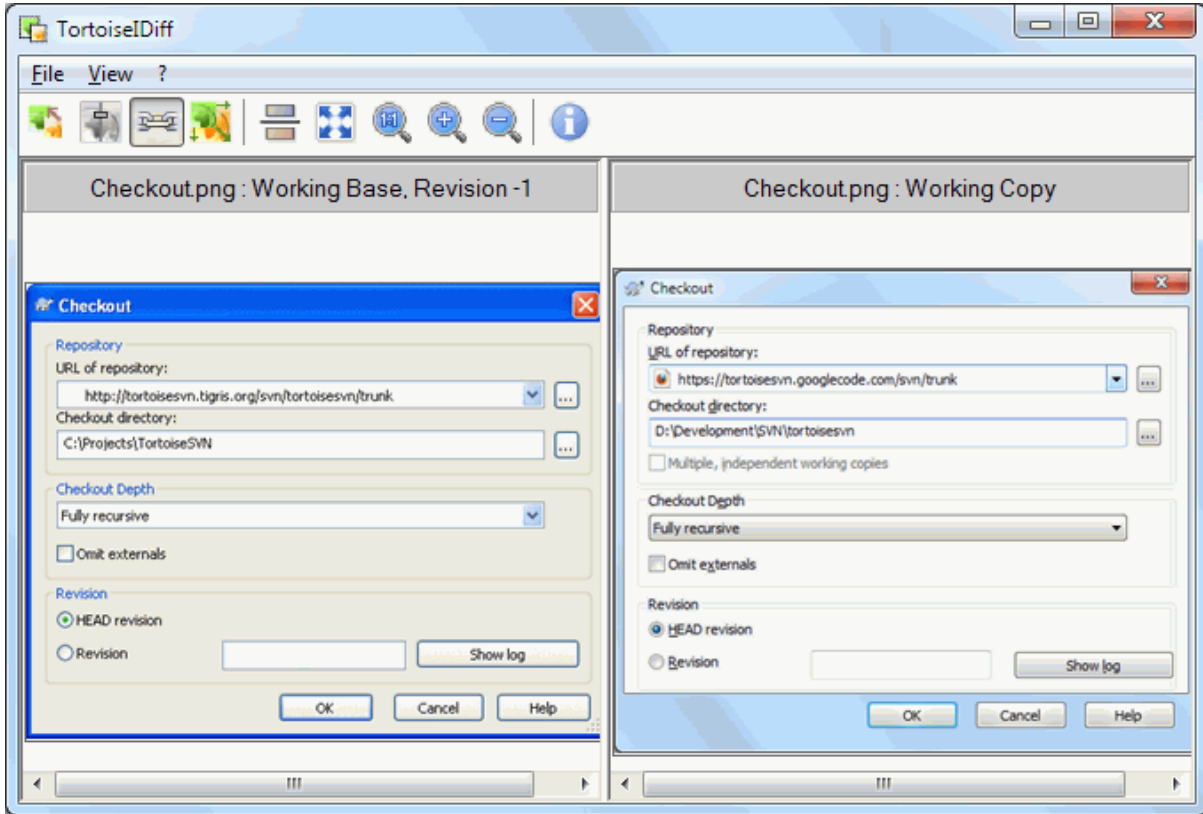
El botón en la parte superior le permite cambiar la dirección de la comparación. Puede mostrar los cambios necesarios para ir de A a B, o si lo prefiere, de B a A.

Los botones con los números de revisión pueden utilizarse para cambiar a un rango de revisiones diferente. Cuando cambia el rango, la lista de ítems que difieren entre las dos revisiones se actualizará automáticamente.

Si la lista de nombres de archivos es muy larga, puede utilizar la caja de búsqueda para reducir la lista a los nombres de archivos que contengan un texto específico. Tenga en cuenta que se utiliza una búsqueda de texto simple, así que si desea restringir la lista a archivos de código fuente C debería introducir `.c` en vez de `*.c`.

#### 4.11.4. Diferenciando imágenes utilizando TortoiseIDiff

Hay muchas herramientas disponibles para diferenciar archivos de texto, incluyendo nuestro propio TortoiseMerge, pero a veces nos encontramos en la situación de querer ver también cómo ha cambiado un archivo de imagen. Por eso es por lo que hemos creado TortoiseIDiff.



**Figura 4.30. El visor de diferencias de imágenes**

TortoiseSVN → Diff para cualquiera de los formatos de imágenes comunes iniciará TortoiseIDiff para mostrar las diferencias entre las imágenes. Por defecto las imágenes se muestran una al lado de la otra, pero puede utilizar el menú Ver o la barra de herramientas para cambiar a una vista arriba-abajo, o si lo prefiere, puede superponer las imágenes y simular que está usando una caja de luces.

Naturalmente, también puede acercar y alejar el zoom y moverse por la imagen. También puede desplazarse por la imagen simplemente arrastrando con el botón izquierdo. Si selecciona la opción **Unir las imágenes**, los controles de desplazamiento (barras de desplazamiento, rueda del ratón) de ambas imágenes quedan unidos.

Un cuadro de información de imagen le muestra detalles sobre el archivo de imagen, tales como el tamaño en píxeles, la resolución y la profundidad de color. Si la caja le molesta, utilice **Ver → Información de imagen** para ocultarla. Puede obtener la misma información en una etiqueta de ayuda si pasa el cursor por encima de la barra de título de la imagen.

Cuando las imágenes se superponen, la intensidad relativa de las imágenes (alpha blend) se controla mediante un control deslizante en el lado izquierdo. Puede pulsar en cualquier parte del deslizador para establecer el blend directamente, o puede arrastrar el deslizador para cambiar el blend de forma interactiva. **Ctrl+Mayúsculas**-Rueda para cambiar el blend.

El botón debajo del deslizador cambia entre fusión 0% y 100%, y si hace doble click en el botón, la fusión cambia automáticamente cada segundo hasta que pulse el botón de nuevo. Esto puede ser útil cuando busca múltiples cambios pequeños.

A veces quiere ver una diferencia en vez de un fundido. Puede que tenga las imágenes de dos revisiones de una placa de circuito impreso y desea ver qué pistas han cambiado. Si deshabilita el modo alpha blend, la diferencia se mostrará como un *XOR* de los valores de color del pixel. Las áreas sin cambios se mostrarán en blanco y los cambios se colorearán.

#### 4.11.5. Diferenciando los documentos de Office

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



#### Problemas con Office 2010

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

#### 4.11.6. Herramientas externas de diferencias/fusión

Si las herramientas que le proporcionamos no son lo que necesita, pruebe alguno de los muchos programas de código abierto o comerciales disponibles. Cada uno tiene sus propios favoritos, y esta lista por supuesto no está completa, pero aquí hay algunas que debería considerar:

##### WinMerge

[WinMerge](https://winmerge.sourceforge.net/) [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) [https://www.perforce.com/perforce/products/merge.html].

##### KDiff3

KDiff3 es una herramienta de diferencias gratuita que también maneja directorios. Puede descargarla desde [aquí](http://kdiff3.sf.net/) [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/) [https://www.sourcegear.com/diffmerge/].

##### ExamDiff

ExamDiff Standard es gratuito. Puede manejar archivos pero no directorios. ExamDiff Pro es shareware y añade un número de ventajas que incluyen diferenciación de directorios y capacidades de edición. En ambos casos, la versión 3.2 y las siguientes pueden manejar también unicode. Puede descargarlas desde [PrestoSoft](http://www.prestosoft.com/) [http://www.prestosoft.com/].

##### Beyond Compare

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

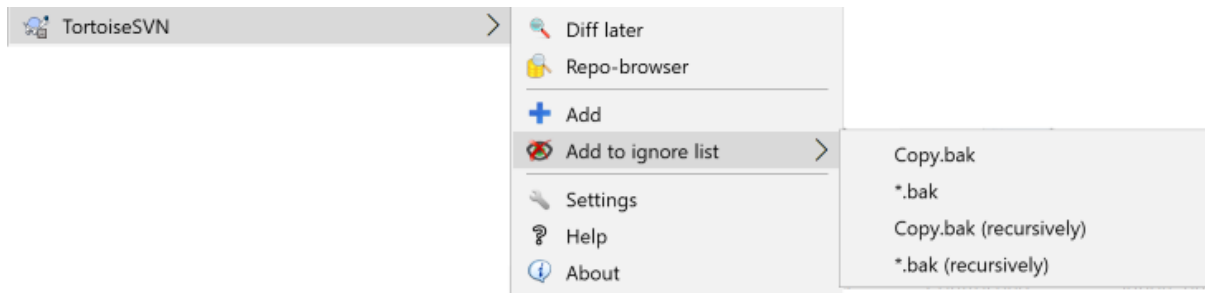
##### Araxis Merge

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

Lea [Sección 4.31.5, “Configuración de programas externos”](#) para más información sobre cómo preparar TortoiseSVN para utilizar estas herramientas.



## 4.12. Añadiendo nuevos archivos y directorios



**Figura 4.31. Menú contextual del explorador para archivos no versionados**

Si ha creado nuevos archivos y/o directorios durante su proceso de desarrollo, necesitará añadirlos también al control de código. Seleccione los archivos y/o directorios y utilice TortoiseSVN → Añadir.

Después que añada los archivos o directorios al control de código, el archivo aparece con un ícono superpuesto añadido que significa que primero debe confirmar su copia de trabajo para que esos archivos y directorios estén disponibles para otros desarrolladores. ¡Añadir un archivo/directorio *no* afecta al repositorio!



### Añadir muchos

También puede usar el comando Añadir en carpetas que ya estén versionadas. En ese caso, el diálogo de añadir le mostrará todos los archivos sin versionar dentro de esa carpeta versionada. Esto ayuda si tiene muchos archivos nuevos y necesita añadirlos todos de golpe.

Para añadir archivos desde fuera de su copia de trabajo puede usar el manejador de arrastrar-y-soltar:

1. seleccione los archivos que desea añadir
2. arrástrelos con el botón derecho a su nuevo destino dentro de la copia de trabajo
3. suelte el botón derecho del ratón
4. seleccione Menú contextual → SVN Añadir archivos a esta Copia de Trabajo. En ese momento los archivos se copiarán a la copia de trabajo y se añadirán al control de versiones.

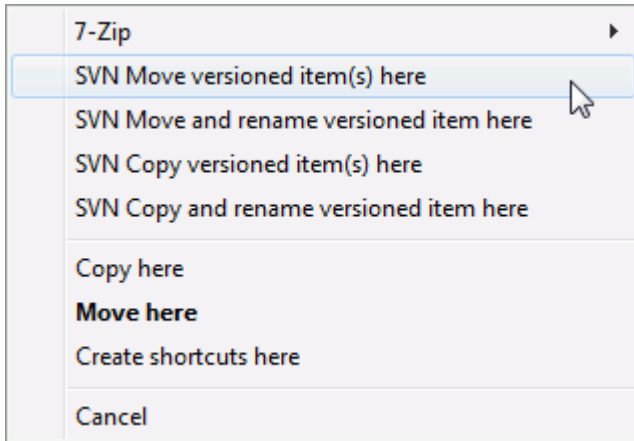
También puede añadir archivos dentro de una copia de trabajo simplemente arrastrándolos con el botón izquierdo y soltándolos en el diálogo de confirmación.

Si añade un archivo o carpeta por error, puede deshacer la adición antes de confirmar utilizando TortoiseSVN → Deshacer añadir....

## 4.13. Copiando/Moviendo/Renombrando archivos y carpetas

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.

La manera más fácil de copiar archivos y carpetas desde dentro de una copia de trabajo es utilizar el menú contextual que aparece al arrastrar con el botón derecho. Cuando arrastra con el botón derecho un archivo o una carpeta de una copia de trabajo a otra, o incluso dentro de la misma carpeta, aparece un menú contextual cuando suelta el botón del ratón.



**Figura 4.32. Menú de arrastre con el botón derecho para un directorio bajo el control de versiones**

Ahora puede copiar contenido existente versionado a un nuevo lugar, posiblemente renombrándolo al mismo tiempo.

También puede copiar o mover archivos versionados dentro de una copia de trabajo, o entre dos copias de trabajo, utilizando el método familiar corta-y-pegar. Utilice el método estándar de Windows Copiar o Cortar para copiar uno o más ítems versionados al portapapeles. Si el portapapeles contiene ítems versionados, puede utilizar TortoiseSVN → Pegar (tenga en cuenta: no es el método estándar de Windows Pegar) para copiar o mover estos ítems al nuevo lugar de la copia de trabajo.

Puede copiar archivos y carpetas desde su copia de trabajo a otro lugar del repositorio utilizando TortoiseSVN → Rama/Etiqueta. Para saber más, vea [Sección 4.20.1, “Crand una rama o etiqueta”](#).

Puede localizar una versión anterior de un archivo o carpeta en el diálogo de registro y copiarlo a un nuevo lugar del repositorio directamente desde el diálogo de registro utilizando Menú contextual → Crear rama/etiqueta desde la revisión. Para saber más, vea [Sección 4.10.3, “Obteniendo información adicional”](#).

También puede utilizar el navegador de repositorio para localizar el contenido que quiere, y copiarlo en su copia de trabajo directamente desde el repositorio, o copiar entre dos lugares dentro del repositorio. Para más información lea [Sección 4.25, “El navegador de repositorios”](#).

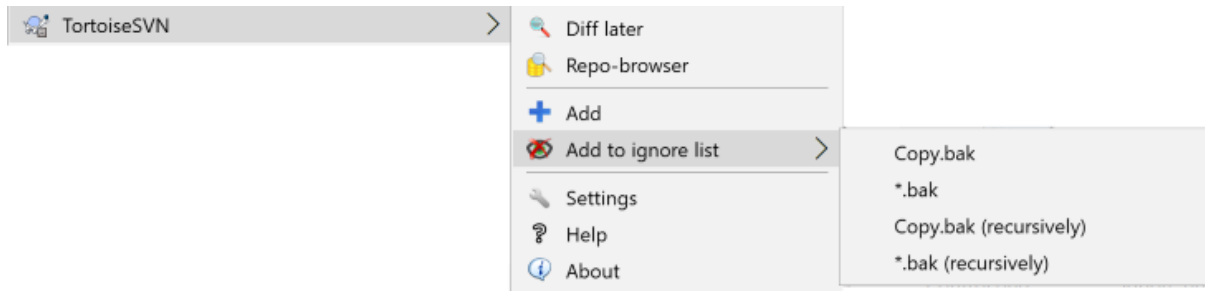


### **No se puede copiar entre repositorios**

Mientras que puede copiar y mover archivos y carpetas *dentro* de un repositorio, usted *no puede* copiar o mover de un repositorio a otro y preservar la historia simultáneamente, utilizando TortoiseSVN. Ni siquiera aún y cuando los repositorios residan en el mismo servidor. Lo único que usted puede hacer, es copiar el contenido en su estado actual y añadirlo a el segundo repositorio como contenido nuevo .

Si no está seguro de si dos URLs en el mismo servidor se refieren al mismo repositorio o a repositorios distintos, utilice el navegador de repositorios para abrir una de las URLs y averiguar dónde está la raíz del repositorio. Si puede ver ambos lugares en una única ventana del navegador de repositorios, entonces las dos rutas están en el mismo repositorio.

## **4.14. Ignorando archivos y directorios**



**Figura 4.33. Menú contextual del explorador para archivos no versionados**

En la mayoría de los proyectos tendrá archivos y carpetas que no deberán estar sujetos al control de versiones. Esto puede incluir archivos creados por el compilador, `*.obj`, `*.lst`, o quizás una carpeta de salida donde se almacenan los ejecutables. Cuando confirma los cambios, TortoiseSVN le muestra los archivos no versionados, que rellenan la lista de archivos del diálogo de confirmar. Por supuesto que puede desactivarlos para que no se muestren, pero entonces quizás se olvide de añadir un nuevo archivo de código fuente.

La mejor forma de evitar estos problemas es añadir los archivos derivados a la lista de ignorados del proyecto. De esta forma nunca se mostrarán en el diálogo de confirmar, pero se seguirán señalando los genuinos archivos de código no versionados.

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.

Si desea eliminar uno o más ítems de la lista de ignorados, haga click con el botón derecho en dichos ítems y seleccione TortoiseSVN → Eliminar de la Lista de Ignorados También puede acceder a la propiedad `svn:ignore` de la carpeta directamente. Esto le permitirá especificar patrones más generales utilizando expansión de comodines en los nombres de los archivos, descrito más adelante. Para más información sobre establecer propiedades directamente, lea [Sección 4.18, “Configuración del proyecto”](#). Por favor, tenga en cuenta que cada patrón de ignorar debe escribirse en una nueva línea. No funcionará si los separa por espacios.



### La lista global de ignorados

Otra forma de ignorar archivos es añadirlos a la *lista global de ignorados*. La mayor diferencia aquí es que la lista global de ignorados es una propiedad del cliente. Se aplica a *todos* los proyectos de Subversion, pero sólo en el PC cliente. En general es mejor utilizar la propiedad `svn:ignore` donde sea posible, porque puede aplicarse a áreas del proyecto específicas, y funciona para todos los que obtengan el proyecto. Para más información, lea la [Sección 4.31.1, “Configuración general”](#).



### Ignorando ítems versionados

Los archivos y las carpetas versionadas nunca pueden ser ignoradas - esta es una característica de Subversion. Si ha versionado un archivo por error, lea [Sección B.8, “Ignorar archivos que ya están versionados”](#) para ver las instrucciones sobre cómo “desversionarlo”.

#### 4.14.1. Concordancia de patrones en las listas de ignorados

El patrón de ignorados de Subversion hace uso de la expansión de comodines en los nombres de los archivos, una técnica que originariamente se utilizaba en Unix para especificar archivos utilizando meta-caracteres como son los comodines. Los siguientes caracteres tienen un significado especial:

\*

Concuerta con cualquier cadena de caracteres, incluyendo la cadena vacía (sin caracteres).

?

Concuerta con un único carácter cualquiera.

[...]

Concuerta con cualquiera de los caracteres incluidos entre los corchetes. Dentro de los corchetes, un par de caracteres separados por “-” concuerda con cualquier caracter existente entre los dos en el orden lexicográfico. Por ejemplo, [AGm-p] concuerda con cualquiera de estos: A, G, m, n, o o p.

La concordancia de patrones distingue entre mayúsculas y minúsculas, lo que puede causar problemas en Windows. Puede forzar que no se distingan a lo bruto, emparejando los caracteres; es decir, por ejemplo, para ignorar \*.tmp sin tener en cuenta las mayúsculas y minúsculas, puede utilizar un patrón como \*. [Tt] [Mm] [Pp].

Si desea una definición oficial de la expansión de comodines en los nombres de archivos, puede encontrarla en las especificaciones de la IEEE para el lenguaje de comandos del shell *Notación de Patrones de Concordancia* [[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)].

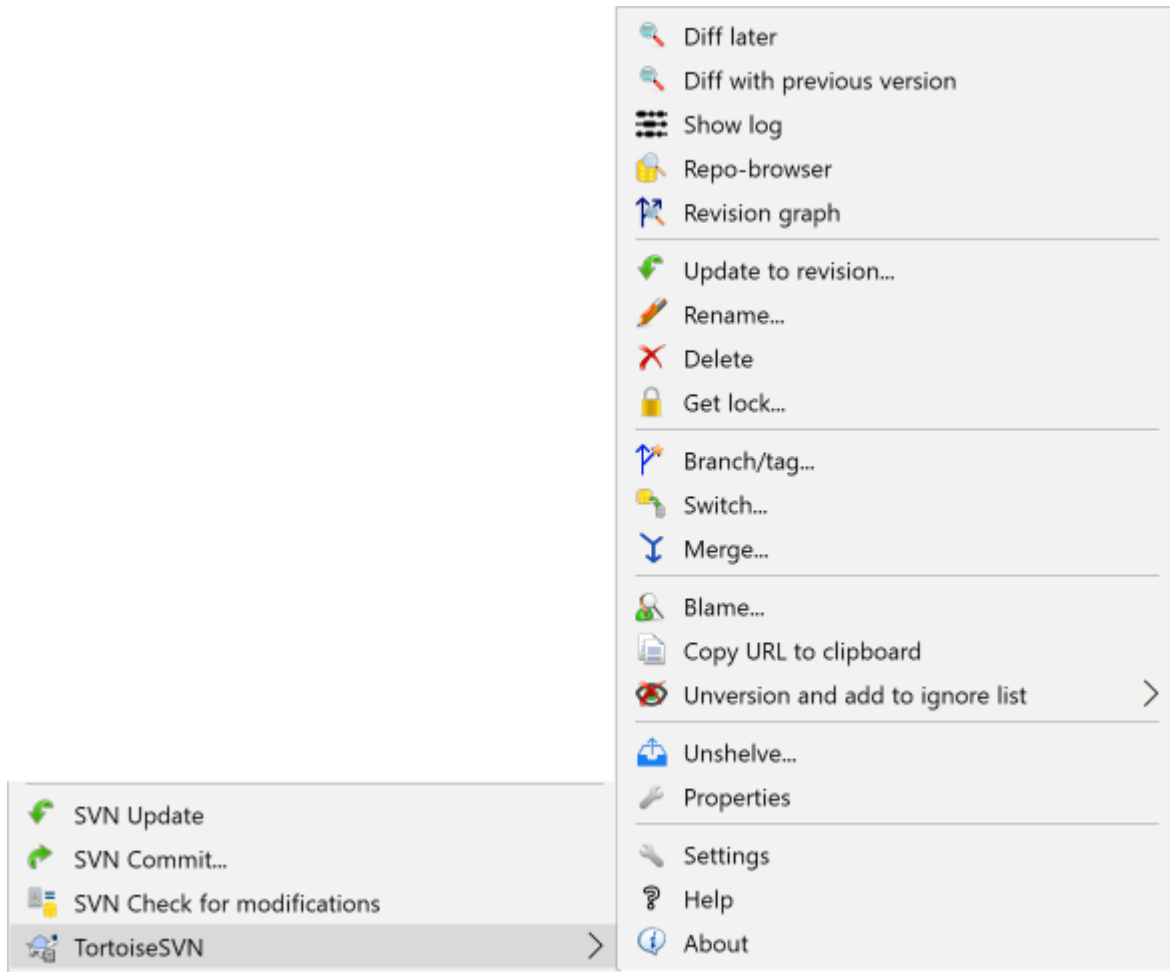


### Sin rutas en la lista global de ignorados

No debería incluir información de ruta en su patrón. La concordancia de patrones está pensada para usarse contra nombres de archivo planos y nombres de carpeta. Si desea ignorar todas las carpetas CVS, simplemente añada CVS a la lista de ignorados. No hay necesidad de especificar CVS \*/CVS como se hacía en versiones anteriores. Si desea ignorar todas las carpetas tmp que existen dentro de una carpeta prog y no dentro de una carpeta doc debería utilizar la propiedad svn:ignore. No hay una forma fiable de conseguir esto utilizando patrones globales de ignorados.

## 4.15. Eliminando, moviendo y renombrando

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



**Figura 4.34. Menú contextual del explorador para archivos versionados**

### 4.15.1. Eliminando archivos y carpetas

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

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

Si desea eliminar un ítem del repositorio, pero mantenerlo localmente como un archivo o carpeta no versionados, utilice Menú contextual extendido → Eliminar (mantener local). Debe mantener pulsada la tecla **Mayús** mientras hace click con el botón derecho sobre el ítem en el panel de lista de exploración (panel derecho) para ver esto en el menú contextual extendido.

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 un archivo o una carpeta borrados

Si ha borrado un archivo o una carpeta y ya ha confirmado esa operación de borrado en el repositorio, entonces un comando TortoiseSVN → Revertir normal no los podrá recuperar. Pero el archivo o la carpeta borrados no están perdidos para nada. Si sabe la revisión en la que se borraron el archivo o la carpeta (si no lo sabe, utilice el diálogo de registro para averiguarlo), abra el navegador de repositorios y cambie a esa revisión. Luego seleccione el archivo o la carpeta que ha borrado, haga click con el botón derecho y seleccione Menú Contextual → Copiar a..., y como destino de esa operación de copia seleccione la ruta de su copia de trabajo.

### 4.15.2. Moviendo archivos y carpetas

Si desea simplemente renombrar en el sitio un archivo o una carpeta, utilice Menú contextual → Renombrar... Introduzca el nuevo nombre para el ítem, y ya está.

Si quiere mover archivos dentro de una copia de trabajo, quizás a una subcarpeta distinta, utilice el manejador de arrastrar-y-soltar con el botón derecho del ratón:

1. seleccione los archivos o directorios que desea mover
2. arrástrelos con el botón derecho a su nuevo destino dentro de la copia de trabajo
3. suelte el botón derecho del ratón
4. en el menú contextual seleccione Menú contextual → SVN Mover archivos versionados aquí



## Confirmar la carpeta padre

Dado que las operaciones mover y renombrar se realizan como un borrado seguido de un añadir, debe confirmar la carpeta padre de los archivos movidos/renombrados para que se muestre la parte de borrado de la operación en el diálogo de confirmación. Si no confirma la parte eliminada de los archivos movidos/renombrados, se quedarán en el repositorio, y cuando sus compañeros se actualicen no se eliminará el archivo antiguo, es decir, que ellos tendrán *ambas* copias, la antigua y la nueva.

Usted *debe* confirmar un cambio de nombre de carpeta antes de cambiar cualquier archivo dentro de la carpeta; si no, su copia de trabajo realmente puede quedar estropeada.

Otra forma de mover o copiar archivos es usar los comandos copiar/cortar de Windows. Seleccione los archivos que quiere copiar, pulse el botón derecho del ratón y elija Menú Contextual → Copiar del menú contextual del explorador. Vaya al directorio objetivo, pulse el botón derecho del ratón y elija TortoiseSVN → Pegar. Para mover archivos, elija Menú Contextual → Cortar en vez de Menú Contextual → Copiar.

También puede utilizar el navegador de repositorios para mover ítems. Lea [Sección 4.25, “El navegador de repositorios”](#) para saber más.



## No SVN Mueva externos

*NO* debería utilizar los comandos TortoiseSVN Mover o Renombrar en una carpeta que ha sido creada utilizando `svn:externals`. Esta acción puede provocar que el ítem externo se elimine de su repositorio padre, probablemente molestando a muchas otras personas. Si necesita mover una carpeta externa, debería moverla como lo hace con el resto de archivos sin versionar, y luego ajustar las propiedades `svn:externals` de las carpetas padres origen y destino.

### 4.15.3. Lidiando con conflictos en las mayúsculas/minúsculas de un nombre de archivo

En el caso que tenga dos archivos en el repositorio con el mismo nombre pero diferenciándose únicamente en las mayúsculas (por ejemplo, TEST.TXT y test.txt), ya no podrá actualizar u obtener el directorio padre en un cliente de Windows. Aunque Subversion soporta los nombres de archivos que se distinguen en las mayúsculas, Windows no.

Esto ocurre a veces cuando dos personas confirman, desde copias de trabajo separadas, archivos tales que tienen el mismo nombre pero con las mayúsculas/minúsculas diferentes. Esto también puede ocurrir cuando se confirman los archivos desde un sistema con un sistema de archivos que distingue mayúsculas/minúsculas, como Linux.

En ese caso, necesita decidir cuál desea mantener y borrar (o renombrar) el otro en el repositorio.



#### Evitando dos archivos con el mismo nombre

Hay un script de servidor disponible en: <https://svn.apache.org/repos/asf/subversion/trunk/contrib/hook-scripts/> que impide subidas resultantes en caso de conflicto.

### 4.15.4. Reparando renombrados de archivos

A veces su amistoso IDE renombrará archivos por usted como parte de un ejercicio de refactorización, y por supuesto no se lo dirá a Subversion. Si intenta confirmar sus cambios, Subversion verá el nombre de archivo antiguo como faltante y el nuevo nombre como archivo no versionado. Podría simplemente confirmar el nuevo nombre de archivo para añadirlo, pero entonces perdería la traza de la historia, ya que Subversion no sabe la relación entre los archivos.

Una forma mejor es notificar a Subversion que el cambio es realmente un renombrado, y puede hacer esto dentro de los diálogos Confirmar y Comprobar modificaciones. Simplemente seleccione tanto el nombre de archivo antiguo (faltante) como el nuevo nombre (no versionado) y utilice Menú contextual → Reparar movimiento para emparejar los dos archivos como un renombrado.

### 4.15.5. Eliminando archivos no versionados

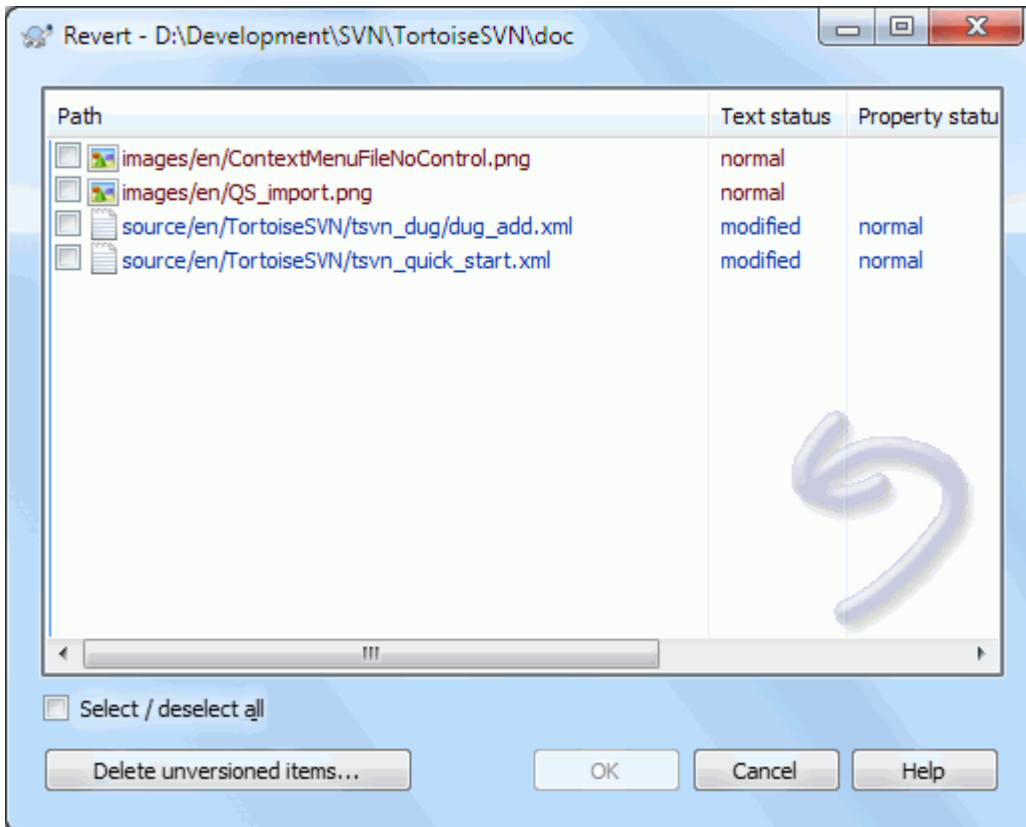
Normalmente especifica un la lista de ignorados tal que todos los archivos generados se ignoran en Subversion. Pero ¿y si lo que quiere es limpiar todos esos ítems ignorados para producir una compilación limpia? Normalmente lo haría en su makefile (archivo de instrucciones de compilación), pero si está depurando el makefile, o cambiando el sistema de compilación, es útil tener una forma de barrer la casa.

TortoiseSVN proporciona exactamente esa opción utilizando Menú contextual extendido → Eliminar ítems no versionados.... Debe mantener pulsada la tecla **Mayús** mientras hace click con el botón derecho sobre el ítem en el panel de lista de exploración (panel derecho) para ver esto en el menú contextual extendido. Esto mostrará un diálogo que lista todos los archivos no versionados en cualquier lugar de su copia de trabajo. Ahí podrá seleccionar o deseleccionar ítems para ser eliminados.

Cuando dichos ítems se eliminan, se utiliza la papelera de reciclaje, por lo que si comete un error aquí y borra un archivo que debería haber estado versionado, aún puede recuperarlo.

## 4.16. Deshacer cambios

Si desea deshacer todos los cambios que ha hecho en un archivo desde la última actualización, necesita seleccionar el archivo, hacer click con el botón derecho para sacar el menú contextual, y luego seleccionar el comando TortoiseSVN → Revertir Aparecerá un diálogo que le muestra los archivos que ha cambiado y que puede revertir. Seleccione los que desee revertir y pulse Aceptar.



**Figura 4.35. Diálogo de Revertir**

Si además quieres limpiar todas las listas de cambios realizados, da clic en la casilla en la parte inferior del cuadro de diálogo.

Si desea deshacer un borrado o un renombrado, necesitará utilizar Revertir en la carpeta padre dado que el ítem borrado ya no existe y no podrá hacer click con el botón derecho sobre él.

Si quiere deshacer la adición de un ítem, esto aparece en el menú contextual como TortoiseSVN → Deshacer añadir.... Esto también es revertir, en realidad, pero su nombre se ha cambiado para hacerlo más obvio.

Las columnas en este diálogo pueden ser personalizadas de la misma forma que las columnas en el diálogo Comprobar modificaciones. Para más detalles, lea [Sección 4.7.3, “Estado local y remoto”](#).

Ya que revertir es a veces utilizado para limpiar una copia de trabajo, existe un botón adicional que le permite eliminar además, objetos no versionados. Al momento de hacer clic en este botón, aparecerá otro diálogo, el cual le mostrará un listado de todos los objetos no versionados y que pueden ser seleccionados para ser eliminados.



### **Deshaciendo cambios que han sido confirmados**

Revertir sólo deshará sus cambios locales. *No* deshace ningún cambio que ya haya sido confirmado. Si desea deshacer todos los cambios que se confirmaron en una revisión en particular, lea [Sección 4.10, “Diálogo de Registro de revisiones”](#) para obtener más información.



### **Revertir es lento**

Cuando revierte los cambios puede que note que la operación lleva bastante más tiempo del que sería de esperar. Esto es así porque la versión modificada del archivo se manda a la papelera de reciclaje, para que pueda recuperar sus cambios si ha revertido por error. Sin embargo, si su papelera



de reciclaje está llena, Windows necesita bastante tiempo para encontrar dónde poner el archivo. La solución es simple: o bien vacía la papelera de reciclaje, o bien desactiva la casilla Utilizar la papelera de reciclaje al revertir en la configuración de TortoiseSVN.

## 4.17. Limpieza

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**

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

### Limpie el estado copia de trabajo

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

### Break write locks

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

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

### Fix time stamps

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

### Vacuum pristine copies

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

### Actualizar las superposiciones del shell

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

### Incluir externos

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

Eliminar archivos y carpetas no versionados, borrar archivos y carpetas ignorado

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

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

Revertir todos los cambios de forma recursiva

Este comando revierte todas las modificaciones locales que no se han confirmado todavía.

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

## 4.18. Configuración del proyecto

### 4.18.1. Propiedades de Subversion

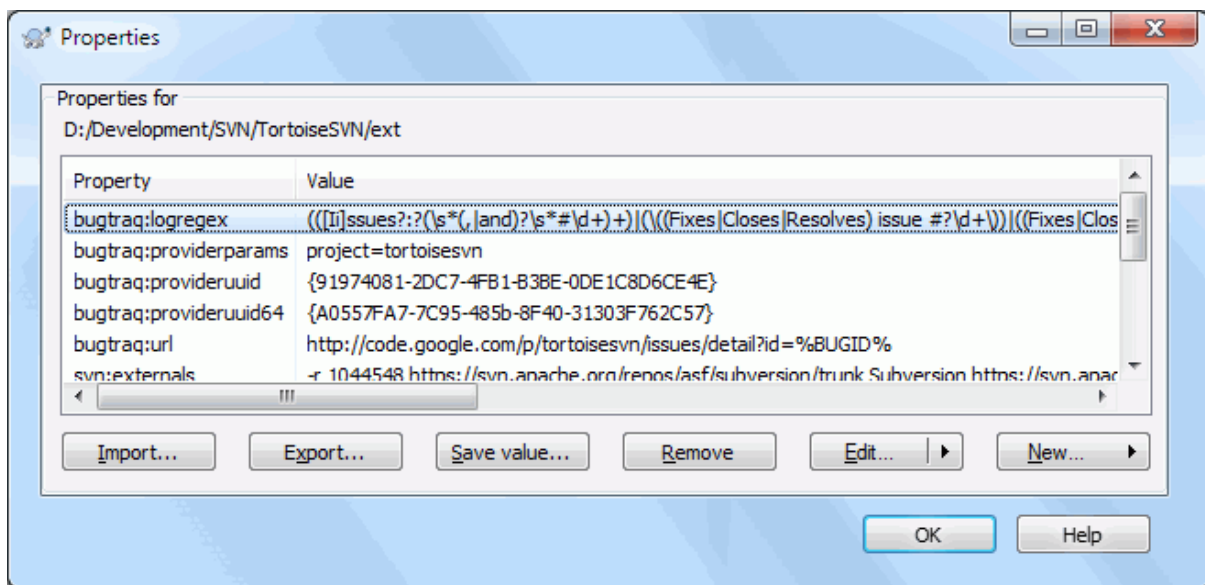


Figura 4.37. Página de propiedades de Subversion

Puede leer y establecer las propiedades de Subversion desde el diálogo de propiedades de Windows, pero también desde TortoiseSVN → Propiedades y dentro de las listas de estado de TortoiseSVN, desde Menú contextual → Propiedades.

Puede añadir sus propias propiedades, o algunas propiedades con un significado especial en Subversion. Éstas empiezan con `svn:`. `svn:externals` es una de esas propiedades; vea cómo manejar externos en [Sección 4.19, “Ítems externos”](#).

#### 4.18.1.1. svn:keywords

Subversion soporta expansión de palabras clave similar a CVS, lo que puede utilizarse para introducir el nombre del archivo y la información de la revisión dentro del propio archivo. Las palabras clave actualmente soportadas son:

\$Date\$

Fecha de la última confirmación conocida. Esto se basa en la información obtenida cuando actualiza su copia de trabajo. *No* se contacta con el repositorio para buscar cambios más recientes.

\$Revision\$

Revisión de la última confirmación conocida.

\$Author\$

Autor que hizo la última confirmación conocida.

\$HeadURL\$

La URL completa de este archivo en el repositorio.

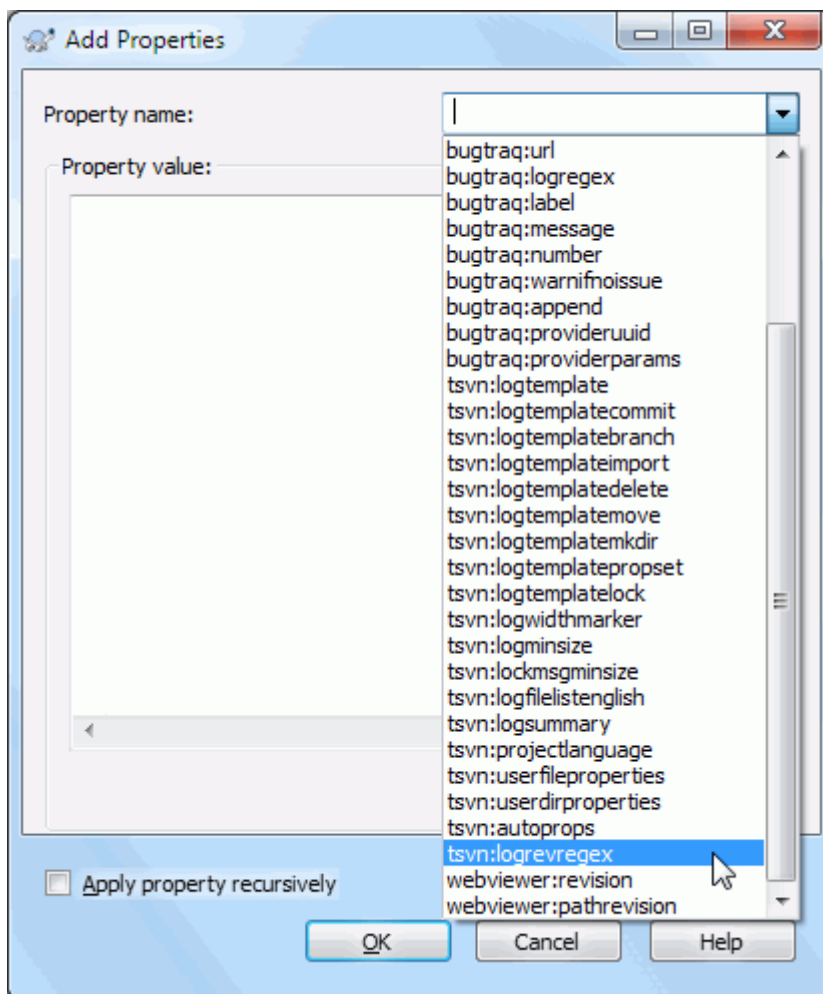
\$Id\$

Una combinación comprimida de las cuatro palabras clave anteriores.

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. Añadiendo y editando propiedades



**Figura 4.38. Añadiendo propiedades**

Para agregar una nueva propiedad, primero presione sobre **Nuevo...** Seleccione el nombre de la propiedad requerida del menu, y luego complete la información requerida en el diálogo de propiedades específico. Estos diálogos de propiedades específicos están descritos más en detalle en [Sección 4.18.3, “Editores de Propiedades”](#).

Para agregar una propiedad que no tiene su propio diálogo, seleccione **Avanzada** del menu **Nuevo...** Luego o bien seleccione una propiedad existente en la lista desplegable o ingrese un nombre nuevo.

Si desea aplicar una propiedad a muchos ítems a la vez, seleccione los archivos/carpetas en el explorador, y luego seleccione **Menú contextual** → **propiedades**.

Si desea aplicar la propiedad a *cada* archivo y carpeta en la jerarquía debajo de la carpeta actual, seleccione la casilla **Recursivo**.

Si desea editar una propiedad existente, primero seleccione dicha propiedad de la lista de propiedades existentes, y luego pulse en **Editar...**

Si desea eliminar una propiedad existente, seleccione dicha propiedad de la lista de propiedades existentes, y luego pulse en **Eliminar**.

La propiedad `svn:externals` se puede utilizar para traer otros proyectos del mismo repositorio o de otro completamente distinto. Para más información, lea [Sección 4.19, “Ítems externos”](#).



### **Editar las propiedades de la revisión HEAD**

Dado que las propiedades están versionadas, no puede editar las propiedades de revisiones previas. Si mira propiedades del diálogo de registro, o desde una revisión non-HEAD del explorador de repositorio, verá una lista de propiedades y valores, pero ningún control de edición.

#### **4.18.1.3. Exportando e importando propiedades**

A menudo se encontrará aplicando el mismo conjunto de propiedades muchas veces, por ejemplo `bugtraq:logregex`. Para simplificar el proceso de copiar propiedades de un proyecto a otro, puede utilizar la característica **Exportar/Importar**.

Desde el archivo o carpeta en el que las propiedades ya están establecidas, utilice **TortoiseSVN** → **Propiedades**, seleccione las propiedades que desea exportar y haga click en **Exportar...** Se le preguntará por un nombre de archivo donde se grabarán los nombres de las propiedades y sus valores.

Desde la(s) carpeta(s) donde desea aplicar estas propiedades, utilice **TortoiseSVN** → **Propiedades** y haga click en **Importar...** Se le preguntará por un nombre de archivo desde el que importar, por lo que deberá navegar al lugar donde almacenó el archivo exportado anteriormente y seleccionarlo. Las propiedades se añadirán a las carpetas de forma no recursiva.

Si desea añadir las carpetas a un árbol recursivamente, siga los pasos anteriores, y luego en el diálogo de propiedades seleccione cada propiedad por turnos, haga click en **Editar...**, seleccione la casilla **Aplicar propiedad recursivamente** y haga click en **Aceptar**.

El formato del archivo de importación es binario y propietario de TortoiseSVN. Su único propósito es transferir las propiedades utilizando **Importar** y **Exportar**, por lo que no hay necesidad de editar estos archivos.

#### **4.18.1.4. Propiedades binarias**

TortoiseSVN puede manejar valores de propiedades binarios utilizando archivos. Para leer un valor de propiedad binario, utilice **Grabar...** para almacenarlo en un archivo. Para establecer un valor binario, utilice un editor hexadecimal u otra herramienta adecuada para crear un archivo con el contenido que necesite, y luego utilice **Cargar...** para leer ese archivo.

Aunque las propiedades binarias no se utilizan a menudo, pueden ser útiles en algunas aplicaciones. Por ejemplo, si está almacenando archivos gráficos muy grandes, o si la aplicación que se utiliza para cargar el archivo es pesada, puede querer guardar una pequeña imagen guía como una propiedad para poder obtener una previsualización rápidamente.

#### **4.18.1.5. Establecer propiedades automáticamente**

Puede configurar Subversion y TortoiseSVN para que establezca propiedades automáticamente en archivos y carpetas cuando se añaden al repositorio. Hay dos formas de hacerlo.

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.

Un método alternativo es establecer la propiedad `tsvn:autoprops` en las carpetas, como se describe en la siguiente sección. Este método funciona sólo con los clientes TortoiseSVN, pero se propaga a todas las copias de trabajo al actualizarlas.

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.

Sea cual sea el método que elija, debe tener en cuenta que las auto-props se aplican sólo a los archivos en el momento en que se añaden al repositorio. Las auto-props nunca cambiarán las propiedades de archivos que ya estén versionados.

Si desea estar totalmente seguro que a los nuevos archivos les son aplicadas las propiedades correctas, debería establecer un script gancho pre-confirmación para rechazar las confirmaciones cuando estas propiedades necesarias no estén establecidas.



### Confirmar propiedades

Las propiedades de Subversion están versionadas. Después que cambie o añada una propiedad tiene que confirmar sus cambios.



### Conflictos en las propiedades

Si hay un conflicto al confirmar los cambios, porque otro usuario ha cambiado la misma propiedad, Subversion genera un archivo `.prej`. Borre este archivo después que haya resuelto del conflicto.

## 4.18.2. Propiedades de proyecto TortoiseSVN

TortoiseSVN tiene unas pocas propiedades especiales para sí mismo, y estas empiezan con `tsvn:`.

- `tsvn:logminsize` establece la longitud mínima de un mensaje de registro para una confirmación. Si introduce un mensaje más corto de lo especificado aquí, la confirmación se deshabilita. Esta característica es muy útil para que se acuerde de proporcionar un mensaje descriptivo apropiado para cada confirmación. Si esta propiedad no se establece, o el valor es cero, se permiten mensajes de registro vacíos.

`tsvn:lockmsgminsize` establece la longitud mínima para un mensaje de bloqueo. Si introduce un mensaje más corto de lo especificado aquí, el bloqueo se deshabilita. Esta característica es muy útil para que se acuerde de proporcionar un mensaje descriptivo apropiado para cada bloqueo que obtenga. Si esta propiedad no se establece, o el valor es cero, se permiten mensajes de bloqueo vacíos.

- `tsvn:logwidthmarker` se utiliza con proyectos que necesitan que los mensajes de registro se formateen con algún ancho máximo (típicamente, 80 caracteres) antes de un salto de línea. Al establecer esta propiedad a un valor distinto de cero ocurren dos cosas en el diálogo de entrada de mensajes de registro: pone un marcador para indicar el ancho máximo, y deshabilita el ajuste de línea en la pantalla, para que pueda ver si el texto que ha introducido es muy largo. Tenga en cuenta que esta característica sólo funcionará correctamente si ha seleccionado una fuente de ancho fijo para los mensajes de registro.
- `tsvn:logtemplate` se utiliza en proyectos que tienen reglas sobre el formato de mensajes. Esta propiedad alberga una cadena de texto de múltiples líneas que se insertará en el cuadro del mensaje de confirmación cuando inicie una confirmación. Puede editarlo para incluir la información necesaria. Tenga en cuenta que si estaba utilizando también `tsvn:logminsize`, debe asegurarse de establecer una longitud mayor que la de la plantilla o perderá el mecanismo de protección.

También hay plantillas de acción específicas que puede utilizar en lugar de `tsvn:logtemplate`. Se utilizarán la plantillas de acción específica si se en los casos configurados, pero se utilizará `tsvn:logtemplate` si no hay plantilla de acción específica establecida.

Los templates de acción posibles son:

- `tsvn:logtemplatecommit` se utiliza para todas las confirmaciones de una copia de trabajo.
- `tsvn:logtemplatebranch` se utiliza cuando se crea una rama/etiqueta, o al copiar archivos o carpetas directamente en el navegador de repositorios.
- `tsvn:logtemplateimport` se utiliza para importaciones.
- `tsvn:logtemplatedelete` se utiliza cuando se eliminan elementos directamente en el navegador de repositorios.
- `tsvn:logtemplatemove` se utiliza cuando se renombran o mueven elementos en el navegador de repositorios.
- `tsvn:logtemplatemkdir` se utiliza cuando se crean directorios en el navegador de repositorios.
- `tsvn:logtemplatepropset` se utiliza al modificar propiedades en el navegador de repositorios.
- `tsvn:logtemplatelock` se utiliza al obtener un bloqueo.
- 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.

Si hay un conflicto entre las autoprops locales y `tsvn:autoprops`, la configuración del proyecto tiene preferencia ya que son específicas para ese proyecto.

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.

- En el diálogo de Confirmar tiene la opción de pegar la lista de archivos cambiados, incluyendo el estado de cada archivo (añadido, modificado, etc). `tsvn:logfilelistenglish` define si el estado del archivo se inserta en inglés o en el idioma traducido. Si la propiedad no se establece, el valor por defecto es `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].

Puede introducir este valor en decimal, o en hexadecimal si se precede por `0x`. Por ejemplo, Español (España, ordenación tradicional) puede introducirse como `0x040a` or `1034`.

- La propiedad `tsvn:logsummary` se utiliza para extraer una porción del mensaje de registro que luego se muestra en el diálogo de registro como el sumario del mensaje de registro.

El valor de la propiedad `tsvn:logsummary` debe establecerse a una cadena de texto regex de una línea que contenga un grupo regex. Lo que satisfaga ese grupo se utilizará como sumario.

Por ejemplo: `\[SUMARIO\]:\s+(.*)` capturará todo lo que haya tras “[SUMARIO]” en el mensaje de registro y lo utilizará como sumario.

- La propiedad `tsvn:logrevregex` define una expresión regular que corresponde directamente a referencias hechas con respecto a revisiones en un mensaje de registro. Esta es utilizada en el diálogo de registro para convertir a tales referencias en ligas, las cuales al momento de ser seleccionadas haciendo clic sobre ellas, podrán llevarle directamente a dicha revisión (si la revisión se muestra en el diálogo de bitácora o si se encuentra disponible en el diálogo de registro) o bien, abrirá un nuevo diálogo de registro que le mostrará dicha revisión.

La expresión regular debe de corresponder a la referencia íntegra, no tan solo a el número de revisión. El número de revisión se extrae de forma automática de la cadena de referencia correspondiente.

Si esta propiedad no se ha establecido, se utilizará una expresión regular por defecto para ligar referencias a revisiones.

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

The available properties/hook-scripts are

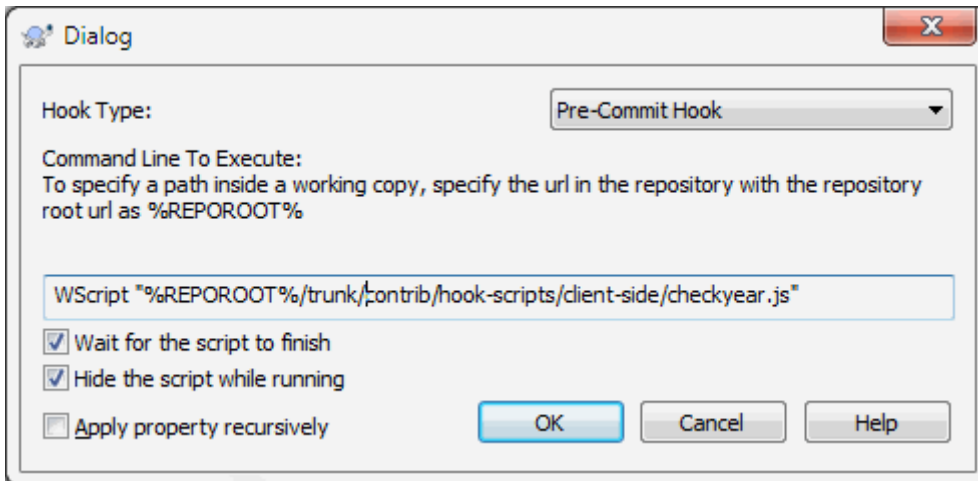
- `tsvn:startcommithook`
- `tsvn:precommithook`
- `tsvn:postcommithook`
- `tsvn:startupdatehook`
- `tsvn:preupdatehook`
- `tsvn:postupdatehook`
- `tsvn:prelockhook`
- `tsvn:postlockhook`

The parameters are the same as if you would configure the hook scripts in the settings dialog. See [Sección 4.31.8, “Scripts gancho del lado del cliente”](#) for the details.

Since not every user has his or her working copy checked out at the same location with the same name, you can configure a script/tool to execute that resides in your working copy by specifying the URL in the repository instead, using `%REPOROOT%` as the part of the URL to the repository root. For example, if your hook script is in your working copy under `contrib/hook-scripts/client-side/checkyear.js`, you would specify the path to the script as `%REPOROOT%/trunk/contrib/hook-scripts/client-side/checkyear.js`. This way even if you move your repository to another server you do not have to adjust the hook script properties.

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

The following screenshot shows how the script to check for current copyright years in source file headers is configured for TortoiseSVN.



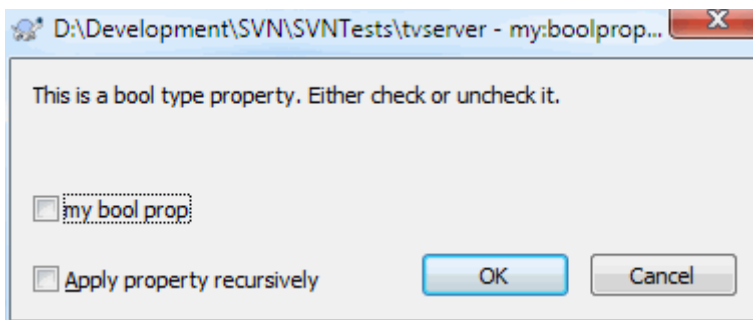
**Figura 4.39. Property dialog for hook scripts**

- Cuando quiera añadir una nueva propiedad, puede o bien elegir una desde la lista en el cuadro desplegable, o puede introducir el nombre de propiedad que desee. Si su proyecto utiliza algunas propiedades personalizadas, y quiere que esas propiedades aparezcan en la lista del cuadro desplegable (para evitar errores ortográficos cuando introduzca un nombre de propiedad), puede crear una llista de sus propias propiedades personalizadas utilizando `tsvn:userfileproperties` y `tsvn:userdirproperties`. Aplique estas propiedades a una carpeta. Cuando vaya a editar las propiedades de cualquier ítem hijo, sus propiedades personalizadas aparecerán en la lista de nombres de propiedades predefinidas.

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

**bool**

If your property can only have two states, e.g., true and false, then you can configure your property as a bool type.



**Figura 4.40. Property dialog boolean user types**

Specify your property like this:

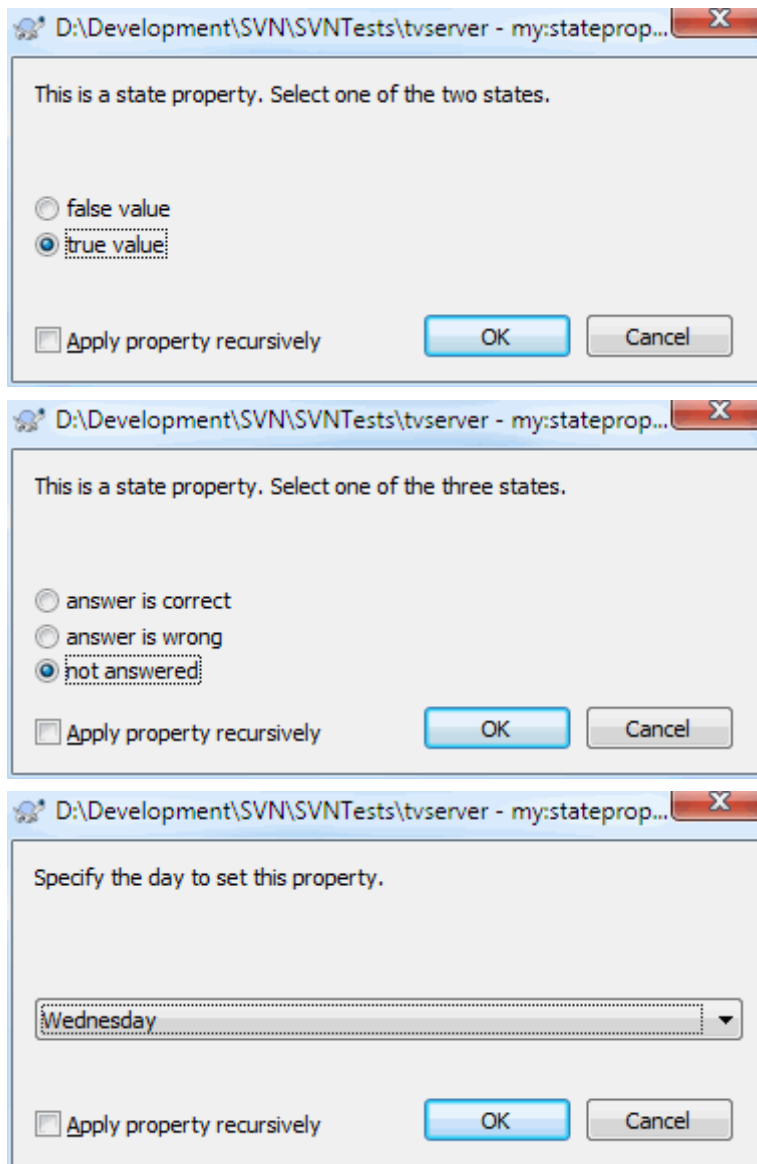
```
propertyname=bool ; labeltext ( YESVALUE ; NOVALUE ; Checkboxtext )
```

the `labeltext` is the text shown in the dialog above the checkbox where you can explain the purpose and use of the property. The other parameters should be self explanatory.

**state**

If your property represents one of many possible states, e.g., `yes`, `no`, `maybe`, then you can configure your property as a `state`





**Figura 4.41. Cuadro de diálogo de tipos de estados de usuario.**

property like this:

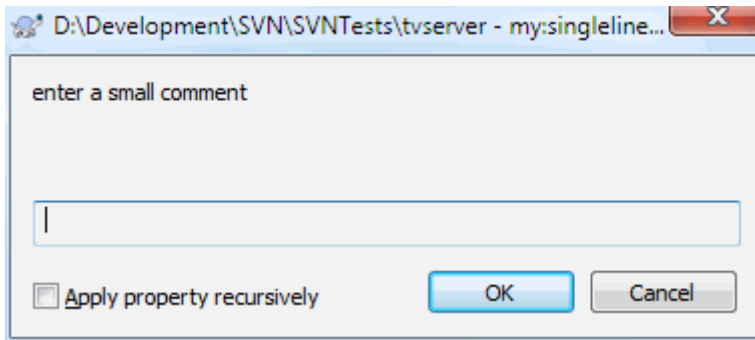
```
propertyname=state;labeltext(DEFVAL;VAL1;TEXT1;VAL2;TEXT2;VAL3;TEXT3;...)
```

The parameters are the same as for the `bool` property, with `DEFVAL` being the default value to be used if the property isn't set yet or has a value that's not configured.

For up to three different values, the dialog shows up to three radio buttons. If there are more values configured, it uses a combo box from where the user can select the required state.

`singleline`

For properties that consist of one line of text, use the `singleline` property type:



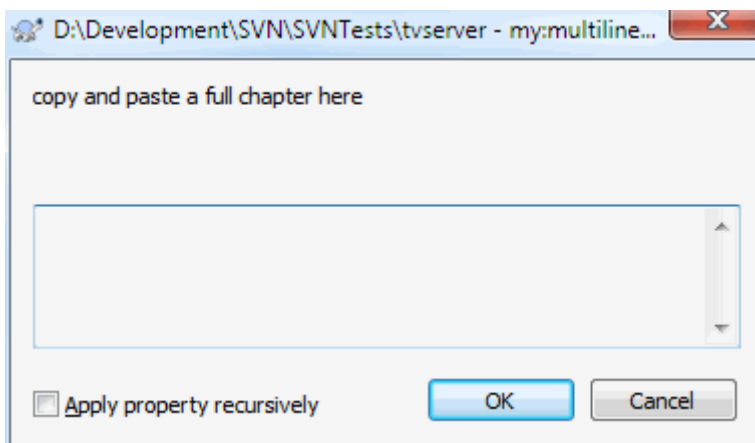
**Figura 4.42. Diálogo de propiedades de tipos de usuario de una línea.**

```
propertyname=singleline;labeltext(regex)
```

the `regex` specifies a regular expression which is used to validate (match) the text the user entered. If the text does not match the `regex`, then the user is shown an error and the property isn't set.

multiline

For properties that consist of multiple lines of text, use the `multiline` property type:



**Figura 4.43. Property dialog multi-line user types**

```
propertyname=multiline;labeltext(regex)
```

the `regex` specifies a regular expression which is used to validate (match) the text the user entered. Don't forget to include the newline (`\n`) character in the `regex`!

The screenshots above were made with the following `tsvn:userdirproperties`:

```
my:boolprop=bool;This is a bool type property. Either check or uncheck it.(true/false)
my:stateprop1=state;This is a state property. Select one of the two states.(true/true)
my:stateprop2=state;This is a state property. Select one of the three states.(maybe/true)
my:stateprop3=state;Specify the day to set this property.(1;1;Monday;2;Tuesday;3;Wednesday)
my:singlelineprop=singleline;enter a small comment(.*)
my:multilineprop=multiline;copy and paste a full chapter here(.*)
```

TortoiseSVN puede integrarse con algunas herramientas de control de errores. Esto utiliza propiedades de proyecto que comienzan con `bugtraq:`. Para más información, lea [Sección 4.29, "Integración con sistemas de control de errores / seguimiento de incidencias"](#).

También puede integrarse con algunos visores de repositorios basados en web, utilizando propiedades de proyecto que comienzan con `webviewer:`. Para más información, lea [Sección 4.30, “Integración con visores de repositorios basados en web”](#).



## Establecer las propiedades de proyecto en carpetas

Estas propiedades especiales del proyecto deben estar establecidas en *carpetas* para que el sistema funcione. Cuando confirma un archivo o una carpeta, se leen las propiedades de esa carpeta. Si no se encuentran allí las propiedades, TortoiseSVN las buscará hacia arriba en el árbol de carpetas para encontrarlas, hasta que llega a una carpeta sin versionar, o se encuentra la raíz del árbol (por ejemplo, `C:\`). Si puede estar seguro de que cada usuario obtiene sólo desde por ejemplo `trunk/` y no desde alguna subcarpeta, entonces es suficiente establecer las propiedades en `trunk/`. Si no puede estar seguro, debería establecer las propiedades recursivamente en cada subcarpeta. Una propiedad establecida en una carpeta más profunda dentro de la jerarquía del proyecto tiene preferencia sobre las propiedades establecidas en niveles más altos (más cerca de `trunk/`).

*Excusivamente para las propiedades de proyecto*, por ejemplo: `tsvn:`, `bugtraq:` and `webviewer:`, puede utilizar la casilla **Recursivo** para establecer la propiedad en todas las subcarpetas de la jerarquía, sin establecerla en todos los archivos.

Cuando añade nuevas subcarpetas a una copia de trabajo utilizando TortoiseSVN, cualquier propiedad de proyecto presente en la carpeta padre se añadirá también automáticamente a la nueva carpeta hija.



## Limitaciones al Usar el Navegador de Repositorios

Obtener propiedades de forma remota es una operación lenta por lo que algunas características descritas antes no funcionarán en el navegador de repositorios de la misma forma como en la copia de trabajo.

- Cuando añade una propiedad usando el navegador de repositorio, sólo las propiedades `svn:` estándar son elegibles en la lista predefinida. Cualquier otra propiedad debe ser añadida manualmente.
- Las propiedades no pueden ser guardadas o borradas recursivamente usando en navegador de repositorio.
- Las propiedades de proyecto *no* se propagarán automáticamente cuando un directorio hijo es añadido usando el navegador de repositorio.
- `tsvn:autoprops` *no* establecerá propiedades en archivos que hayan sido añadidos usando el navegador de repositorio.



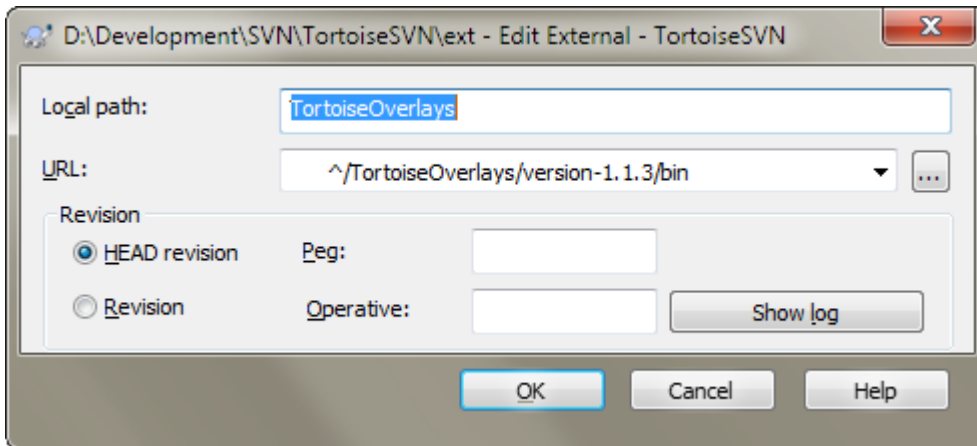
## Atención

Aunque las propiedades de proyecto de TortoiseSVN son extremadamente útiles, sólo funcionan con TortoiseSVN, y algunas sólo funcionarán en las versiones más recientes de TortoiseSVN. Si la gente que trabaja en su proyecto utiliza un número de clientes de Subversion diferentes, o es posible que tengan versiones antiguas de TortoiseSVN, puede que quiera utilizar ganchos de repositorio para forzar políticas de proyectos. Las propiedades de proyecto sólo ayudan a implementar una política, no pueden forzarlas.

### 4.18.3. Editores de Propiedades

Algunas propiedades tienen que utilizar valores específicos, o tener el formato de una manera específica para poder ser utilizados en la automatización. Para ayudar a que el formato sea correcto, TortoiseSVN presenta cuadros de diálogo para editar algunas propiedades particulares que muestran los posibles valores o separa la propiedad en sus componentes individuales.

### 4.18.3.1. Contenido externo



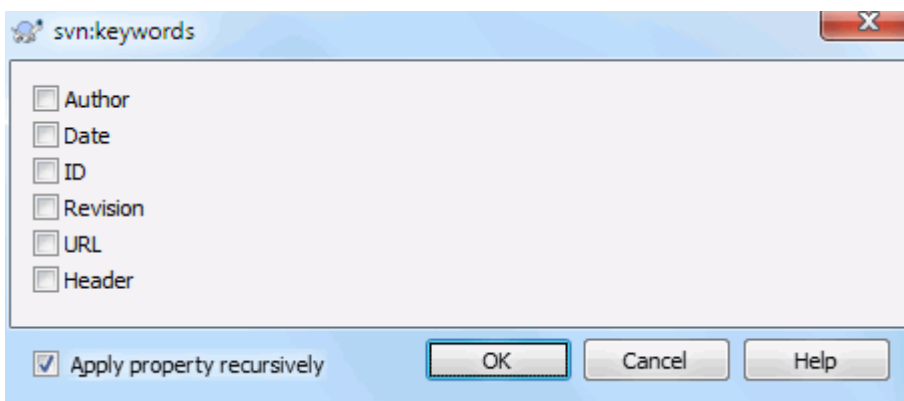
**Figura 4.44. svn:Página de propiedades de externos**

La propiedad `svn:externals` se puede utilizar para traer otros proyectos del mismo repositorio o un repositorio completamente diferente como se describe en [Sección 4.19, “Ítems 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.

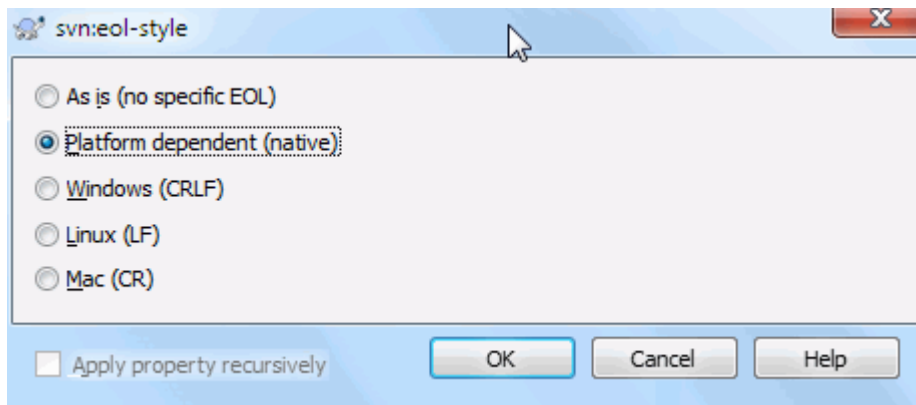
### 4.18.3.2. Palabras clave de SVN



**Figura 4.45. svn: Página de propiedades de palabras clave**

Seleccione las palabras clave que desea que sean ampliadas en el archivo.

### 4.18.3.3. Estilo EOL



**Figura 4.46. svn:Página de propiedades de estilo EOL**

Seleccione el estilo de final de línea que desea utilizar y TortoiseSVN utilizará el valor correcto de la propiedad.

#### 4.18.3.4. Integración con el gestor de incidencias

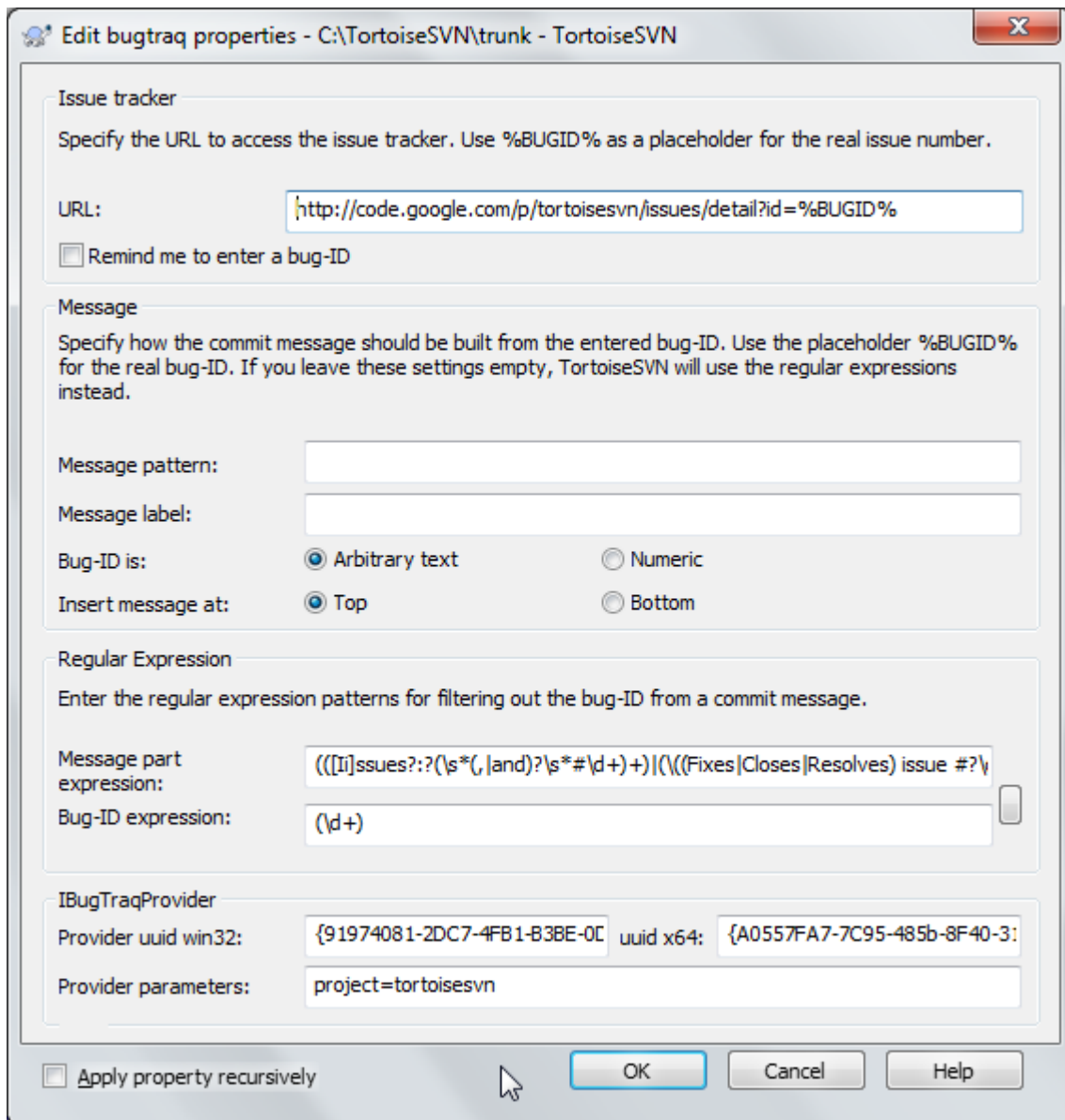
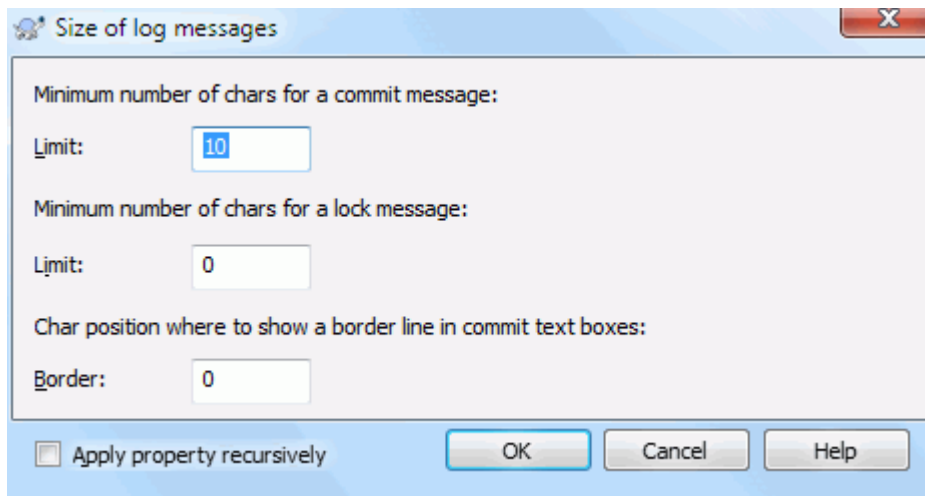


Figura 4.47. tsvn: Página de propiedades de bugtraq

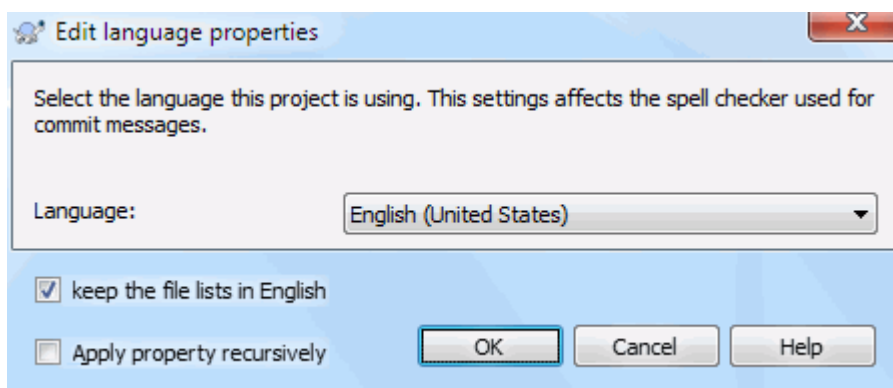
#### 4.18.3.5. Tamaños de mensajes de registro



**Figura 4.48. Tamaño de la página de propiedades del registro de mensajes**

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.

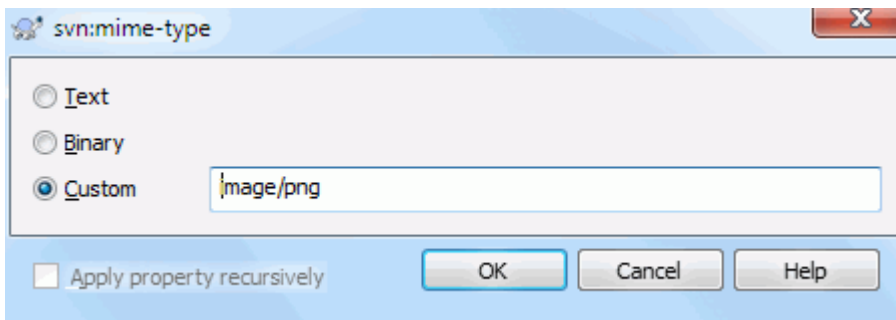
#### 4.18.3.6. Idioma del proyecto



**Figura 4.49. Página de propiedades de idioma.**

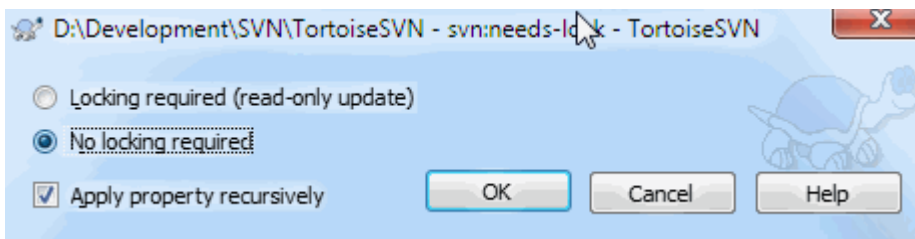
Choose the language to use for spell-checking log messages in the commit dialog. The file lists checkbox comes into effect when you right click in the log message pane and select **Paste file list**. By default the Subversion status will be shown in your local language. When this box is checked the status is always given in English, for projects which require English-only log messages.

#### 4.18.3.7. tipo-MIME



**Figura 4.50. svn:página de propiedades tipo-mime**

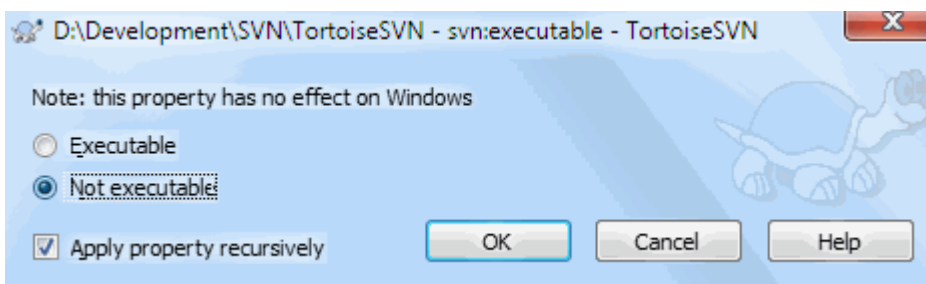
#### 4.18.3.8. svn:needs-lock



**Figura 4.51. svn:página de propiedades necesita-bloquea**

Esta propiedad simplemente controla si un archivo es de sólo lectura en el caso de que no haya un bloqueo mantenido por la copia de trabajo.

#### 4.18.3.9. svn:ejecutable



**Figura 4.52. svn:página de propiedades de ejecutables**

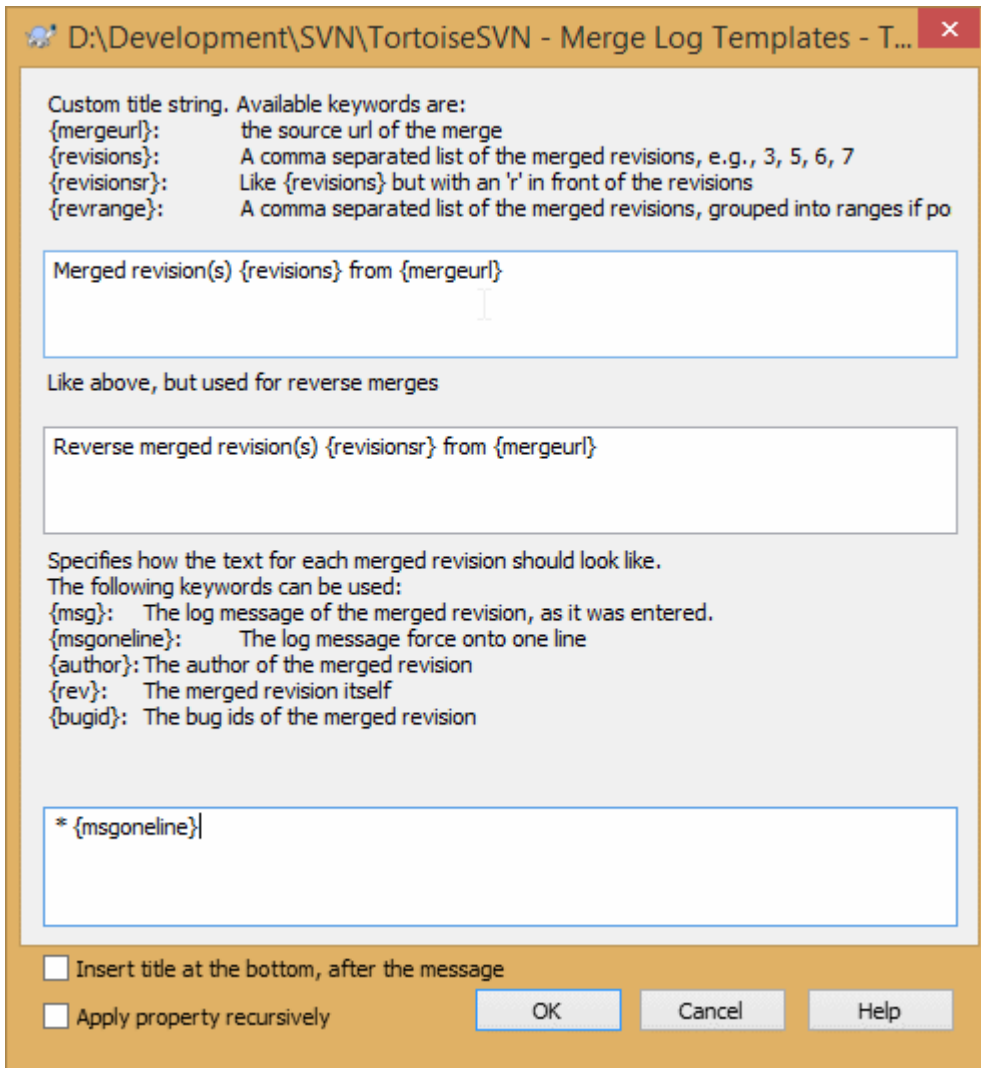
Esta propiedad controla si un archivo se le dará el estado de ejecutable cuando se realice el check out en un sistema Unix / Linux. No tiene ningún efecto en una maquina Windows.

#### 4.18.3.10. Merge log message templates

Whenever revisions are merged into a working copy, TortoiseSVN generates a log message from all the merged revisions. Those are then available from the **Recent Messages** button in the commit dialog.

Usted puede configurar el mensaje generado con las siguientes propiedades:





**Figura 4.53. Property dialog merge log message templates**

tsvn:mergelogtemplatetitle, tsvn:mergelogtemplatereversetitle

This property specifies the first part of the generated log message. The following keywords can be used:

{revisions}

A comma separated list of the merged revisions, e.g., 3, 5, 6, 7

{revisionsr}

Like {revisions}, but with each revision preceded with an r, e.g., r3, r5, r6, r7

{revrange}

A comma separated list of the merged revisions, grouped into ranges if possible, e.g., 3, 5-7

{mergeurl}

The source URL of the merge, i.e., where the revisions are merged from.

The default value for this string is `Merged revision(s) {revrange} from {mergeurl}`: with a newline at the end.

tsvn:mergelogtemplatemsg

This property specifies how the text for each merged revision should look like. The following keywords can be used:

{msg}

The log message of the merged revision, as it was entered.

{msgoneline}

Like {msg}, but all newlines are replaced with a space, so that the whole log message appears on one single line.

{author}

El autor de la versión fusionada

{rev}

The merged revision itself.

{bugids}

The bug IDs of the merged revision, if there are any.

tsvn:mergelogtemplatemsgtitlebottom

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



### Importante

This only works if the merged revisions are already in the log cache. If you have disabled the log cache or not shown the log first before the merge, the generated message won't contain any information about the merged revisions.

## 4.19. Ítems externos

A veces es útil construir una copia de trabajo que esté hecha de un número de diferentes obtenciones. Por ejemplo, puede querer que diferentes archivos o carpetas provengan de orígenes distintos de un repositorio, o quizás de repositorios diferentes. Si desea que todos los usuarios tengan la misma estructura, puede definir las propiedades `svn:externals` para traer el recurso especificado al lugar donde se le necesita.

### 4.19.1. Carpetas externas

Imaginemos que usted ha obtenido la copia de trabajo de `/project1` a `D:\dev\project1`. Seleccione la carpeta `D:\dev\project1`, botón derecho y seleccione **Menu Windows** → **Propiedades** del menú contextual. Aparecerá el diálogo de propiedades. Vaya a la pestaña de subversion. Allí podrá establecer propiedades. Presione **Propiedades...** En el diálogo de propiedades, o bien haga doble clic o `svn:externals` si ya existe, o presione sobre el botón **Nuevo...** y seleccione `externals` del menu. Para agregar un externo nuevo, presione **Nuevo...** y luego complete la información requerida en el diálogo.



### Atención

Los URLs deberán de estar adecuadamente escapados o de lo contrario, no funcionarán, por ejemplo: usted debe de reemplazar cada espacio con `%20`, tal y como se muestra en el segundo ejemplo anterior.

Si usted desea que la ruta local incluya espacios u cualquier otro caracter especial, puede encerrarlo con comillas dobles, o bien, utilizar el `\` símbolo de barras hacia atrás como un caracter de escape al estilo del shell de Unix, precediendo a cualquier caracter especial. Por supuesto que esto significa también que deberá de usar el símbolo `/` (barra hacia delante) como delimitador de ruta. Este comportamiento es nuevo en Subversión 1.6 y no funcionará en clientes anteriores.



### Utilice números globales de revisión

You should strongly consider using explicit revision numbers in all of your externals definitions, as described above. Doing so means that you get to decide when to pull down a different snapshot of external information, and exactly which snapshot to pull. Besides the common sense aspect of

not being surprised by changes to third-party repositories that you might not have any control over, using explicit revision numbers also means that as you backdate your working copy to a previous revision, your externals definitions will also revert to the way they looked in that previous revision, which in turn means that the external working copies will be updated to match the way *they* looked back when your repository was at that previous revision. For software projects, this could be the difference between a successful and a failed build of an older snapshot of your complex code base.

The edit dialog for `svn:externals` properties allows you to select the externals and automatically set them explicitly to the HEAD revision.

Si el proyecto externo está en el mismo repositorio, cualquier cambio que haga allí se incluirá en la lista de confirmación cuando confirme su proyecto 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.

Si utiliza URLs absolutas en las definiciones `svn:externals` y tiene que relocalizar su copia de trabajo (por ejemplo si la URL de su repositorio cambia), entonces sus externos no cambiarán y puede que no funcionen más.

Para evitar dichos problemas, los clientes de Subversion versión 1.5 y superior soportan URLs externas relativas. Hay cuatro formas para especificar una URL relativa. En los siguientes ejemplos, asumimos que tenemos dos repositorios: uno en `http://example.com/svn/repos-1` y otro en `http://example.com/svn/repos-2`. Tenemos una copia de trabajo de `http://example.com/svn/repos-1/project/trunk` en `C:\Working` y la propiedad `svn:externals` está establecida en `trunk`.

Relativa a la carpeta padre

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`.

Tenga en cuenta que la URL es relativa a la URL del directorio con la propiedad `svn:externals`, no al directorio donde la carpeta externa se escribe en el disco.

Relativo a la raíz del repositorio

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`.

Relativa al 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`.

Relativa al nombre del 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.perevs.html) [http://svnbook.red-bean.com/en/1.8/svn.advanced.perevs.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.

Si necesita más información sobre cómo TortoiseSVN maneja las Propiedades, lea [Sección 4.18, “Configuración del proyecto”](#).

Para averiguar los diferentes métodos para acceder a subproyectos comunes lea [Sección B.6, “Incluir un subproyecto común”](#).

## 4.19.2. Archivos externos

Desde Subversion 1.6 puede añadir un archivo suelto externo a su copia de trabajo utilizando la misma sintaxis que se usa para carpetas. Sin embargo, hay algunas restricciones.

- The path to the file external must be a direct child of the folder where you set the `svn:externals` property.
- La URL de un archivo externo debe estar en el mismo repositorio que la URL donde se insertará el archivo externo; los archivos externos inter-repositorio no están soportados.

Un archivo externo se comporta como cualquier otro archivo versionado en muchos aspectos, pero no pueden ser movidos o eliminados utilizando los comandos normales; en vez de eso, debe modificarse la propiedad `svn:externals`.

## 4.19.3. Creando exteriores vía arrastrar y soltar

Si ya tienes una copia funcional de los archivos o directorios que deseas incluir como externos en otra copia funcional, puedes sencillamente añadirlos mediante arrastrar y soltar desde el explorador de Windows.

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. Haciendo ramas / etiquetas

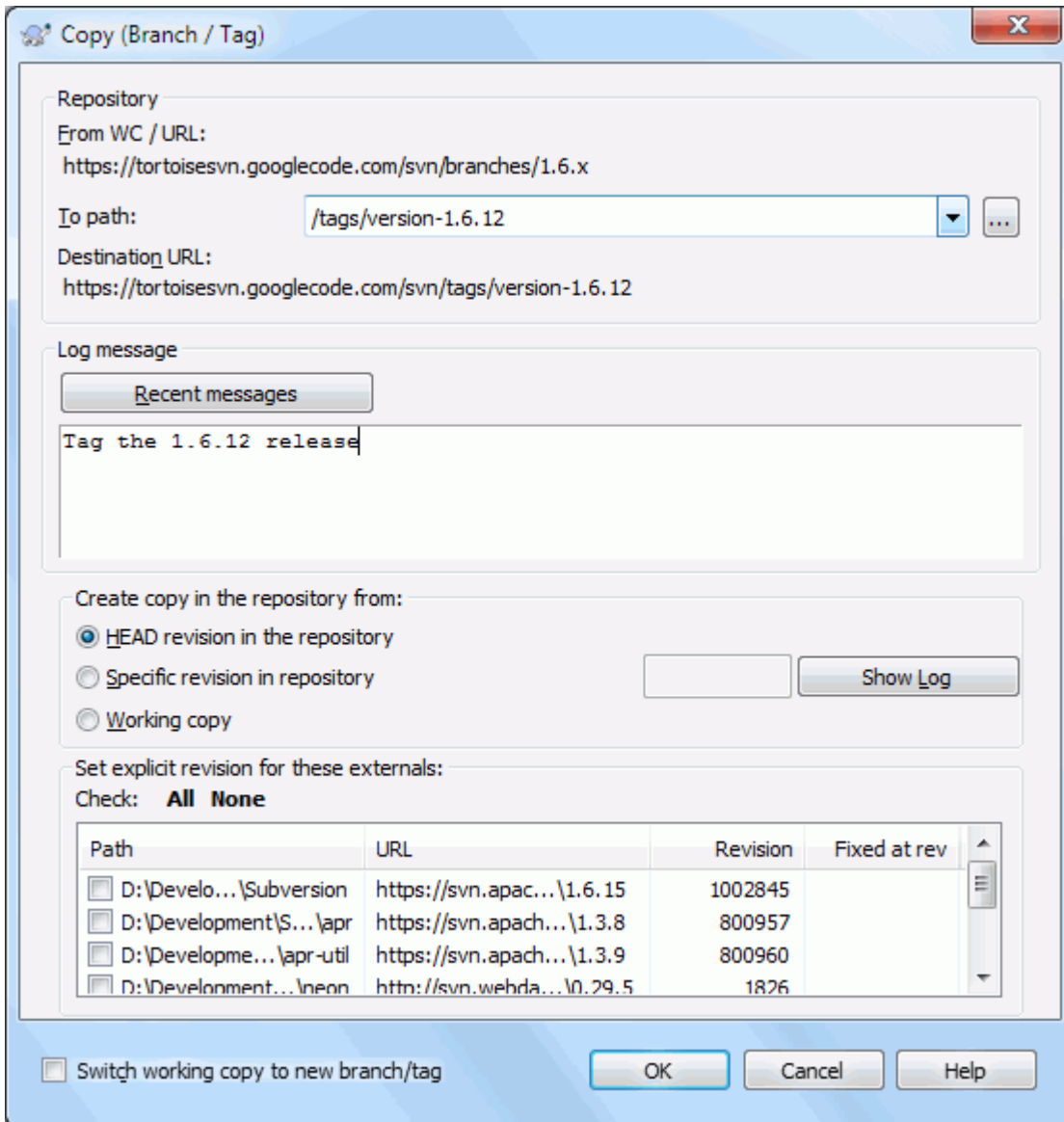
Una de las características de los sistemas de control de versiones es la posibilidad de aislar cambios en una línea separada de desarrollo. Esto se conoce como una *rama*. Las ramas se utilizan a menudo para probar nuevas características sin molestar la línea principal del desarrollo con errores de compilación y errores. Tan pronto como la nueva característica es lo suficiente estable, la rama de desarrollo se *fusiona* de nuevo en la rama principal (trunk, troncal/tronco).

Otra característica de los sistemas de control de versiones es la posibilidad de marcar revisiones particulares (por ejemplo, una versión lanzada a producción), para que pueda en cualquier momento recrear un cierto entorno o compilación. Este proceso se conoce como *etiquetar*.

Subversion no tiene comandos especiales para hacer ramas o etiquetas, pero en cambio utiliza lo que se denomina “copias baratas”. Las copias baratas son similares a los vínculos duros de Unix, que significa que en vez de hacer una copia completa en el repositorio, se crea un vínculo interno, apuntando a una revisión y árbol específicos. Como resultado, las ramas y las etiquetas son muy rápidas de crear, y casi no conllevan espacio extra en el repositorio.

### 4.20.1. Crando una rama o etiqueta

Si ha importado su proyecto con la estructura de directorios recomendados, crear una rama o una etiqueta es muy simple:



**Figura 4.54. El diálogo Rama/Etiqueta**

Seleccione la carpeta en su copia de trabajo de la que desea hacer una rama o una etiqueta, y luego seleccione el comando TortoiseSVN → Rama/Etiqueta....

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

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

you might now use something like

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

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



## intermediate folders

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

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

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

Ahora debe elegir el origen de la copia. Aquí tiene tres opciones:

### Revisión HEAD en el repositorio

La nueva rama se copia directamente en el repositorio desde la revisión HEAD. No se necesita transferir datos desde su copia de trabajo, y la rama se crea muy rápidamente.

### Revisión específica en el repositorio

La nueva rama se copia directamente en el repositorio, pero puede elegir una versión anterior. Esto es útil si se olvidó de crear una etiqueta cuando lanzó una versión de su proyecto la semana pasada. Si no puede acordarse del número de revisión, pulse el botón a la derecha para mostrar el registro de revisiones, y seleccione el número de revisión desde allí. De nuevo no se transfiere datos desde su copia de trabajo, y la rama se crea muy rápidamente.

### Copia de trabajo

La nueva rama es una copia idéntica de su copia de trabajo local. Si ha cambiado algunos archivos a una revisión anterior en su copia de trabajo, o si ha hecho cambios locales, esto es exactamente lo que irá a la copia. Naturalmente, esta clase de etiquetado complejo conlleva transferir datos desde su copia de trabajo al repositorio si no existe ya allí.

Si desea que su copia de trabajo se cambie automáticamente a la rama recién creada, utilice la casilla **Cambiar la copia de trabajo a la nueva rama/etiqueta**. Pero si lo hace, asegúrese primero que su copia de trabajo no contenga modificaciones. Si las tiene, esos cambios se mezclarán en la copia de trabajo de la rama cuando se haga el cambio.

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
Revisión HEAD en el repositorio	external's repos HEAD revision
Revisión específica en el repositorio	external's repos HEAD revision
Copia de trabajo	external's WC BASE revision

**Tabla 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.

Pulse **Aceptar** para confirmar la nueva copia al repositorio. No se olvide de proporcionar un mensaje de registro. Tenga en cuenta que la copia se crea *dentro del repositorio*.

Tenga en cuenta que, salvo que haya optado por cambiar su copia de trabajo a la rama recién creada, crear una etiqueta o una rama *no* afecta a su copia de trabajo. Incluso si creó la rama desde su copia de trabajo, estos cambios se confirmarán en la rama nueva, no en el tronco, así que su copia de trabajo todavía se marcará como modificada respecto al tronco.

### 4.20.2. Otras formas de crear una rama o etiqueta.

Puede también crear una rama o etiqueta sin tener una copia de trabajo. Para ello, abra el navegador de repositorio. Allí puede arrastrar carpetas a un nuevo sitio. Debe mantener pulsada la tecla **Ctrl** mientras arrastra para crear una copia, si no la carpeta será movida, no copiada.

Puede también arrastrar una carpeta con el botón derecho del ratón. Una vez suele el botón del ratón podrá elegir en el menú contextual si quiere mover o copiar la carpeta. Por supuesto, para crear una rama o etiqueta debe copiar la carpeta, no moverla.

Otra manera es a partir del diálogo de log. Puedes mostrar el diálogo de log para por ejemplo el trunk, seleccionar una revisión (ya sea la revisión HEAD que se encuentra al principio u otra anterior), hacer clic con el botón derecho y escoger create branch/tag from revision....

### 4.20.3. Obtener o cambiar...

...esa (realmente no) es la cuestión. Mientras que obtener descarga todo de la rama elegida a su directorio de trabajo, TortoiseSVN → **Cambiar...** sólo transfiere los datos cambiados a su copia de trabajo. Bueno para la carga de la red, bueno para su paciencia. :-)

Para poder trabajar con su rama o etiqueta recién generada tiene varias opciones. Puede:

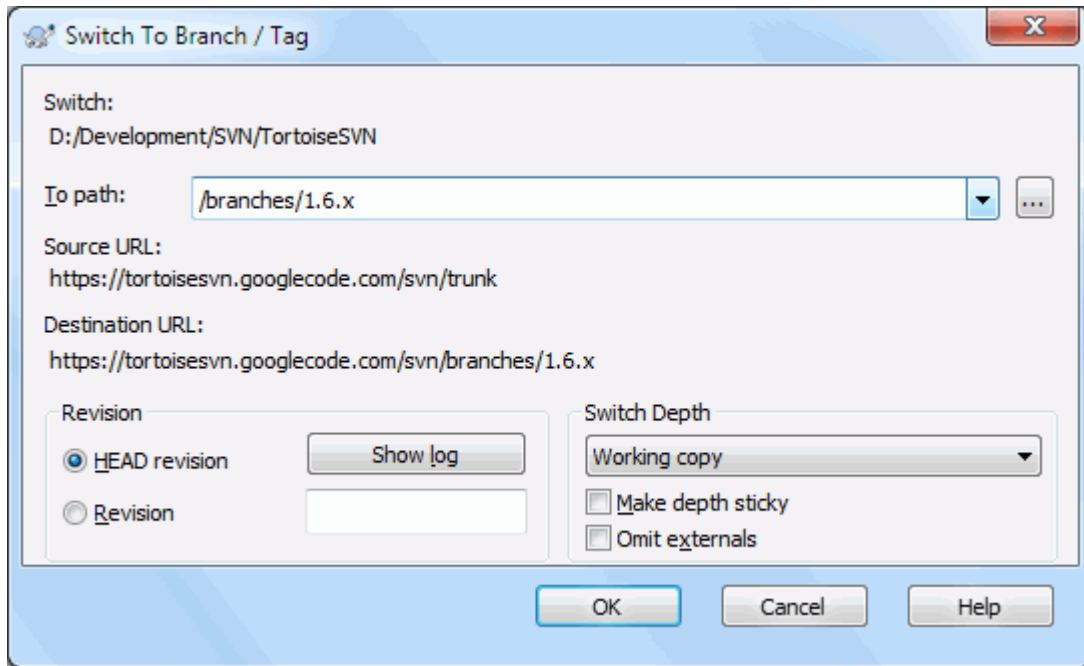
- TortoiseSVN → **Obtener** para obtener una copia nueva en una carpeta vacía. Puede obtener la copia de trabajo en cualquier parte de su disco duro, y puede crear tantas copias de trabajo de su repositorio como desee.
- Cambiar su copia de trabajo actual a la copia recién creada en el repositorio. De nuevo seleccione la carpeta superior de su proyecto y utilice TortoiseSVN → **Cambiar...** del menú contextual.

En el siguiente diálogo, introduzca la URL de la rama que acaba de crear. Deje la opción **Revisión HEAD** activada y pulse **Aceptar**. Su copia de trabajo se cambia a la nueva rama/etiqueta.

Cambiar trabaja igual que Actualizar, en el sentido que nunca pierde sus cambios locales. Cualquier cambio que haya hecho a su copia de trabajo que todavía no se hayan confirmado se fusionarán cuando haga el Cambio. Si no desea que esto ocurra, entonces debe o bien confirmar los cambios antes de cambiar, o revertir su copia de trabajo a una revisión ya-confirmada (típicamente HEAD).

- Si desea trabajar en el tronco (trunk) y en la rama, pero no desea el coste de una obtención nueva, puede utilizar el Explorador de Windows para hacer una copia de la copia de trabajo obtenida, y luego TortoiseSVN → **Cambiar...** esa copia a su nueva rama.





**Figura 4.55. El diálogo Cambiar**

Aunque Subversion por sí mismo no hace ninguna distinción entre etiquetas y ramas, la forma en la que normalmente se usan difiere un poco.

- Las etiquetas se usan típicamente para crear una copia estática de un proyecto en una etapa concreta. Como tales normalmente no se utilizan para el desarrollo - eso es para lo que se utilizan las ramas, y por esa razón recomendamos la estructura del repositorio `/trunk /branches /tags` en primer lugar. Trabajando en una revisión etiquetada *no es una buena idea*, pero dado que sus archivos locales no están protegidos, no hay nada que le impida hacer esto por error. Sin embargo, si intenta confirmar a una ruta en el repositorio que contenga `/tags/` (en inglés), TortoiseSVN le avisará.
- Puede ser que necesite hacer más cambios a una versión que ya había etiquetado. La forma correcta de manejar esta situación es crear primero una nueva rama desde la etiqueta. Haga sus cambios en esta rama, y luego cree una nueva etiqueta para esta rama, por ejemplo `Version_1.0.1`.
- Si modifica una copia de trabajo creada desde una rama y confirma, entonces los cambios irán a la nueva rama y *no* en el tronco. Sólo se guardan las modificaciones. El resto continúa siendo una copia barata.

## 4.21. Fusionando

Mientras que las ramas se utilizan para mantener líneas de desarrollo separadas, en alguna etapa tendrá que fusionar los cambios hechos en una rama de vuelta en el tronco, o viceversa.

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.

El siguiente punto a tener en cuenta es que la fusión *siempre* tiene lugar en una copia de trabajo. Si quiere fusionar cambios *en* una rama, deberá tener una copia de trabajo obtenida de dicha rama, e invocar el asistente de fusionado desde esa copia de trabajo utilizando TortoiseSVN → Fusionar...

En general es una buena idea realizar fusiones en una copia de trabajo sin modificar. Si ha hecho otros cambios en su copia de trabajo, confírmelos primero. Si la fusión no funciona como espera, puede querer revertir los cambios, y el comando `Revertir` descartará *todos* los cambios, incluidos cualquiera que haya hecho antes de la fusión.

Hay tres casos de uso comunes para fusionar que se manejan de formas ligeramente diferentes, como se describen a continuación. La primera página del asistente de fusión le pide seleccionar el método que necesita.

Fusionando un rango de revisiones.

Este método cubre el caso en el que ha hecho una o más revisiones a una rama (o al tronco) y desea portar estos cambios a una rama 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.

Fusionando dos árboles 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. Fusionando un rango de revisiones

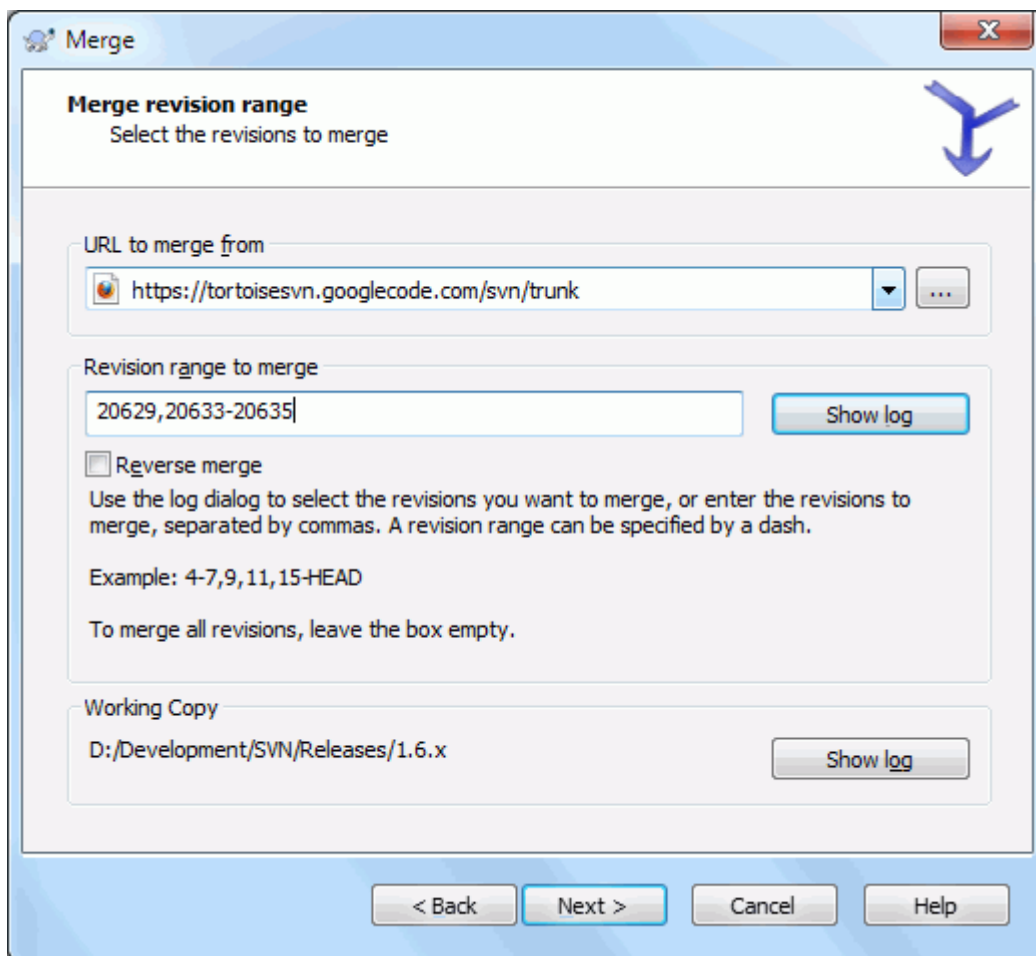


Figura 4.56. El asistente de fusión - Seleccionar el rango de revisiones

En el campo **Desde**: introduzca la URL completa de la carpeta de la rama o la etiqueta que contiene los cambios que desea portar a su copia de trabajo. También puede pulsar ... para navegar el repositorio y encontrar la rama deseada. Si ha fusionado desde esta rama antes, entonces utilice la lista desplegable que le muestra una historia de las URLs utilizadas anteriormente.

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.

En el campo **Rango de revisiones a fusionar** introduzca la lista de revisiones que desea fusionar. Puede ser una única revisión, una lista de revisiones específicas separadas por comas, un rango de revisiones separado por un guión, o cualquier combinación de éstas.

If you need to specify a peg revision for the merge, add the peg revision at the end of the revisions, e.g. 5-7, 10@3. In the above example, the revisions 5,6,7 and 10 would be merged, with 3 being the peg revision.



## Importante

Hay una diferencia importante en la forma que se especifican los rangos de revisiones en TortoiseSVN comparado con el cliente de línea de comandos. La forma más fácil de visualizar esto es pensar en una valla con postes y paneles.

Con el cliente de línea de comandos se especifican los cambios a fusionar utilizando dos revisiones “postes” que especifican los puntos *anterior* y *posterior*.

Con TortoiseSVN se especifican los conjuntos de cambios a fusionar utilizando “paneles de vallas”. La razón para esto queda más clara cuando utiliza el diálogo de registro para seleccionar las revisiones a fusionar, donde cada revisión aparece como un conjunto de cambios.

If you are merging revisions in chunks, the method shown in the Subversion book will have you merge 100-200 this time and 200-300 next time. With TortoiseSVN you would merge 100-200 this time and 201-300 next time.

Esta diferencia ha generado un montón de reacciones en las listas de trabajo. Reconocemos que es una diferencia respecto al cliente de línea de comandos, pero creemos que para la mayoría de usuarios GUI el método que hemos implementado es más fácil de entender.

La manera más sencilla de seleccionar el rango de revisiones que necesita es pulsar en **Mostrar Registro**, dado que esto mostrará los cambios recientes con sus comentarios de registro. Si desea fusionar los cambios de una única revisión, seleccione esa revisión. Si desea fusionar los cambios de varias revisiones, entonces seleccione ese rango (utilizando el modificador habitual **Mayúsculas**). Pulse **Aceptar** y la lista de números de revisión a fusionar se rellenará automáticamente.

Si desea fusionar los cambios de nuevo *fuera* de su copia de trabajo, para revertir un cambio que ya ha sido confirmado, seleccione las revisiones a revertir y asegúrese que la casilla **Fusión inversa** está marcada.

Si ya ha fusionado algunos cambios desde esta rama, esperemos que haya apuntado la última revisión fusionada en el mensaje de registro cuando confirmó ese cambio. En ese caso, puede utilizar **Mostrar Registro** en la Copia de Trabajo para buscar ese mensaje de registro. Recordando que estamos pensando en las revisiones como conjuntos de cambios, debería utilizar la revisión siguiente al punto final de la última fusión como punto de inicio para esta fusión. Por ejemplo, si la última vez fusionó las revisiones de la 37 a la 39, entonces el punto de inicio de esta fusión sería la revisión 40.

Si está utilizando las características de registro de fusiones de Subversion, no necesita recordar qué revisiones ya han sido fusionadas - Subversion lo apuntará por usted. Si deja el rango de revisiones en blanco, se incluirán todas las revisiones que no hayan sido aún fusionadas. Lea [Sección 4.21.5, “Registro de fusión”](#) para saber más.

When merge tracking is used, the log dialog will show previously merged revisions, and revisions pre-dating the common ancestor point, i.e. before the branch was copied, as greyed out. The **Hide non-mergeable revisions** checkbox allows you to filter out these revisions completely so you see only the revisions which *can* be merged.

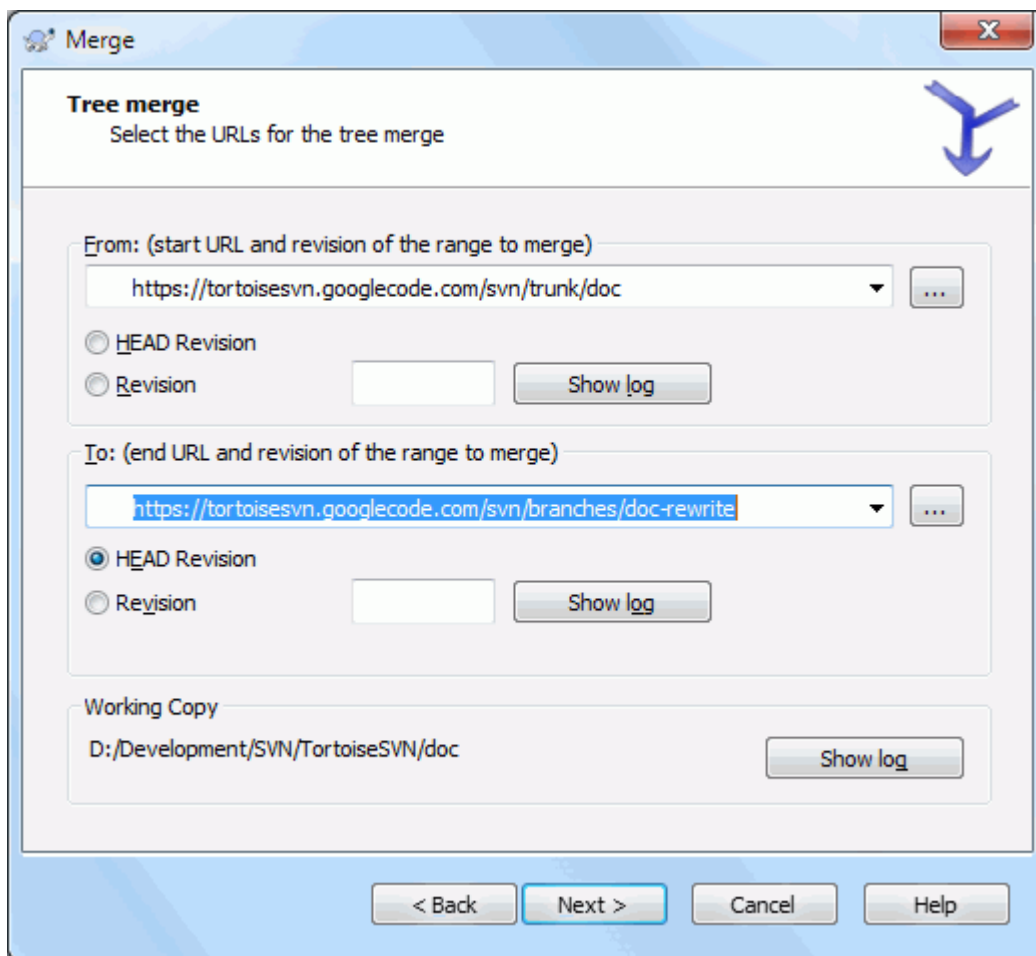
Si hay otras personas que puedan estar confirmando cambios, entonces tenga cuidado al utilizar la revisión HEAD. Puede que no se refiera a la revisión que usted cree si alguien más ha hecho una confirmación después de su última actualización.

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.

Hay algunas condiciones que se aplican a la fusión de reintegración. En primer lugar, el servidor debe soportar el registro de fusiones. La copia de trabajo debe ser de profundidad infinita (sin obtenciones parciales), y no debe tener ninguna modificación local, ítems apuntando a otra ruta o ítems que hayan sido actualizados a otras revisiones distintas de HEAD. Todos los cambios del tronco hechos durante el desarrollo de la rama deben haberse fusionado en la rama (o marcados como que ya han sido fusionados). El rango de revisiones para la fusión se calculará automáticamente.

Clic **Siguiente** y vaya a **Sección 4.21.3, “Opciones de fusión”**.

#### 4.21.2. Fusionando dos árboles diferentes



**Figura 4.57. El asistente de fusión - Fusión de árboles**

Si está utilizando este método para fusionar una rama de característica de nuevo en el tronco, necesitará iniciar el asistente de fusión desde una copia de trabajo del tronco.

En el campo Desde: introduzca la URL completa del *tronco* (trunk). Esto puede sonar erróneo, pero recuerde que el tronco es el punto de inicio al que desea añadir los cambios de la rama. También puede pulsar ... para navegar el repositorio.

En el campo A: introduzca la URL completa de la rama de la característica.

En los dos campos **Desde la Revisión** y **Hasta la Revisión**, introduzca el último número de revisión en el que se sincronizaron los dos árboles. Si está seguro que nadie más está haciendo confirmaciones puede utilizar HEAD en ambos casos. Si hay alguna posibilidad que alguien haya hecho una confirmación desde esa sincronización, utilice el número de revisión específico para evitar perder cambios más recientes.

También puede usar **Mostrar registro** para seleccionar la revisión.

### 4.21.3. Opciones de fusión

Esta página del asistente le permite especificar opciones avanzadas antes de empezar el proceso de fusión. La mayoría de las veces puede simplemente utilizar las opciones por defecto.

Puede especificar la profundidad de la fusión, es decir, cuántos niveles debe bajar la fusión en su copia de trabajo. Los términos de profundidad se describen en **Sección 4.3.1, “Profundidad de obtención”**. La profundidad por defecto es **Copia de trabajo**, que utiliza la configuración de profundidad existente, y que casi siempre es lo que quiere hacer.

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>].

Puede especificar la forma en la que se manejan los cambios en los finales de línea y en los espacios en blanco. Estas opciones se describen en **Sección 4.11.2, “Opciones de fin de línea y espacios en blanco”**. El comportamiento por defecto es tratar todos los espacios en blanco y las diferencias en los finales de líneas como cambios reales que deben fusionarse.

La casilla marcada como **Forzar la fusión** is usada para evitar conflictos de árbol en los que un borrado entrante afecta a archivos que o están modificados localmente o ni siquiera están versionados. Si el archivo es eliminado no hay forma de recuperarlo. Esa es la razón por la que esta opción no está marcada por defecto.

Si está utilizando el registro de fusiones y desea marcar una revisión como que ya se ha fusionado, sin hacer realmente efectiva la fusión aquí, marque la casilla **Sólo registrar la fusión**. Hay dos posibles razones por las que puede querer hacer esto. Puede ser que la fusión es demasiado complicada para los algoritmos de fusión, por lo que hace los cambios en el código a mano, y luego marca el cambio como fusionado para que el algoritmo de registro de fusión se de por enterado. O quizás quiera evitar que una revisión en concreto se fusione. Marcándola como ya fusionada evitará que la fusión se produzca con los clientes manejan la información de registro de fusión.

Ahora que todo está preparado, todo lo que debe hacer es pulsar en el botón **Fusionar**. Si quiere previsualizar los resultados, el botón **Probar fusión** realiza la operación de la fusión, pero *no* modifica la copia de trabajo en absoluto. Le muestra una lista de archivos que serán cambiados por la fusión real, y le notifica las áreas donde *podrían* ocurrir conflictos. Ya que el seguimiento de la fusión complica mucho el proceso de fusionado, no hay una forma garantizada de saber de antemano si el fusionado se completará sin conflictos, por lo que ficheros marcados como en conflicto en una prueba de fusión podrían en realidad fusionarse sin problemas.

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. Revisando los resultados de la fusión

La fusión ya ha concluído. Es una buena idea observar la fusión y comprobar si el resultado es el esperado. Normalmente fusionar es un proceso complicado. A menudo aparecen conflictos cuando la rama se ha desviado mucho del tronco.



## Sugerencia

Whenever revisions are merged into a working copy, TortoiseSVN generates a log message from all the merged revisions. Those are then available from the **Recent Messages** button in the commit dialog.

To customize that generated message, set the corresponding project properties on your working copy. See [Sección 4.18.3.10, “Merge log message templates”](#)

Los clientes y servidores de Subversion anteriores a la versión 1.5 no almacenan la información de fusión, y las revisiones fusionadas deben ser registradas de forma manual. Cuando haya comprobado los cambios y vaya a confirmar esta revisión, su mensaje de registro de confirmación debería incluir *siempre* los números de revisión que han sido portados en la fusión. Si desea aplicar otra fusión más tarde, necesitará saber lo que ya ha fusionado, porque no querrá portar un cambio más de una vez. Puede encontrar más información sobre esto en [Las mejores prácticas para fusionar](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] en el libro de Subversion.

Si su servidor y todos los clientes están ejecutando Subversion 1.5 o superior, las facilidades de registro de fusiones apuntarán las revisiones fusionadas y evitarán que una revisión se fusione más de una vez. Esto le facilita mucho la vida ya que puede simplemente fusionar el rango entero de revisiones cada vez y saber qué sólo las nuevas revisiones serán realmente fusionadas.

El manejo de las ramas es importante. Si desea mantener esta rama actualizada respecto al tronco, debería asegurarse de fusionar a menudo para que la rama y el tronco no se alejen mucho. Por supuesto, sigue debiendo evitar fusiones repetidas de cambios, como se explicó anteriormente.



## Sugerencia

Si acaba de fusionar una rama de característica en el tronco, el tronco contiene ahora todo el código de la nueva característica, y la rama es obsoleta. Si lo necesita, puede borrar la rama del repositorio.



## Importante

Subversion no puede fusionar un archivo con una carpeta, y viceversa - sólo carpetas con carpetas y archivos con archivos. Si pulsa en un archivo y abre el diálogo de fusionar, entonces deberá darle una ruta a un archivo en ese diálogo. Si selecciona una carpeta y llama al diálogo, debe especificar una URL de una carpeta para la fusión.

### 4.21.5. Registro de fusión

Subversion 1.5 introdujo facilidades para el registro de fusiones.. Cuando fusiona cambios de un árbol en otro, los números de revisión fusionados se almacenan, y esta información se puede utilizar para diferentes propósitos.

- Puede evitar el peligro de fusionar la misma revisión dos veces (problema de fusiones repetidas). Una vez que una revisión se ha marcado como que ha sido fusionada, las fusiones futuras que incluyan esa revisión en el rango la saltarán.
- Cuando fusiona una rama de nuevo al tronco, el diálogo de registro le puede mostrar las confirmaciones de la rama como parte del registro del tronco, proporcionando una mejor trazabilidad de los cambios.
- Cuando se muestra el diálogo de registro desde el diálogo de fusión, las revisiones ya fusionadas aparecen en gris.
- Cuando muestre información de autoría para un archivo, puede elegir mostrar el autor original de la revisión fusionada, en vez de la persona que hizo la fusión.
- Puede marcar revisiones como *no fusionar* si las incluye en la lista de revisiones fusionadas sin ejecutar realmente la fusión.

La información de registro de fusiones se almacena en la propiedad `svn:mergeinfo` por parte del cliente cuando realiza la fusión. Cuando la fusión se confirma, el servidor almacena dicha información en una base de datos, y cuando le pide información de la fusión, registro o autoría, el servidor puede responder apropiadamente. Para que el sistema funcione correctamente debe asegurarse que el servidor y todos los clientes estén actualizados. Los clientes antiguos no almacenarán la propiedad `svn:mergeinfo` y los servidores anteriores no proporcionarán la información que piden los nuevos clientes.

Find out more about merge tracking from Subversion's [Merge tracking documentation](http://svn.apache.org/repos/asf/Subversion/trunk/notes/merge-tracking/index.html) [http://svn.apache.org/repos/asf/Subversion/trunk/notes/merge-tracking/index.html].

#### 4.21.6. Handling Conflicts after Merge



### 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 4.58. 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 [Sección 4.6.3, “Conflictos de árbol”](#) 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 4.59. 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.

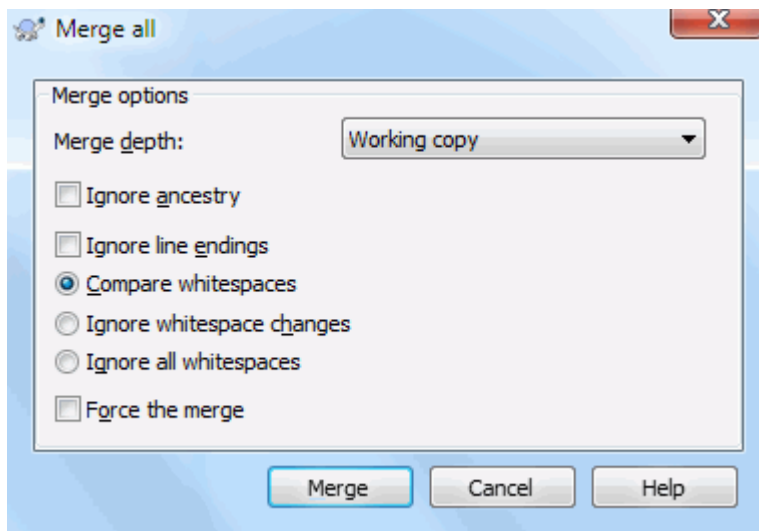
## 4.21.7. Mantenimiento de ramas de características

Cuando desarrolla una nueva característica en una rama separada es una buena idea mantener una política de re-integración cuando la característica está completa. Si se está modificando a la vez `trunk` (el tronco), puede que encuentre que las diferencias se vuelven significativas según pasa el tiempo, y la fusión se convierte en una pesadilla.

Si la característica es relativamente simple y el desarrollo no llevará mucho tiempo, puede adoptar un camino más simple, que es mantener la rama totalmente separada hasta que la característica se completa, y luego fusionar los cambios de la rama de nuevo en el tronco. En el asistente de fusión esto sería un simple **Fusionar un rango de revisiones**, donde el rango de revisiones es la rango de revisiones de la rama.

Si la característica va a llevar más tiempo y necesita tener en cuenta los cambios en `trunk`, entonces necesita mantener las ramas sincronizadas. Esto simplemente significa que periódicamente fusionará el tronco en la rama, para que la rama contenga todos los cambios de la rama *además* de la nueva característica. El proceso de sincronización utiliza **Fusionar un rango de revisiones**. Cuando la característica esté completa, podrá fusionarla de nueva en `trunk` utilizando o bien **Reintegrar una rama** o **Fusionar dos árboles distintos**.

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 4.60. The Merge-All Dialog

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



## 4.22. Bloqueando

Subversion generalmente trabaja mejor sin bloqueos, utilizando los métodos “Copiar-Modificar-Fusionar” que se describieron anteriormente en el [Sección 2.2.3, “La solución copiar-modificar-fusionar”](#). Sin embargo, hay algunas pocas situaciones en las que puede querer implementar alguna forma de política de bloqueo.

- Está utilizando archivos “no fusionables”, por ejemplo, archivos de imagen. Si dos personas cambian el mismo archivo, la fusión no es posible, así que alguno de ellos perderá sus cambios.
- Su compañía ha utilizado en el pasado un sistema de control de versiones bloqueante, y la dirección ha decidido que “bloquear es lo mejor”.

Primero necesita asegurarse que su servidor de Subversion está actualizado al menos a la versión 1.2. Las versiones anteriores no tienen ningún soporte de bloqueos. Si está usando el acceso `file://`, sólo necesitará actualizar sus clientes, por supuesto.



### The Three Meanings of “Lock”

In this section, and almost everywhere in this book, the words “lock” and “locking” describe a mechanism for mutual exclusion between users to avoid clashing commits. Unfortunately, there are two other sorts of “lock” with which Subversion, and therefore this book, sometimes needs to be concerned.

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

And third, files and folders can get locked if they're in use by another process, for example if you have a word document opened in Word, that file is locked and can not be accessed by TortoiseSVN.

You can generally forget about these other kinds of locks until something goes wrong that requires you to care about them. In this book, “lock” means the first sort unless the contrary is either clear from context or explicitly stated.

### 4.22.1. Cómo trabaja el bloqueo en Subversion

Por defecto, nada se bloquea y todo el mundo que tiene acceso de confirmación puede confirmar cambios a cualquier archivo en cualquier momento. Los demás actualizarán sus copias de trabajo periódicamente y los cambios en el repositorio se fusionarán con los cambios locales.

Si *Obtiene un Bloqueo* en un archivo, entonces sólo usted puede confirmar ese archivo. Las confirmaciones de los demás se bloquearán hasta que quite el bloqueo. Un archivo bloqueado no puede ser modificado de ninguna forma en el repositorio, por lo que no puede ser siquiera borrado o renombrado, excepto por el dueño del bloqueo.



### Importante

Un bloqueo no se asigna a un usuario específico, sino a un usuario específico y a una copia de trabajo. Tener un bloqueo en una copia de trabajo también impide al mismo usuario hacer un commit del fichero bloqueado desde otra copia de trabajo.

Por poner un ejemplo, imagina que el usuario Jon tiene un copia de trabajo en el PC de su oficina. Allí empieza a trabajar en una imagen, y por lo tanto adquiere el bloqueo de ese fichero. Cuando deja su oficina aún no ha terminado con ese fichero, por lo que no libera dicho bloqueo. De vuelta a casa Jon también tiene una copia de trabajo y decide trabajar un poco más en el proyecto. Pero no puede modificar o hacer un commit de ese fichero de imagen, puesto que el bloqueo para el fichero se encuentra en la copia de trabajo de su oficina.

Sin embargo, los demás usuarios no necesariamente saben que usted ha obtenido un bloqueo. A menos que ellos comprueben el estado de bloqueo regularmente, la primera vez que sabrán sobre él es cuando sus confirmaciones fallen, lo que muchas veces no es muy útil. Para hacer más fácil el manejo de bloqueos, hay una nueva propiedad de Subversion `svn:needs-lock`. Cuando se establece esta propiedad (a cualquier valor) en un archivo, siempre que el archivo se obtiene o actualiza, la copia local se deja como sólo-lectura *a menos* que la copia de trabajo tenga un bloqueo para ese archivo. Esto actúa como una advertencia indicando que no debería modificar el archivo a menos que obtenga un bloqueo. Los archivos que están versionados y marcados como sólo-lectura se marcan con un ícono superpuesto especial en TortoiseSVN para indicar que necesita obtener un bloqueo antes de editar.

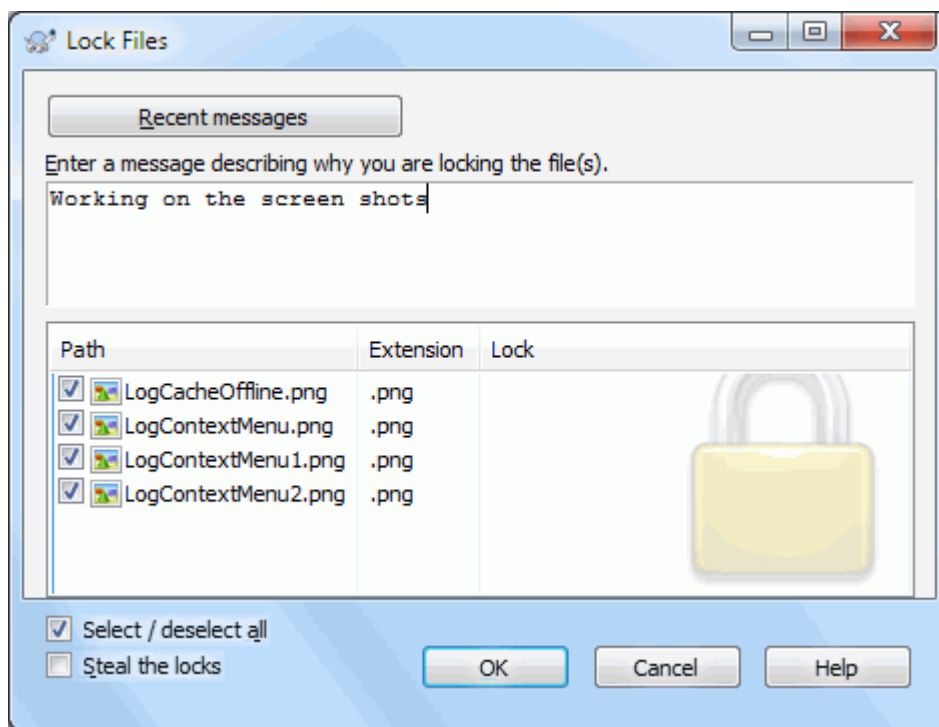
Los bloqueos se graban con el lugar de la copia local y también con el propietario. Si tiene varias copias de trabajo (en casa, en el trabajo) entonces sólo puede obtener un bloqueo en *una* de esas copias de trabajo.

Si uno de sus compañeros de trabajo obtiene un bloqueo y luego se va de vacaciones sin quitarlo, ¿qué hace? Subversion proporciona un método para forzar bloqueos. Quitar un bloqueo que tiene otra persona se define como *Romper* el bloqueo, y obtener forzosamente un bloqueo que tiene otro se define como *Robar* el bloqueo. Naturalmente estas no son cosas que hacer a la ligera si desea continuar siendo amigo de sus compañeros de trabajo.

Los bloqueos se guardan en el repositorio, y se crea un token de bloqueo en su copia local de trabajo. Si hay una discrepancia, por ejemplo si alguien ha roto el bloqueo, el token local se convierte en inválido. El repositorio siempre es la referencia definitiva.

### 4.22.2. Obteniendo un bloqueo

Seleccione el archivo o archivos en su copia de trabajo para los que desee obtener un bloqueo, y seleccione el comando TortoiseSVN → Obtener Bloqueo....



**Figura 4.61. El diálogo Bloquear**

Aparece un diálogo, que le permite introducir un comentario, para que los demás vean por qué ha bloqueado el archivo. El comentario es opcional, y de momento sólo se utiliza con repositorios basados en Svnserve. Si (y sólo si) necesita robar el bloqueo de alguien, marque la casilla **Robar bloqueo**, y luego pulse **Aceptar**.

Puede activar la propiedad `tsvn:logtemplatelock` para proporcionar una plantilla de mensaje para que la rellenen los usuarios como el mensaje del bloqueo. Consulte [Sección 4.18, “Configuración del proyecto”](#) para las instrucciones sobre cómo activar las propiedades.

Si selecciona una carpeta y luego utiliza TortoiseSVN → Obtener Bloqueo... se abrirá el diálogo de bloquear con *todos* los archivos en *todas* las subcarpetas seleccionados para bloquearlos. Si realmente quiere bloquear una jerarquía completa, ésta es la forma de conseguirlo, pero puede ganarse la antipatía de sus compañeros de trabajo si les bloquea todo el proyecto. Utilícelo con cuidado...

### 4.22.3. Quitando un bloqueo

Para asegurarse que no se olvida de quitar un bloqueo que no necesita más, los archivos bloqueados se muestran en el diálogo de confirmar y se seleccionan por defecto. Si continúa con la confirmación, los bloqueos que tenga sobre los archivos seleccionados se quitan, incluso si los archivos no se han modificado. Si no desea quitar el bloqueo en algunos archivos, puede desmarcarlos (si no están modificados). Si desea mantener los bloqueos en un archivo que ha modificado, tiene que habilitar la casilla Mantener bloqueos antes de confirmar sus cambios.

Para quitar un bloqueo manualmente, seleccione el archivo o archivos de su copia de trabajo para los que desee quitar el bloqueo, y luego seleccione el comando TortoiseSVN → Quitar Bloqueo. No hay nada más que introducir por lo que TortoiseSVN contactará con el repositorio y quita los bloqueos. También puede utilizar este comando en una carpeta para quitar todos los bloqueos recursivamente.

### 4.22.4. Comprobando el estado de los bloqueos

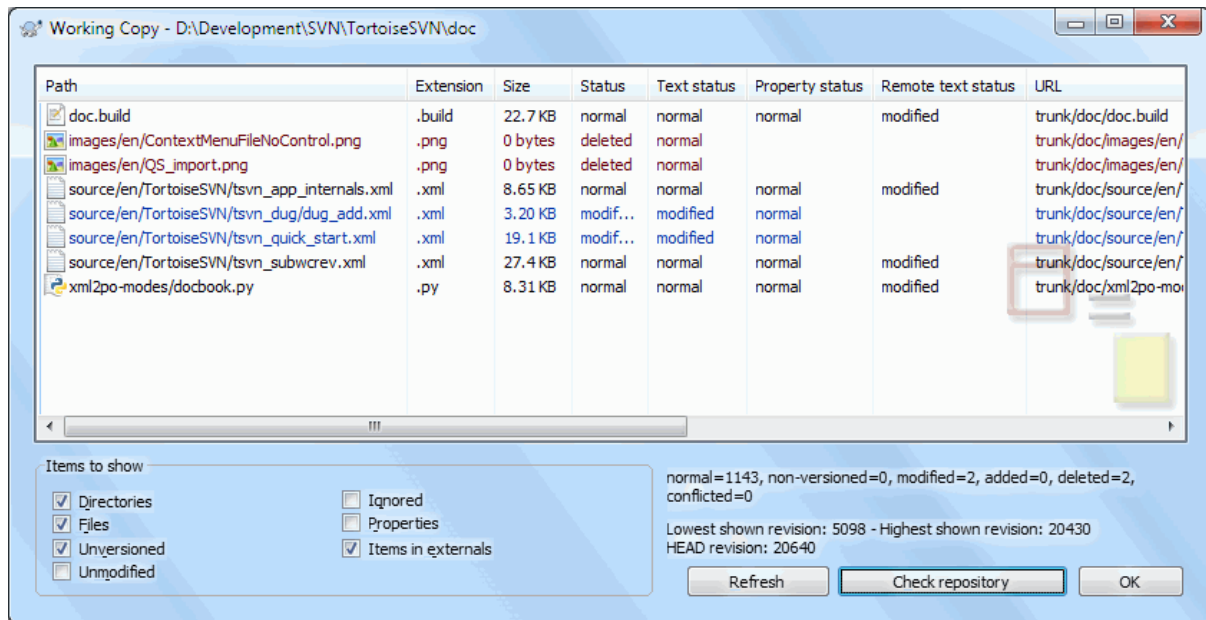


Figura 4.62. El diálogo Comprobar modificaciones

Para ver qué bloqueos tiene usted y los de los demás, puede utilizar el comando TortoiseSVN → Comprobar Modificaciones.... Los tokens de bloqueos que se tienen localmente se muestran inmediatamente. Para comprobar los bloqueos que tienen los demás (y para ver si alguno de sus bloqueos se han roto o han sido robados) necesita pulsar Comprobar Repositorio.

Desde el menú contextual aquí, también puede obtener y quitar bloqueos, además de romper y robar bloqueos que tienen otros.



### Evite romper y robar bloqueos

Si ha roto o robado el bloqueo de otro sin decírselo, puede causar una pérdida potencial de trabajo. Si está trabajando con tipos de archivos no fusionables y roba el bloqueo de otro, una vez que usted

quite el bloqueo ellos son libres de confirmar sus cambios y sobrescribir los suyos. Subversion no pierde datos, pero ha perdido la protección para trabajo en equipo que el bloqueo le proporcionaba.

#### 4.22.5. Configurando los archivos no-bloqueados como de sólo-lectura

Como se menciona arriba, la manera más efectiva de utilizar los bloqueos es establecer la propiedad `svn:needs-lock` en los archivos. Consulte [Sección 4.18, “Configuración del proyecto”](#) para obtener instrucciones sobre cómo establecer propiedades. Los archivos con esta propiedad establecida siempre se obtendrán y actualizarán con la marca de sólo-lectura a menos que su copia de trabajo tenga un bloqueo.



Como recordatorio, TortoiseSVN utiliza un ícono de superpuesto especial para indicarlo.

Si opera una política donde todos los archivos han de ser bloqueados, puede que encuentre más fácil utilizar la característica de Subversion auto-props para establecer la propiedad automáticamente cada vez que añada nuevos archivos. Para más información, lea [Sección 4.18.1.5, “Establecer propiedades automáticamente”](#).

#### 4.22.6. Los scripts ganchos de bloqueo

Cuando crea un nuevo repositorio con Subversion 1.2 o superior, se crean cuatro plantillas en el directorio `hooks` del repositorio. Éstos se llaman antes y después de obtener un bloqueo, y antes y después de quitar un bloqueo.

Es una buena idea instalar un script gancho `post-lock` y `post-unlock` en el servidor que envíe un email indicando el archivo que se ha bloqueado. Poniendo este script en su sitio, todos sus usuarios serán informados si alguien bloquea/quita el bloqueo de un archivo. Puede encontrar un script gancho de ejemplo `hooks/post-lock.tmpl` en su carpeta del repositorio.

También puede utilizar ganchos para no permitir romper o robar bloqueos, o quizás para limitarlo a un administrador designado para ello. O quizás desea enviar un email al propietario cuando uno de sus bloqueos se haya roto o robado.

Para más información, lea la [Sección 3.3, “Scripts gancho en el lado del servidor”](#).

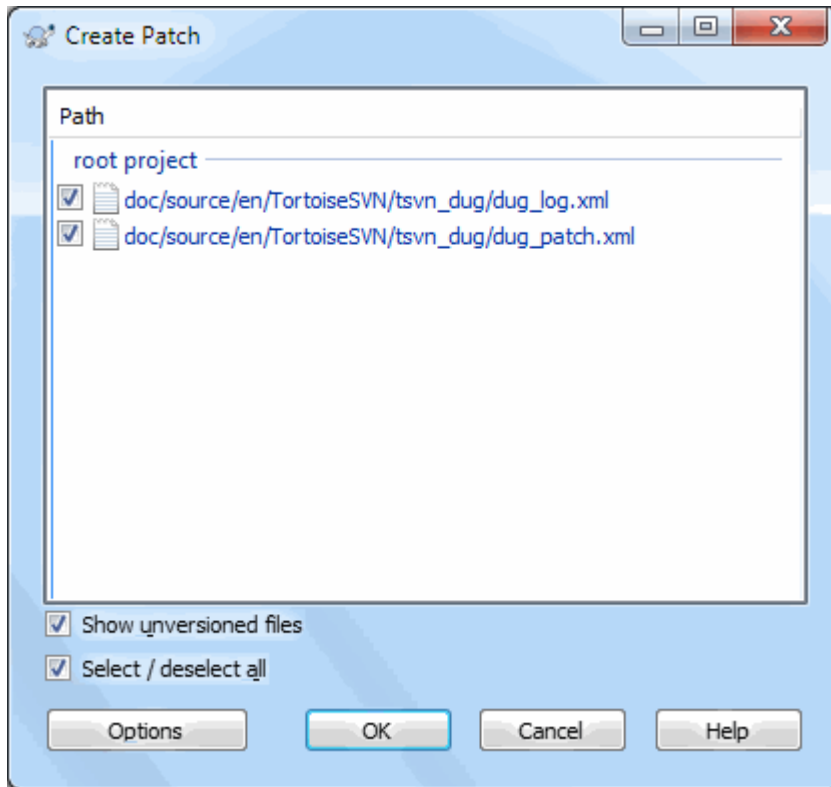
### 4.23. Creando y aplicando parches

En proyectos de código abierto (como éste), todos tienen acceso de lectura al repositorio, y cualquiera puede hacer una contribución al proyecto. ¿Así que cómo se controlan esas contribuciones? Si cualquiera pudiera confirmar cambios, el proyecto estaría permanentemente inestable y probablemente roto de forma permanente. En esta situación, el cambio se maneja mediante un archivo de *parche* enviado al equipo de desarrollo, que tienen acceso de escritura. Ellos pueden revisar el parche antes, y luego o bien confirmarlo en el repositorio o devolvérselo al autor.

Los archivos de parche son simplemente archivos de diff unificados que muestran las diferencias entre su copia de trabajo y la revisión base.

#### 4.23.1. Creando un archivo parche

Primero necesita hacer y *probar* sus cambios. Luego en vez de utilizar el comando TortoiseSVN → Confirmar... sobre la carpeta padre, seleccione TortoiseSVN → Crear Parche...



**Figura 4.63. El diálogo Crear parche**

ahora puede seleccionar los archivos que desea incluir en el parche, del mismo modo a como lo haría con una confirmación completa. Esto producirá un único archivo que contendrá un resumen de todos los cambios que ha hecho a los archivos seleccionados desde la última vez que se actualizó desde el repositorio.

Las columnas en este diálogo pueden ser personalizadas de la misma forma que las columnas en el diálogo Comprobar modificaciones. Para más detalles, lea [Sección 4.7.3, “Estado local y remoto”](#).

By clicking on the Options button you can specify how the patch is created. For example you can specify that changes in line endings or whitespaces are not included in the final patch file.

Puede producir parches separados que contengan cambios a diferentes conjuntos de archivos. Por supuesto, si crea un archivo de parche, hace algunos cambios más a los *mismos* archivos y luego crea otro parche, el segundo archivo de parche incluirá *ambos* conjuntos de cambios.

Simplemente grabe el archivo utilizando un nombre de archivo de su elección. Los archivos de parche pueden tener la extensión que desee, pero por convención se suele utilizar las extensiones `.patch` o `.diff` extension. Ya está preparado para enviar su archivo de parches.



### Sugerencia

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.

También puede almacenar el parche en el portapapeles en vez de en un archivo. Puede querer hacer esto para poder pegarlo en un email de forma que lo revisen otros. O si tiene dos copias de trabajo en una máquina y quiere transferir cambios de una a otra, un parche en el portapapeles es una forma cómoda de hacerlo.

If you prefer, you can create a patch file from within the Commit or Check for Modifications dialogs. Just select the files and use the context menu item to create a patch from those files. If you want to see the Options dialog you have to hold **shift** when you right click.

### 4.23.2. Aplicando un archivo parche

Los archivos de parches se aplican en su copia de trabajo. Esto debe hacerse desde el mismo nivel de carpetas que se utilizó para crear el parche. Si no está seguro de cuál es, mire la primera línea del archivo de parche. Por ejemplo, si el primer archivo en el que se trabajó era `doc/source/english/chapter1.xml` y la primera línea en el archivo de parche es `Index: english/chapter1.xml` entonces necesita aplicar el parche en la carpeta `english`. Sin embargo, suponiendo que esté en la copia de trabajo correcta, si ha seleccionado un nivel de carpeta erróneo, TortoiseSVN se dará cuenta y le sugerirá el nivel correcto.

Para aplicar un archivo de parche en su copia de trabajo, necesita al menos acceso de lectura al repositorio. La razón para esto es que el programa de fusión debe referenciar los cambios sobre la revisión contra la que se hicieron por el desarrollador remoto.

Desde el menú contextual de esa carpeta, pulse en **TortoiseSVN → Aplicar Parche...** Esto mostrará un diálogo de abrir archivo que le permitirá seleccionar el archivo de parche a aplicar. Por defecto sólo aparecen los archivos `.patch` o `.diff`, pero puede elegir “Todos los archivos”. Si había guardado previamente un parche en el portapapeles, puede usar **Abrir desde el portapapeles...** en el diálogo de abrir archivo. Note que esta opción sólo está disponible si guardó el parche en el portapapeles usando **TortoiseSVN → Crear Parche....** Copiar un parche al portapapeles desde otra aplicación no hará que el botón aparezca.

Alternativamente, si el archivo de parche tiene una extensión `.patch` o `.diff`, puede hacer click con el botón derecho en él directamente y seleccionar **TortoiseSVN → Aplicar Parche...** En este caso se le preguntará la ruta de la copia de trabajo.

Estos dos métodos le ofrecen formas diferentes de hacer lo mismo. Con el primer método selecciona la copia de trabajo y busca el archivo de parche. Con el segundo selecciona el archivo de parche y busca la copia de trabajo.

Una vez que haya seleccionado el archivo de parche y la ruta de la copia de trabajo, se ejecuta **TortoiseMerge** para fusionar los cambios del archivo de parche contra su copia de trabajo. Una pequeña ventana le muestra los archivos que han sido cambiados. Haga doble click por turnos en cada uno de ellos, compruebe los cambios y grabe los archivos fusionados.

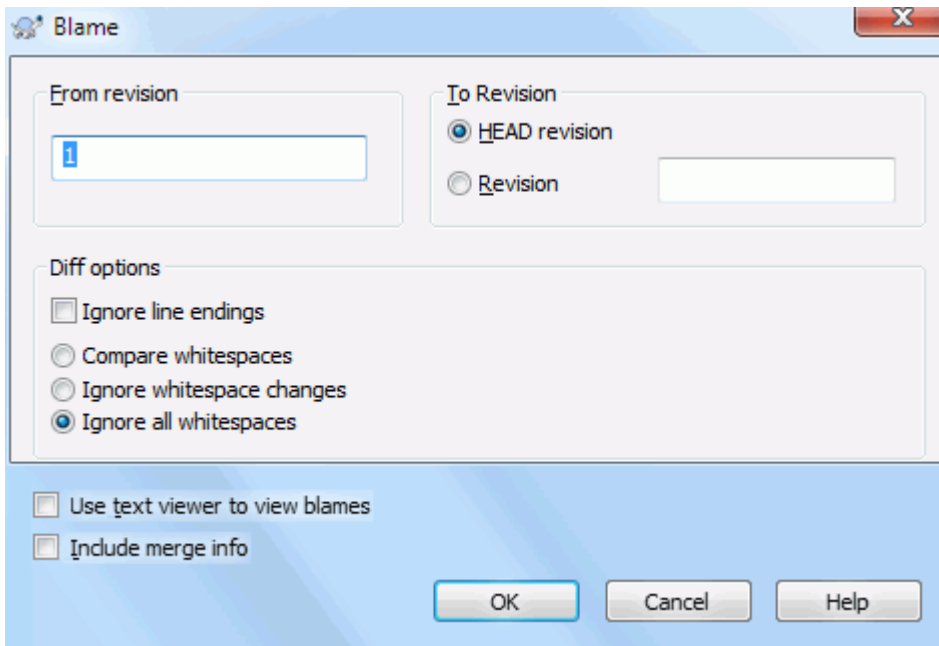
El parche del desarrollador remoto ya ha sido aplicado a su copia de trabajo, por lo que necesita confirmar para que todos los demás accedan a los cambios desde el repositorio.

### 4.24. ¿Quién cambió qué línea?

A veces necesita saber no sólo qué líneas han cambiado, sino también exactamente quién cambió líneas específicas en un archivo. Entonces es cuando el comando **TortoiseSVN → Autoría...**, a veces conocido como comando de *anotar*, tiene su utilidad.

Este comando muestra, por cada línea en un archivo, su autor y la revisión en la que se cambió la línea.

### 4.24.1. Autoría de archivos



**Figura 4.64. El diálogo Anotar / Autoría**

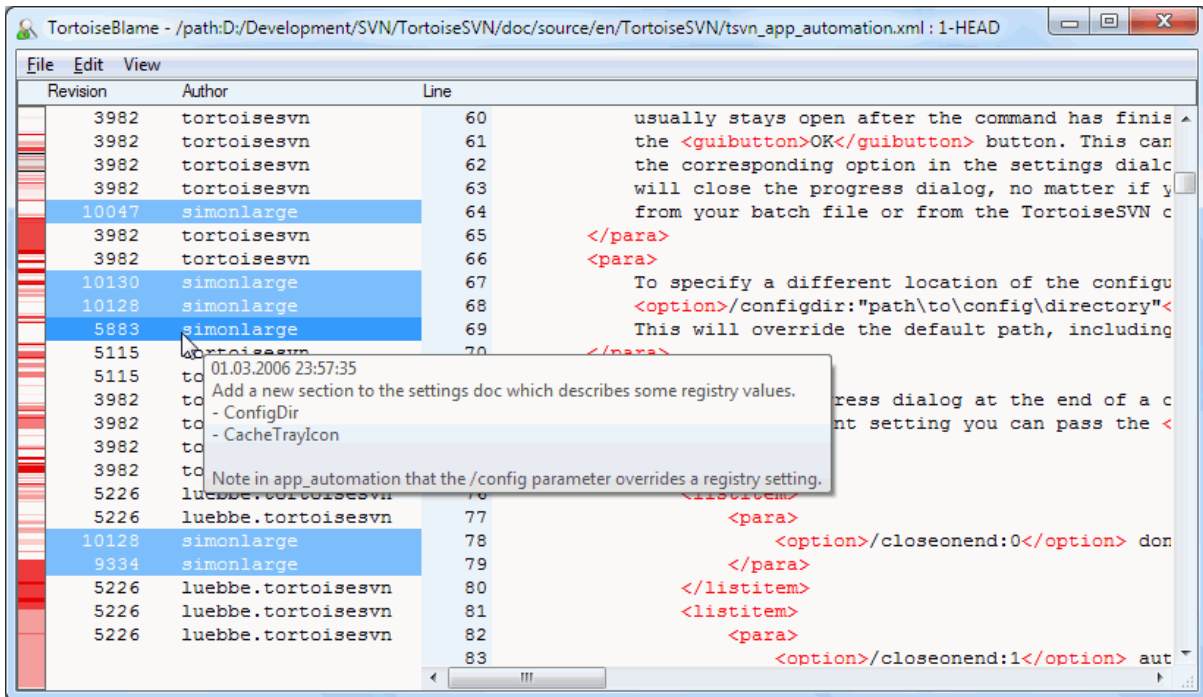
Si no está interesado en cambios de revisiones anteriores puede establecer la revisión desde la cual debe empezar la autoría. Ponga este valor a 1 si desea ver la autoría de *cada* revisión.

Por defecto, el fichero de autoría se ve utilizando *TortoiseBlame*, que resalta las diferentes revisiones para que sea más fácil de leer. Si desea imprimir o editar el fichero de autoría, seleccione *Utilizar visor de textos para ver autorías*.

Puede especificar la forma en la que se manejarán los cambios en los finales de línea y en los espacios en blanco. Estas opciones se describen en [Sección 4.11.2, “Opciones de fin de línea y espacios en blanco”](#). El comportamiento por defecto es tratar todas las diferencias en los espacios en blanco y en los finales de línea como cambios reales, pero si desea ignorar un cambio en la indentación y encontrar al autor original, puede elegir la opción apropiada aquí.

You can include merge information as well if you wish, although this option can take considerably longer to retrieve from the server. When lines are merged from another source, the blame information shows the revision the change was made in the original source as well as the revision when it was merged into this file.

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



**Figura 4.65. TortoiseBlame**

TortoiseBlame, que se incluye con TortoiseSVN, hace más fáciles de leer los archivos de autoría. Cuando pasa el ratón por encima de una línea en la columna de información de autoría, todas las líneas con la misma revisión se señalan con un fondo más oscuro. Las líneas de otras revisiones que fueron cambiadas por el mismo autor se señalan con un fondo claro. Los colores pueden no funcionar de forma muy eficiente si su pantalla está en el modo de 256 colores.

Si hace click en una línea, todas las líneas con la misma revisión se señalan, y las líneas de otras revisiones del mismo autor se señalan en un color más claro. Este señalado se mantiene, permitiéndole mover el ratón sin perder los señalados. Pulse en esa revisión de nuevo para desactivar el señalado.

Los comentarios de la revisión (mensaje de registro) se muestra en un texto de ayuda cuando se pasa el cursor sobre la columna de información de autoría. Si quiere copiar el mensaje de registro de esa revisión, utilice el menú contextual que aparece cuando hace click con el botón derecho sobre la columna de información de autoría.

Puede buscar dentro del informe de Autoría utilizando Edición → Buscar.... Esto le permite buscar por números de revisión, autores y por el contenido del archivo en si mismo. Los mensajes de registro no se incluyen en la búsqueda - deberá utilizar el diálogo de Mostrar Registro para buscar en ellos.

También puede ir directamente a un número de línea concreto utilizando Editar → Ir a la línea....

Cuando el ratón se coloca encima de las columnas de información de autoría, tiene disponible un menú contextual que le ayuda a comparar revisiones y examinar la historia, utilizando el número de revisión bajo el ratón como referencia. Menú contextual → Ver autoría de la revisión anterior genera un informe de autoría para el mismo archivo, pero utilizando la revisión anterior como límite superior. Esto le da el informe de autoría para el estado del archivo justo antes que la línea que está viendo cambiara. Menú contextual → Mostrar cambios inicia su visor de diferencias, mostrándole lo que cambió en la revisión referenciada. Menú contextual → Mostrar registro le muestra el diálogo de registro de revisiones empezando en la revisión referenciada.

Si necesita un indicador visual mejor de dónde están los cambios más antiguos y más nuevos, seleccione Ver → Colorear la antigüedad de las líneas. Esto utilizará un gradiente de color para mostrar las líneas más nuevas en



rojo y las más antiguas en azul. La paleta de color por defecto es bastante suave, pero puede cambiarla utilizando la configuración de TortoiseBlame.

If you are using Merge Tracking and you requested merge info when starting the blame, merged lines are shown slightly differently. Where a line has changed as a result of merging from another path, TortoiseBlame will show the revision and author of the last change in the original file rather than the revision where the merge took place. These lines are indicated by showing the revision and author in italics. The revision where the merge took place is shown separately in the tooltip when you hover the mouse over the blame info columns. If you do not want merged lines shown in this way, uncheck the **Include merge info** checkbox when starting the blame.

If you want to see the paths involved in the merge, select **View → Merge paths**. This shows the path where the line was last changed, excluding changes resulting from a merge.

The revision shown in the blame information represents the last revision where the content of that line changed. If the file was created by copying another file, then until you change a line, its blame revision will show the last change in the original source file, not the revision where the copy was made. This also applies to the paths shown with merge info. The path shows the repository location where the last change was made to that line.

Las configuraciones para TortoiseBlame se pueden acceder utilizando **TortoiseSVN → Configuración...** en la pestaña TortoiseBlame. Para más información, vea [Sección 4.31.9, “Configuración de TortoiseBlame”](#).

#### 4.24.2. Autoría de las diferencias

One of the limitations of the Blame report is that it only shows the file as it was in a particular revision, and the last person to change each line. Sometimes you want to know what change was made, as well as who made it. If you right click on a line in TortoiseBlame you have a context menu item to show the changes made in that revision. But if you want to see the changes *and* the blame information simultaneously then you need a combination of the diff and blame reports.

El diálogo del historial de revisiones incluye varias opciones que le permiten hacerlo.

##### Autoría de las revisiones

En el panel superior, seleccione dos revisiones, y luego seleccione **Menú Contextual → Ver autoría de las revisiones**. Esto obtendrá los datos de autoría de las dos revisiones, y luego utilizará el visor de diferencias para comparar los dos archivos de autoría.

##### Ver la autoría de los cambios

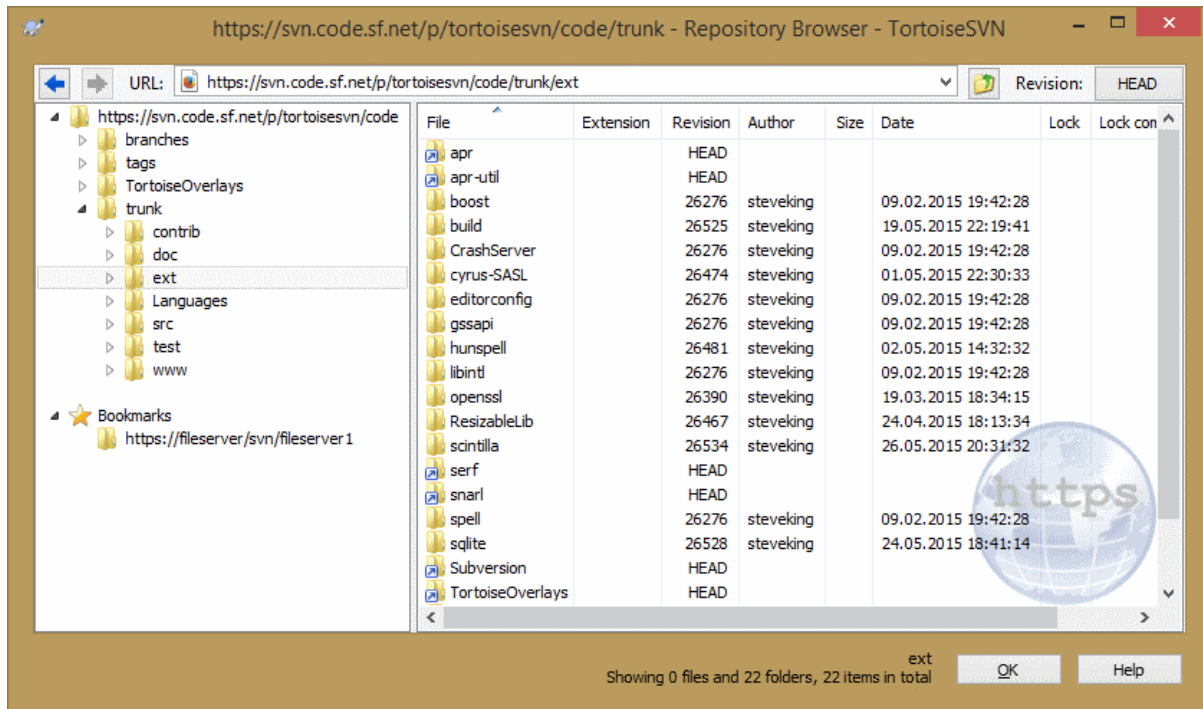
Seleccione una revisión en el panel superior, luego seleccione un archivo en el panel inferior y seleccione **Menú contextual → Autoría de los cambios**. Esto obtendrá los datos de autoría de la revisión seleccionada y la anterior, y utilizará el visor de diferencias para comparar los dos archivos de autoría.

##### Comparar y ver autoría con la BASE de trabajo

Muestre el registro de un único archivo, y en el panel superior, seleccione una única revisión, y luego seleccione **Menú contextual → Comparar y ver autoría con la BASE de trabajo**. Esto obtendrá los datos de autoría de la revisión seleccionada, y también del archivo en la BASE de trabajo, y utilizará el visor de diferencias para comparar los dos archivos de autoría.

### 4.25. El navegador de repositorios

A veces necesitará trabajar directamente en el repositorio, sin tener una copia de trabajo. Ésa es la razón por la que existe el *Navegador de Repositorios*. Igual que el Explorador y los íconos superpuestos le permiten ver su copia de trabajo, el navegador de repositorios le permite ver la estructura y el estado del repositorio.



**Figura 4.66. El navegador de repositorios**

Con el navegador de repositorios puede ejecutar comandos como copiar, mover, renombrar... directamente en el repositorio.

El navegador de repositorios se parecerá al explorador de Windows, excepto que está mostrando el contenido del repositorio en una revisión concreta en vez de los archivos de su ordenador. En el panel izquierdo puede ver un árbol de directorios, y en el panel derecho están los contenidos del directorio seleccionado. En la parte superior de la ventana del visor de repositorios puede introducir la URL del repositorio y la revisión que desea visualizar.

Las carpetas que contienen la propiedad `svn:externals` también se muestran en el navegador del repositorio. Estas carpetas se muestran con una pequeña flecha sobre ellas para indicar que no son parte de la estructura del repositorio, sino simplemente enlaces.

Al igual que el explorador de Windows, puede pulsar sobre los encabezados de las columnas en el panel derecho si desea establecer la ordenación. Y como en el explorador hay menús contextuales en ambos paneles.

El menú contextual para un archivo le permite:

- Abrir el archivo seleccionado, bien con el visor por defecto para ese tipo de archivo, o bien con el programa que elija.
- Edit the selected file. This will checkout a temporary working copy and start the default editor for that file type. When you close the editor program, if changes were saved then a commit dialog appears, allowing you to enter a comment and commit the change.
- Mostrar el registro de las revisiones para ese archivo, o mostrar un gráfico de todas las revisiones para que pueda ver de dónde vino el archivo.
- Ver la autoría del archivo, para saber quién cambió qué línea y cuándo.
- Checkout a single file. This creates a “sparse” working copy which contains just this one file.
- Eliminar o renombrar el archivo.
- Guardar una copia no versionada del archivo en su disco duro.
- Copia la URL que aparece en la barra de direcciones al portapapeles.

- Hacer una copia del archivo, bien a una parte diferente del repositorio, o bien a una copia de trabajo incluida dentro del mismo repositorio.
- Ver/editar las propiedades del archivo.
- Crear un acceso directo para que usted pueda comenzar de nuevo el navegador de repositorios, abierto directamente en esta ubicación.

El menú contextual para una carpeta le permite:

- Mostrar el registro de revisiones para la carpeta, o mostrar un gráfico de todas las revisiones para que pueda ver de dónde vino la carpeta.
- Exportar la carpeta a una copia local no versionada de su disco duro.
- Obtener la carpeta para producir una copia de trabajo local en su disco duro.
- Crear una nueva carpeta en el repositorio.
- Add unversioned files or folders directly to the repository. This is effectively the Subversion Import operation.
- Borrar o eliminar la carpeta.
- Hacer una copia de la carpeta, bien a una parte diferente del repositorio, o bien a una copia de trabajo dentro del mismo repositorio. Esto puede usarse para crear una rama/etiqueta sin necesidad de tener una copia de trabajo chequeada.
- Ver/Editar las propiedades de la carpeta.
- Marcar la carpeta para comparación. La carpeta marcada se mostrará en negrita.
- Comparar la carpeta con una carpeta marcada previamente, bien como diff unificado, o bien como una lista de archivos que pueden diferenciarse visualmente utilizando la herramienta de diferencias por defecto. Esto puede ser particularmente útil para comparar dos etiquetas, o el tronco y una rama para ver qué se ha cambiado.

Si selecciona dos carpetas en el panel derecho, puede ver las diferencias bien como diff unificado, o bien como una lista de archivos que pueden diferenciarse visualmente utilizando la herramienta de diferencias por defecto.

Si selecciona múltiples carpetas en el panel derecho, puede obtenerlos todos a la vez en una carpeta padre común.

Si selecciona dos etiquetas que se han copiado desde la misma raíz (típicamente, /trunk/), puede utilizar **Menú Contextual** → **Mostrar Registro...** para ver la lista de revisiones entre los dos puntos etiquetados.

External items (referenced using `svn:externals`) are also shown in the repository browser, and you can even drill down into the folder contents. External items are marked with a red arrow over the item.

Puede utilizar **F5** para refrescar la vista como siempre. Esto refrescará todo lo que se esté viendo actualmente. Si quiere precargar o refrescar la información para los nodos que aún no han sido abiertos, utilice **Ctrl-F5**. Después de eso, la expansión de cualquier nodo ocurrirá automáticamente sin que haya una pausa por la red mientras se recoge la información.

También puede utilizar el navegador de repositorios para operaciones de arrastrar-y-soltar. Si arrastra una carpeta desde el Explorador al navegador de repositorios, se importará esa carpeta en el repositorio. Tenga en cuenta que si arrastra múltiples items, se importarán en confirmaciones separadas.

Si desea mover un ítem dentro del repositorio, simplemente arrastre con el botón izquierdo el ítem a su nuevo lugar. Si desea crear una copia en vez de mover el ítem, entonces deberá **Ctrl**-arrastrar con el botón izquierdo. Cuando copia, el cursor tiene un símbolo “más” en él, tal y como lo hace en el Explorador.

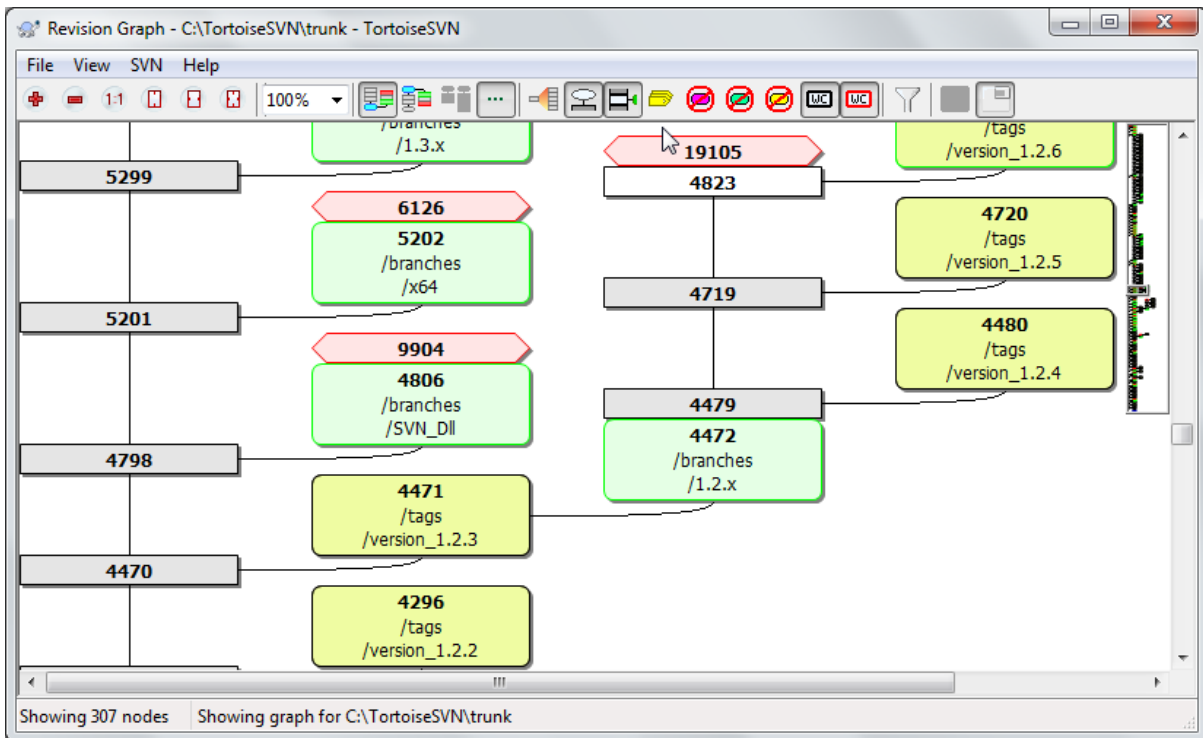
Si desea copiar/mover un archivo o una carpeta a otro lugar y darle también otro nombre a la vez, puede arrastrar con el botón derecho o **Ctrl**-arrastrar con el botón derecho el ítem en vez de arrastrarlo con el botón izquierdo. En ese caso, aparece un diálogo en el que puede introducir un nuevo nombre para el archivo o la carpeta.

Siempre que haga cambios en el repositorio utilizando uno de estos métodos, se le presentará un diálogo para introducir un mensaje de registro. Si ha arrastrado algo por error, esta es su oportunidad para cancelar la acción.

A veces cuando intenta abrir una ruta obtendrá un mensaje de error en vez de los detalles del ítem. Esto puede ocurrir si ha especificado una URL inválida, si no tiene permisos de acceso, o si hay algún otro problema en el servidor. Si necesita copiar este mensaje para incluirlo en un email, simplemente haga click con el botón derecho sobre él y utilice Menú Contextual → Copiar mensaje de error al portapapeles, o simplemente utilice **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. Gráficos de revisión



**Figura 4.67. Un gráfico de revisiones**

A veces necesita saber desde qué revisión del tronco se tomaron las ramas y las etiquetas, y la forma ideal de ver este tipo de información es en un gráfico o una estructura de árbol. Ahí es cuando necesita utilizar TortoiseSVN → Gráfico de Revisiones...

Este comando analiza la historia de las revisiones e intenta crear un árbol mostrando los puntos en los que se tomaron las copias, y cuando se borraron las ramas/etiquetas.



### Importante

Para generar el gráfico, TortoiseSVN debe obtener todos los mensajes de registro desde la raíz del repositorio. No es necesario decir que esto puede llevar varios minutos incluso con un repositorio con unos pocos miles de revisiones, dependiendo de la velocidad del servidor, el ancho de banda de la red, etc. Si intenta esto con algo como el proyecto *Apache*, que actualmente tiene más de 500.000 revisiones, puede estar esperando un buen rato.

La buena noticia es que si está utilizando Caché de registro, sólo tendrá que sufrir este retraso una vez. Después, la información de registro se maneja localmente. El caché de registro se habilita desde la configuración de TortoiseSVN.

### 4.26.1. Nodos del gráfico de revisión

Cada nodo de gráfico de revisión representa una revisión en el repositorio donde algo cambió en el árbol que está viendo. Los diferentes tipos de nodos se pueden distinguir por forma y color. Las formas son fijas, pero los colores se pueden cambiar utilizando TortoiseSVN → Configuración

#### Ítems añadidos o copiados

Los objetos que se han añadido, o creado al copiar otro fichero/carpeta, se muestran empleando un rectángulo redondeado. El color por defecto es el verde. Etiquetas y trunks se tratan como un caso especial y utilizan un tono diferente, dependiendo de TortoiseSVN → Settings.

#### Ítems eliminados

Objetos eliminados, por ejemplo una rama que ya no es necesaria, se muestran usando un octágono (rectángulo con las esquinas recortadas). El color por defecto es el rojo.

#### Ítems renombrados

Los ítems renombrados también se muestran utilizando un octágono, pero el color por defecto es azul.

#### Revisión punta de la rama

El gráfico normalmente se restringe para mostrar los puntos de ramas, pero a menudo es útil poder ver las revisiones HEAD respectivas de cada rama. Si selecciona **Mostrar revisiones HEAD**, cada nodo de revisión HEAD se mostrará como una elipse. Tenga en cuenta que HEAD aquí hace referencia a la última revisión confirmada en esa ruta, no a la revisión HEAD del repositorio.

#### Revisión de la copia de trabajo

Si ha invocado el gráfico de revisiones desde una copia de trabajo, puede optar por mostrar la revisión BASE del gráfico utilizando **Mostrar revisión copia de trabajo**, que marca el nodo BASE con un borde remarcado.

#### Copia de trabajo modificada

Si ha invocado el gráfico de revisiones desde una copia de trabajo, puede optar por mostrar un nodo adicional representando su copia de trabajo modificada utilizando **Mostrar modificaciones en la copia de trabajo**. Éste por defecto es un nodo elíptico con un borde remarcado en rojo.

#### Ítem normal

Todos los demás ítems se muestran utilizando un rectángulo normal.

Tenga en cuenta que por defecto el gráfico sólo muestra los puntos en los que se añadieron, copiaron o borraron ítems. Mostrar cada revisión de un proyecto generará un gráfico muy grande para casos no triviales. Si realmente desea ver *todas* las revisiones en las que se hicieron cambios, hay una opción para conseguirlo en el menú **Ver** y en la barra de herramientas.

La vista por defecto (sin agrupación) pone los nodos con su posición vertical en estricto orden de revisión, para que tenga una pista visual del orden en el que se hicieron las cosas. Cuando dos nodos están en la misma columna, el orden es muy obvio. Cuando dos nodos están en columnas contiguas la separación es mucho menor porque no hay necesidad de evitar que los nodos se superpongan, y como resultado el orden es un poco menos obvio. Estas optimizaciones son necesarias para mantener gráficos complejos en un tamaño razonable. Por favor tenga en cuenta que esta ordenación utiliza el *borde* del nodo en el lado *más antiguo* como referencia, es decir, el borde inferior del nodo cuando el gráfico se muestra con el nodo más antiguo en la parte inferior. El borde de referencia es importante porque no todas las formas de los nodos tienen la misma altura.

### 4.26.2. Cambiando la vista

Dado que los gráficos de revisiones son a menudo complejos, hay un número de funcionalidades que pueden utilizarse para ajustar la vista a lo que desea obtener. Estas opciones están disponibles en el menú **Ver** y en la barra de herramientas.

#### Agrupar ramas

En el comportamiento por defecto (sin agrupaciones) todas las filas estarán ordenadas estrictamente por revisión. Como resultado, las ramas de larga vida con confirmaciones esporádicas ocupan una columna completa sólo para unos pocos cambios, y el gráfico resulta muy ancho.

This mode groups changes by branch, so that there is no global revision ordering: Consecutive revisions on a branch will be shown in (often) consecutive lines. Sub-branches, however, are arranged in such a way that later branches will be shown in the same column above earlier branches to keep the graph slim. As a result, a given row may contain changes from different revisions.

#### Más antiguo arriba

Normalmente el gráfico muestra la revisión más antigua abajo, y el árbol crece hacia arriba. Utilice esta opción para hacer crecer el árbol desde arriba.

#### Alinear árboles en la parte superior

Cuando un gráfico se parte en varios árboles más pequeños, los árboles pueden aparecer bien en su orden de revisión natural, o alineados en la parte inferior de la ventana, dependiendo de si utiliza la opción **Agrupar por ramas**. Utilice esta opción para hacer crecer todos los árboles hacia abajo desde arriba.

#### Reducir líneas cruzadas

This option is normally enabled and avoids showing the graph with a lot of confused crossing lines. However this may also make the layout columns appear in less logical places, for example in a diagonal line rather than a column, and the graph may require a larger area to draw. If this is a problem you can disable the option from the **View** menu.

#### Nombres de ruta diferenciales

Los nombres largos de ruta pueden necesitar un montón de espacio y hacer que las cajas de los nodos sean muy largas. Utilice esta opción para mostrar sólo la parte cambiada de una ruta, reemplazando la parte común con puntos. Por ejemplo, si crea la rama `/branches/1.2.x/doc/html` desde `/trunk/doc/html` la rama se podría mostrar de forma compacta como `/branches/1.2.x/`. dado que los dos últimos niveles, `doc` y `html`, no han cambiado.

#### Mostrar todas las revisiones

Esto hace exactamente lo que espera, y muestra cada revisión donde algo (en el árbol del que está mostrando el gráfico) ha cambiado. Para historiales largos esto puede producir un árbol realmente enorme.

#### Mostrar revisiones HEAD

Esto asegura que la última revisión de cada rama se muestra siempre en el gráfico.

#### Orígenes de copia exactos

Cuando se crea una rama/etiqueta, el comportamiento por defecto es mostrar la rama como si se hubiera tomado desde el último nodo donde se hizo un cambio. Estrictamente esto no es preciso, dado que las ramas generalmente se crean desde HEAD y no desde una revisión específica. Por lo que es posible mostrar la revisión más correcta (pero menos útil) que se utilizó para crear la copia. Note que esta revisión puede ser más reciente que la revisión HEAD de la rama origen.

#### Plegar etiquetas

When a project has many tags, showing every tag as a separate node on the graph takes a lot of space and obscures the more interesting development branch structure. At the same time you may need to be able to access the tag content easily so that you can compare revisions. This option hides the nodes for tags and shows them instead in the tooltip for the node that they were copied from. A tag icon on the right side of the source node indicates that tags were made. This greatly simplifies the view.

Note that if a tag is itself used as the source for a copy, perhaps a new branch based on a tag, then that tag will be shown as a separate node rather than folded.

#### Ocultar rutas eliminadas

Ocultar las rutas que ya no están presentes en la revisión HEAD del repositorio, por ejemplo ramas eliminadas.

If you have selected the **Fold tags** option then a deleted branch from which tags were taken will still be shown, otherwise the tags would disappear too. The last revision that was tagged will be shown in the colour used for deleted nodes instead of showing a separate deletion revision.

If you select the **Hide tags** option then these branches will disappear again as they are not needed to show the tags.

#### Ocultar ramas sin usar

Ocultar ramas donde no se confirmaron cambios en el respectivo archivo o sub-carpeta. Esto no indica necesariamente que la rama no se utilizó, sino que no se hicieron cambios en *esta* parte de ella.

#### Mostrar la revisión de la copia de trabajo

Marca la revisión en el gráfico que corresponde a la revisión de actualización del ítem para el que obtuvo el gráfico. Si acaba de actualizar, esto será HEAD, pero si otros han confirmado cambios desde su última actualización, su copia de trabajo puede estar unas revisiones por debajo. Este nodo se marca dándole un borde remarcado.

#### Mostrar modificaciones en la copia de trabajo

Si su copia de trabajo contiene cambios locales, esta opción lo dibuja como un nodo separado elíptico, enlazado al nodo al que su copia de trabajo fue actualizado por última vez. El color de resalte por defecto es rojo. Puede necesitar refrescar el gráfico utilizando **F5** para capturar cambios recientes.

#### Filtro

A veces los gráficos de revisión contienen más revisiones de las que desea ver. Esta opción abre un diálogo que le permitirá restringir el rango de revisiones mostrado, y ocultar rutas concretas por su nombre.

If you hide a particular path and that node has child nodes, the children will be shown as a separate tree. If you want to hide all children as well, use the **Remove the whole subtree(s)** checkbox.

#### Bandas de árboles

Cuando el gráfico contiene varios árboles, es útil a veces utilizar colores alternados en el fondo para ayudar a distinguir entre árboles.

#### Mostrar resumen

Muestra un pequeño dibujo del gráfico completo, con la ventana de la vista actual como un rectángulo que puede arrastrar. Esto le permite navegar por el gráfico más fácilmente. Note que para gráficos grandes la vista previa puede volverse inutilizable debido al factor de zoom extremo que se usa y por lo tanto no mostrarse en esos casos.

### 4.26.3. Usando el gráfico

Para hacer más fácil la navegación en un gráfico grande, utilice la ventana superpuesta. En ella se muestra el gráfico completo en una pequeña ventana, con la parte que se muestra actualmente iluminada. Puede arrastrar el área iluminada para cambiar la región que se muestra.

La fecha de la revisión, el autor y los comentarios se muestran en una caja de ayuda cuando se mueve el ratón encima de una caja de revisión.

Si selecciona dos revisiones (mediante **Ctrl**-click con el botón izquierdo), puede utilizar el menú contextual para mostrar las diferencias entre esas revisiones. Puede elegir mostrar las diferencias en los puntos de creación de las ramas, pero normalmente querrá mostrar las diferencias en los puntos finales de las ramas, esto es, en la revisión HEAD.

Puede ver las diferencias como un archivo de Diff Unificado, que le muestra todas las diferencias en un archivo único con contexto mínimo. Si opta por **Menú Contextual** → **Comparar Revisiones** se le presentará una lista de archivos cambiados. Haga doble click en un nombre de archivo para obtener ambas revisiones y compararlas utilizando una herramienta de diferencias visual.

Si hace click con el botón derecho en una revisión, puede utilizar **Menú Contextual** → **Mostrar registro** para ver la historia.

También puede fusionar cambios de la(s) revisión(es) seleccionada(s) en una copia de trabajo distinta. Un diálogo de selección de carpeta le permitirá elegir la copia de trabajo donde desea fusionar, pero después de eso no hay diálogo de confirmación, ni oportunidad de probar la fusión sin ejecutarla realmente. Es una buena idea fusionar en una copia de trabajo sin cambios, ¡y así poder revertir los cambios si no funcionan! Esta es una funcionalidad útil si desea fusionar las revisiones seleccionadas de una rama a otra.



## Aprenda a Leer un Gráfico de Revisiones

Los usuarios novatos pueden sorprenderse porque el gráfico de revisiones muestra algo que no concuerda con el modelo mental del usuario. Si una revisión cambia múltiples copias o ramas de un archivo o carpeta, por ejemplo, habrá nodos múltiples para esa revisión. Es una buena práctica empezar con las opciones de la barra de herramientas que están más a la izquierda y customizar el gráfico paso a paso hasta que casi coincida con su modelo mental.

Todas las opciones de filtro tratan de perder la mínima información posible. Esto puede hacer que algunos nodos cambien de color. Siempre que un resultado sea inesperado, deshaga la última operación de filtrado y trate de entender qué tiene de especial esa revisión o rama. En la mayoría de los casos, el resultado inicial esperado de la operación de filtrado sería inexacto o engañoso.

### 4.26.4. Refrescando la vista

Si desea comprobar el servidor de nuevo para obtener la información más reciente, puede simplemente refrescar la vista utilizando **F5**. Si está utilizando la caché de registro (habilitada por defecto), esto buscará en el repositorio las confirmaciones más recientes y traerá sólo las nuevas. Si la caché de registro estaba en modo desconectado, esto intentará volver a ponerla en línea.

Si está utilizando la caché de registros y cree que el contenido del mensaje o el autor han podido cambiar, debería utilizar el diálogo de registro para refrescar los mensajes que necesita. Dado que el gráfico de revisiones trabaja desde la raíz del repositorio, tendríamos que invalidar la caché completa de registro, y recargarla podría llevar *muchísimo* tiempo.

### 4.26.5. Podando árboles

Un árbol grande puede ser difícil para navegar y a veces querrá ocultar partes de él, o partirlo en un bosque de árboles más pequeños. Si mueve el ratón sobre el punto donde un enlace de nodo entra o sale del nodo, verá que aparecen uno o más botones que le permitirán hacer esto.



Pulse en el botón menos para colapsar el subárbol adjunto.



Pulse el botón más para expandir un árbol colapsado. Cuando un árbol se ha colapsado, este botón permanece visible para indicar un subárbol oculto.



Pulse en el botón cruz para separar el subárbol adjunto y mostrarlo como un árbol separado en el gráfico.



Pulse en el botón círculo para readjuntar un árbol separado. Cuando se ha separado un árbol, este botón permanece visible para indicar que hay un subárbol separado.

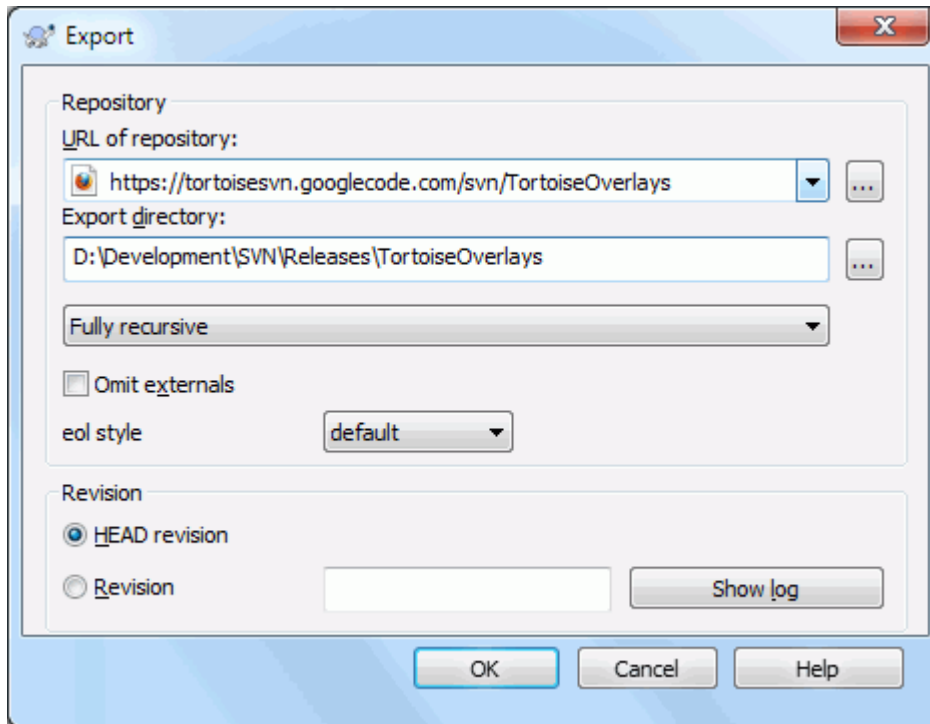
Pulse sobre el fondo del gráfico para acceder al menú contextual principal, el cual ofrece opciones para Expandir todo y Juntar todo. Si ninguna rama ha sido colapsada o partida, el menú contextual no será mostrado.

## 4.27. Exportando una copia de trabajo de 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 4.68. El diálogo Exportar-desde-URL**

Si ejecuta este comando sobre una carpeta sin versionar, TortoiseSVN asumirá que la carpeta de destino es el objetivo, y abrirá un diálogo para que introduzca la URL y la revisión desde la que desea realizar la exportación. Este diálogo tiene opciones para exportar sólo la carpeta de más alto nivel, para omitir las referencias externas, y para obligar un estilo de fin de línea concreto en aquellos archivos que tuvieran establecida la propiedad `svn:eol-style`.

Por supuesto también puede exportar directamente desde el repositorio. Utilice el Navegador de repositorios para navegar al subárbol relevante en su repositorio, y luego utilice Menú contextual → Exportar. Obtendrá el diálogo Exportar desde URL descrito más arriba.

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.

Cuando se exporta de una copia de trabajo, si la carpeta objetivo una carpeta con el mismo nombre que aquella que estás exportando, se te dará la opción de sobrescribir el contenido existente, o crear una nueva carpeta con un nombre generado de manera automática, por ejemplo `Objetivo (1)`.



### Exportando archivos sueltos

El diálogo de exportar no permite exportar archivos sueltos, aunque Subversion puede hacerlo.

Para exportar archivos sueltos con TortoiseSVN, tendrá que utilizar el navegador de repositorios (Sección 4.25, “El navegador de repositorios”). Simplemente arrastre el o los archivos que desea exportar desde el navegador de repositorios a donde los desee tener en el explorador, o utilice el menú contextual del navegador de repositorios para exportar los archivos.



## Exportando un árbol de cambios

Si quiere exportar una copia de su estructura de árbol del proyecto pero conteniendo sólo los archivos que han cambiado en una revisión en concreto, o entre dos revisiones, utilice la característica de comparar revisiones descrita en Sección 4.11.3, “Comparando carpetas”.

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. Eliminando una copia de trabajo del control de versiones

Sometimes you have a working copy which you want to convert back to a normal folder without the `.svn` directory. All you need to do is delete the `.svn` directory from the working copy root.

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

## 4.28. Relocalizando una copia de trabajo

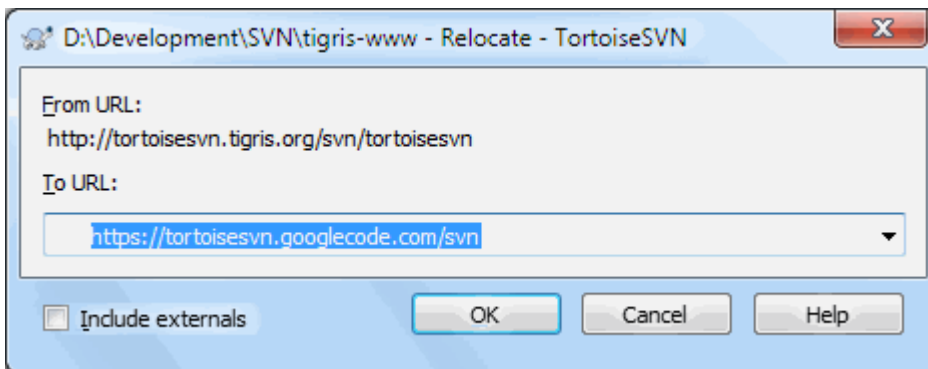


Figura 4.69. El diálogo Relocalizar

If your repository has for some reason changed it's location (IP/URL). Maybe you're even stuck and can't commit and you don't want to checkout your working copy again from the new location and to move all your changed data back into the new working copy, TortoiseSVN → Relocate is the command you are looking for. It basically does very little: it rewrites all URLs that are associated with each file and folder with the new URL.

### Nota

Esta operación sólo funciona para raíces de la copia de trabajo. Así que la entrada del menú contextual sólo se muestra para raíces de la copia de trabajo.

Puede sorprenderse al ver que TortoiseSVN contacta el repositorio como parte de esta operación. Todo lo que hace es realizar algunas comprobaciones sencillas para asegurarse que la nueva URL realmente se refiere al mismo repositorio que la copia de trabajo existente.



## Aviso

*Esta es una operación que se utiliza realmente poco.* El comando `relocalizar` se utiliza *sólo* si la URL de la raíz del repositorio ha cambiado. Estas son algunas posibles razones:

- La dirección IP del servidor ha cambiado.
- El protocolo ha cambiado (por ejemplo, de `http://` a `https://`).
- La ruta raíz del repositorio ha cambiado en la configuración del servidor.

De otra forma, necesita `relocalizar` cuando su copia de trabajo se refiere al mismo lugar en el mismo repositorio, pero es el propio repositorio el que se ha movido.

Esto no se aplica si:

- Quiere moverse a un repositorio de Subversion diferente. En ese caso debería realizar una obtención limpia desde la nueva localización del repositorio.
- Quiere cambiar a una rama o a un directorio diferente dentro del mismo repositorio. Para hacer eso debería utilizar TortoiseSVN → Cambiar.... Para más información, lea [Sección 4.20.3, “Obtener o cambiar...”](#).

Si utiliza `relocalizar` en alguno de los casos anteriores, *corromperá su copia de trabajo* y obtendrá muchos mensajes de error inexplicables cuando actualice, confirme, etc. Una vez que esto ha ocurrido, el único arreglo es hacer una obtención limpia.

## 4.29. Integración con sistemas de control de errores / seguimiento de incidencias

Es muy común en el Desarrollo de Software que los cambios se refieran a un ID de error o incidencia específico. A los usuarios de los sistemas de control de errores (seguimiento de incidencias) les gustaría asociar los cambios que hacen en Subversion con un ID específico en su programa de seguimiento de incidencias. La mayoría de programas de seguimiento de incidencias proporcionan un script gancho pre-commit que parsea el mensaje de log para encontrar el ID del error con el que se asocia la confirmación. Esto es de alguna forma propenso a errores, porque se basa en el que usuario escriba el mensaje de registro de forma correcta para que el script gancho pre-commit pueda parsearlo correctamente.

TortoiseSVN puede ayudar al usuario de dos formas:

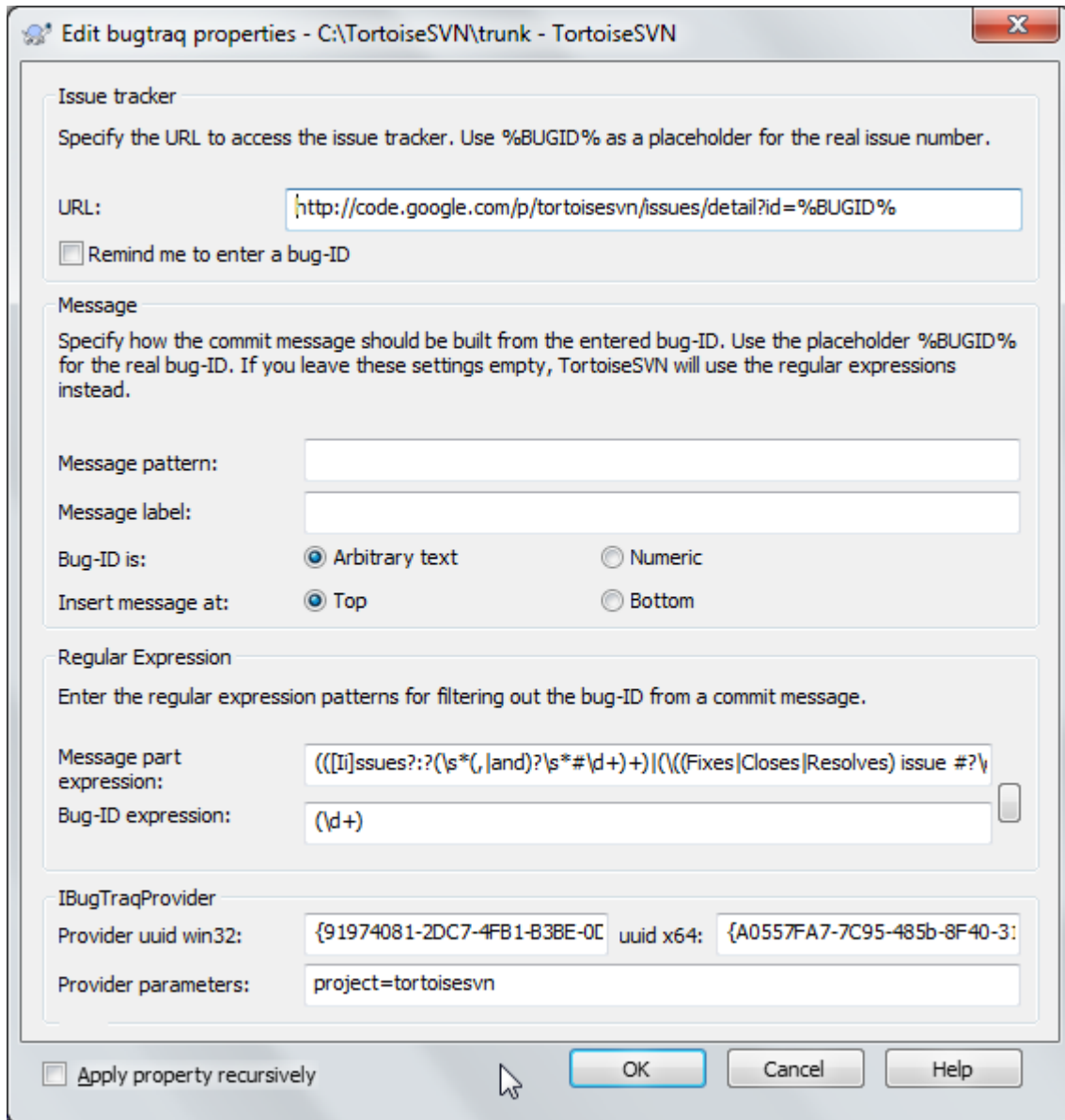
1. Cuando el usuario introduce un mensaje de registro, puede añadirse una línea bien definida incluyendo el número de incidencia asociada a la confirmación. Esto reduce el riesgo que el usuario introduzca el número de incidencia que las herramientas de control de errores no puedan parsear correctamente.

O TortoiseSVN puede remarcar la parte del mensaje de registro introducido que el programa de seguimiento de incidencias reconoce. De esta forma el usuario sabrá que el mensaje de registro introducido puede parsearse correctamente.

2. Cuando el usuario navega por los mensajes de registro, TortoiseSVN crea un enlace desde cada ID de error en el mensaje de registro, que lanza el navegador con la incidencia mencionada.

### 4.29.1. Añadiendo números de incidencia en los mensajes de registro

Puede integrar una herramienta de control de errores de su elección con TortoiseSVN. Para hacerlo, necesita definir algunas propiedades, que empiezan por `bugtraq:`. Deben establecerse sobre las carpetas: ([Sección 4.18, “Configuración del proyecto”](#))



**Figura 4.70. El diálogo de propiedades de Bugtraq**

When you edit any of the bugtraq properties a special property editor is used to make it easier to set appropriate values.

Hay dos formas de integrar TortoiseSVN con los programas de control de incidencias. Una se basa en simples cadenas de texto, la otra se basa en *expresiones regulares*. Las propiedades que se utilizan en ambos casos son:

bugtraq:url

Establezca esta propiedad a la URL de su herramienta de seguimiento de errores. Debe ser una URI debidamente codificada y debe contener %BUGID%. %BUGID% se sustituye con el número de incidencia que haya introducido. Esto permite que TortoiseSVN muestre un vínculo en el diálogo de registro, de modo que cuando usted está buscando en el registro de revisión pueda saltar directamente a la herramienta de seguimiento de errores. Usted no tiene que proporcionar esta propiedad, pero entonces TortoiseSVN sólo mostrará el número de incidencia y no el vínculo a la misma. Por ejemplo, el proyecto TortoiseSVN utiliza `http://issues.tortoisesvn.net/?do=details&id=%BUGID%`.

También puede utilizar URLs relativas en vez de absolutas. Esto es útil en caso que su control de incidencias esté en el mismo dominio/servidor que su repositorio. En caso que el nombre de dominio cambie alguna vez, no tendrá que ajustar la propiedad `bugtraq:url`. Hay dos formas de especificar una URL relativa:

Si empieza con la cadena `^/` se asume que es relativa a la raíz del repositorio. Por ejemplo, `^/../?do=details&id=%BUGID%` se resolverá como `http://tortoisesvn.net/?do=details&id=%BUGID%` si su repositorio está en `http://tortoisesvn.net/svn/trunk/`.

Una URL que comience con la cadena `/` se asume que es relativo al nombre del servidor. Por ejemplo, `/?do=details&id=%BUGID%` se resolverá como `http://tortoisesvn.net/?do=details&id=%BUGID%` si su repositorio está localizado en cualquier parte de `http://tortoisesvn.net`.

`bugtraq:warnifnoissue`

Establezca esta propiedad a `true` si desea que TortoiseSVN le avise cuando deje en blanco el campo de texto del número de incidencia. Los valores válidos son `true/false`. *Si no se define, se supone false.*

#### 4.29.1.1. Número de incidencia en caja de texto

En el caso simple, TortoiseSVN muestra al usuario un campo de entrada separado donde se puede introducir un ID del error. En ese caso se añade una línea adicional al final o al principio del mensaje de registro que el usuario introduce.

`bugtraq:message`

Esta propiedad activa el sistema de control de errores en el modo *campo de entrada*. Si se establece esta propiedad, TortoiseSVN le preguntará por el número de incidencia cuando confirme sus cambios. Se utiliza para añadir una línea al final del mensaje de registro. Debe contener `%BUGID%`, que se reemplaza por el número de la incidencia al confirmar. Esto asegura que su registro de confirmación contiene una referencia al número de incidencia que siempre está en un formato consistente y que puede ser parseado por su herramienta de control de errores para asociar el número de incidencia con una confirmación en concreto. Un ejemplo que puede que le sirva es `Incidencia : %BUGID%`, pero esto depende de su herramienta.

`bugtraq:label`

Éste es el texto que TortoiseSVN muestra en el diálogo de confirmar para etiquetar el cuadro de texto en el que introduce el número de incidencia. Si no se establece, aparecerá `Bug-ID / N°-Incid. : .` Tenga en cuenta, sin embargo, que la ventana no se cambiará de tamaño para acomodar esta etiqueta, por lo que debería mantener el tamaño de la etiqueta en menos de 20-25 caracteres.

`bugtraq:number`

Si se establece a `true` sólo se permitirán números en el campo de número de incidencia. La coma es una excepción, para que pueda introducir varios números separados por comas. Los valores válidos son `true/false`. *Si no se define, se asumirá true.*

`bugtraq:append`

Esta propiedad define si el ID del error se añade (`true`) al final del mensaje de registro o se inserta (`false`) al inicio del mensaje de registro. Los valores válidos son `true/false`. *Si no se define, se asume true, para que los proyectos ya existentes no se rompan.*

#### 4.29.1.2. Números de incidencia utilizando expresiones regulares

En el caso de utilizar *expresiones regulares*, TortoiseSVN no muestra un campo de entrada separado sino que marca, del mensaje de registro que el usuario introduce, la parte que reconoce el programa de seguimiento de incidencias. Esto se realiza mientras el usuario introduce el mensaje de registro. ¡Esto también significa que el ID del error puede introducirse en cualquier parte dentro de un mensaje de registro! Este método es mucho más flexible, y es el que se utiliza en el propio proyecto TortoiseSVN.

`bugtraq:logregex`

Esta propiedad activa el sistema de control de errores en modo *Regex*. Contiene una expresión regular, o dos (en líneas separadas).

Si se dan dos expresiones, la primera expresión se utiliza como un pre-filtro para encontrar expresiones que contengan identificadores de incidencias. La segunda expresión se utiliza para extraer los identificadores individuales desde los resultados de la primera expresión regular. Esto le permite utilizar una lista de identificadores de incidencias y expresiones en lenguaje natural, si así lo desea. Por ejemplo puede arreglar varias incidencias e incluir una cadena como esta: "Este cambio resuelve las incidencias #23, #24 y #25".

Si desea capturar identificadores de incidencias utilizados como en la expresión anterior dentro de un mensaje de registro, puede utilizar las siguientes cadenas de expresiones regulares, que son los utilizados en el proyecto TortoiseSVN: `[Ii]ssues?:?(\s*(,|and)?\s*\#\d+)+ y (\d+)`

La primera expresión extrae “incidencias #23, #24 y #25” de entre el resto del mensaje de registro. La segunda expresión regular extrae los números decimales individuales desde la salida de la primera expresión regular, por lo que devolverá “23”, “24” y “25” para utilizarlos como identificadores de incidencias.

Profundizando en la primera expresión regular un poco, debe empezar con la palabra “incidencia”, posiblemente empezando por mayúscula. Opcionalmente puede ir seguida de una “s” (más de una incidencia) y opcionalmente dos puntos. A esto le sigue uno o más grupos, cada uno empezando por cero o más espacios en blanco, una coma opcional o “y” y más espacios opcionales. Finalmente, hay un “#” obligatorio y un número decimal obligatorio.

Si sólo se establece una expresión, entonces los identificadores individuales de incidencias deben concordar con los grupos de la expresión regular. Por ejemplo: `[Ii]ncidencia(?:s)? \#?(\d+)` Este método se necesita por algunos gestores de incidencias, por ejemplo trac, pero la expresión regular es más difícil de construir. Le recomendamos que sólo utilice este método si la documentación de su gestor de incidencias se lo indica.

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

It's not always easy to get the regex right so to help out there is a test dialog built into the bugtraq properties dialog. Click on the button to the right of the edit boxes to bring it up. Here you can enter some test text, and change each regex to see the results. If the regex is invalid the edit box background changes to red.

Si se establecen las dos propiedades `bugtraq:message` y `bugtraq:logregex`, `logregex` tiene preferencia.



### Sugerencia

¡Incluso si no tiene un programa de seguimiento de incidencias con un gancho pre-commit parseando sus mensajes de log, puede seguir utilizando ésto para convertir las incidencias mencionadas en sus mensajes de registro en vínculos!

E incluso si no necesita los vínculos, los números de incidencias se muestran como una columna separada en el diálogo de registro, facilitándole la búsqueda de los cambios que se refieran a una incidencia en concreto.

Algunas propiedades `tsvn:` necesitan un valor `true/false` (verdadero/falso). TortoiseSVN también entiende `yes` (sí) como un sinónimo de `true` y `no` como un sinónimo de `false`.



### Estableciendo las propiedades en las carpetas

Esas propiedades deben activarse en carpetas del sistema a trabajar. Cuando confirmas un archivo o carpeta, las propiedades son leídas de esa carpeta. Si no se encuentran las propiedades ahí, TortoiseSVN buscara en orden ascendente en el árbol de carpetas, para encontrarlas, hasta que encuentre una carpeta fuera del control de versiones, o la raíz del árbol (por ejemplo: `C:\`) es encontrada. Si estas seguro que cada usuario hace check out de y no de alguna sub-carpeta, entonces sera suficiente si se ponen las propiedades en `.` Si no estas seguro, entonces se deberán poner las propiedades recursivamente en cada sub-carpeta. Una propiedad mas profunda en la jerarquía del proyecto, tendrá preferencia sobre niveles mas altos (mas cercanos a `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.

*Excusivamente para las propiedades de proyecto* , por ejemplo: `tsvn:`, `bugtraq:` and `webviewer:`, puede utilizar la casilla **Recursivo** para establecer la propiedad en todas las subcarpetas de la jerarquía, sin establecerla en todos los archivos.

Cuando añade nuevas subcarpetas a una copia de trabajo utilizando TortoiseSVN, cualquier propiedad de proyecto presente en la carpeta padre se añadirá también automáticamente a la nueva carpeta hija.



## Sin información de seguimiento de incidencias desde el navegador de repositorios

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.

Por la misma razón, las propiedades del proyecto no se propagarán automáticamente cuando un se añade una carpeta hija usando el navegador de repositorios.

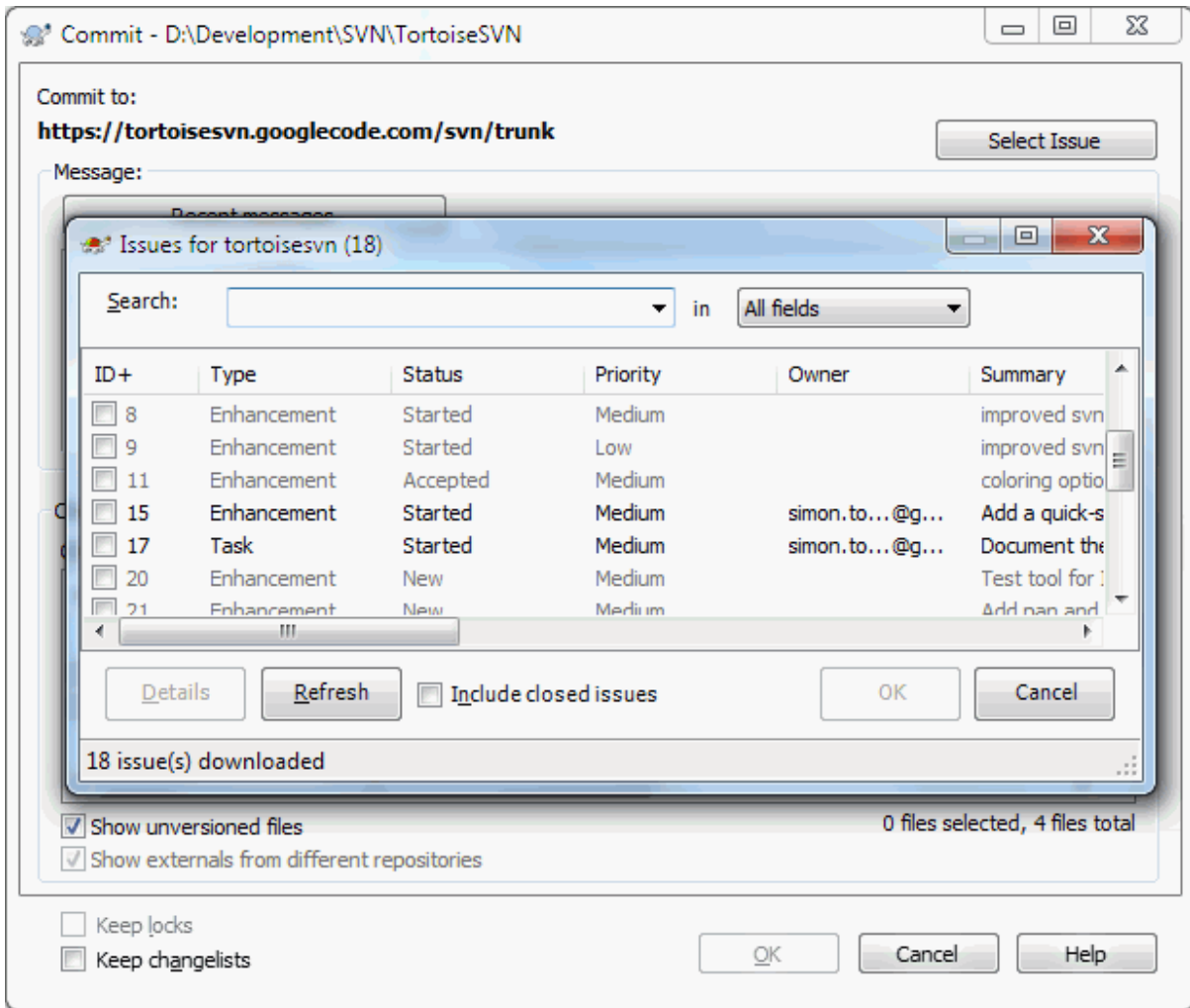
This issue tracker integration is not restricted to TortoiseSVN; it can be used with any Subversion client. For more information, read the full *Issue Tracker Integration Specification* [<https://svn.code.sf.net/p/tortoisesvn/code/trunk/doc/notes/issuetrackers.txt>] in the TortoiseSVN source repository.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

### 4.29.2. Obteniendo información desde el gestor de incidencias

La sección anterior se refiere a cómo añadir información de incidencias en los mensajes de registros. Pero, ¿y si lo que necesita es obtener información desde el gestor de incidencias? El diálogo de confirmación tiene un interfaz COM que le permite la integración de un programa externo que pueda hablar con su gestor de incidencias. Típicamente, querrá interrogar al gestor para obtener una lista de incidencias abiertas que tenga asignadas, para que pueda seleccionar las incidencias de las que se va a hacer cargo en esta confirmación.

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>]. (Sección 3, “Licencia” explains how to access the repository.) A summary of the API is also given in *Capítulo 7, Interfaz 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.

Por poner un ejemplo ilustrativo, supongamos que el administrador de su sistema le ha proporcionado un plugin de gestor de incidencias que ya ha instalado, y que ha preparado algunas de sus copias de trabajo para que utilicen el plugin en el diálogo de configuración de TortoiseSVN. Cuando abra la ventana de confirmación desde una copia de trabajo que tenga el plugin asignado, verá un nuevo botón en la parte superior del diálogo.



**Figura 4.71. Diálogo de ejemplo de la interacción con el gestor de incidencias**

En este ejemplo, puede seleccionar una o más incidencias abiertas. El plugin podrá entonces generar texto específicamente formateado para añadirlo a su mensaje de registro.

### 4.30. Integración con visores de repositorios basados en web

Hay varios visores de repositorios basados en web disponibles para utilizarlos con Subversion como [ViewVC](http://www.viewvc.org/) [http://www.viewvc.org/] y [WebSVN](http://websvn.tigris.org/) [http://websvn.tigris.org/]. TortoiseSVN dispone de medios para enlazarse con estos visores.

Puede integrar un visor de repositorios de su elección en TortoiseSVN. Para hacerlo, debe definir algunas propiedades que definen la unión. Deben establecerse sobre las carpetas: (Sección 4.18, “Configuración del proyecto”)

`webviewer:revision`

Establezca esta propiedad a la URL del visor de tu repositorio para ver todos los cambios en una revisión específica. Debe estar adecuadamente codificada en URI y tiene que contener `%REVISION%`. `%REVISION%` es sustituido por el número de revisión en cuestión. Esto permite a TortoiseSVN mostrar un entrada en el menú de contexto en el registro de diálogo Menú de Contexto → Vea la revisión en el visor web.

`webviewer:pathrevision`

Establece esta propiedad en la URL del visualizador del repo para ver los cambios a un archivo específico en un revisión específica. Debe estar URI encodeado y debe contener `%REVISION%` y `%PATH%`. `%PATH%` es



reemplazado con la ruta relativa a la raíz del repositorio. Esto le permite a TortoiseSVN mostrar una entrada en el menú contextual en el diálogo del logContext Menu → View revision for path in webviewer Por ejemplo, si haces click con el botón derecho en la parte inferior del diálogo del log en un archivo `/trunk/src/file` entonces el `%PATH%` en la URL será reemplazado con `/trunk/src/file`.

También puede utilizar URLs relativas en vez de absolutas. Esto es útil en caso que su control de incidencias esté en el mismo dominio/servidor que su repositorio. En caso que el nombre de dominio cambie alguna vez, no tendrá que ajustar las propiedades `webviewer:revision` y `webviewer:pathrevision`. El formato es el mismo que el de la propiedad `bugtraq:url`. Vea [Sección 4.29, "Integración con sistemas de control de errores / seguimiento de incidencias"](#).



### Estableciendo las propiedades en las carpetas

Esas propiedades deben activarse en carpetas del sistema a trabajar. Cuando confirmas un archivo o carpeta, las propiedades son leídas de esa carpeta. Si no se encuentran las propiedades ahí, TortoiseSVN buscara en orden ascendente en el arbol de carpetas, para encontrarlas, hasta que encuentre una carpeta fuera del control de versiones, o la raíz del árbol (por ejemplo: `C:\`) es encontrada. Si estas seguro que cada usuario hace check out de y no de alguna sub-carpeta, entonces sera suficiente si se ponen las propiedades en `.` Si no estas seguro, entonces se deberán poner las propiedades recursivamente en cada sub-carpeta. Una propiedad mas profunda en la jerarquía del proyecto, tendrá preferencia sobre niveles mas altos (mas cercanos a `trunk/`).

*Excusivamente para las propiedades de proyecto* , por ejemplo: `tsvn:`, `bugtraq:` and `webviewer:`, puede utilizar la casilla **Recursivo** para establecer la propiedad en todas las subcarpetas de la jerarquía, sin establecerla en todos los archivos.

Cuando añade nuevas subcarpetas a una copia de trabajo utilizando TortoiseSVN, cualquier propiedad de proyecto presente en la carpeta padre se añadirá también automáticamente a la nueva carpeta hija.



### Limitaciones al Usar el Navegador de Repositorios

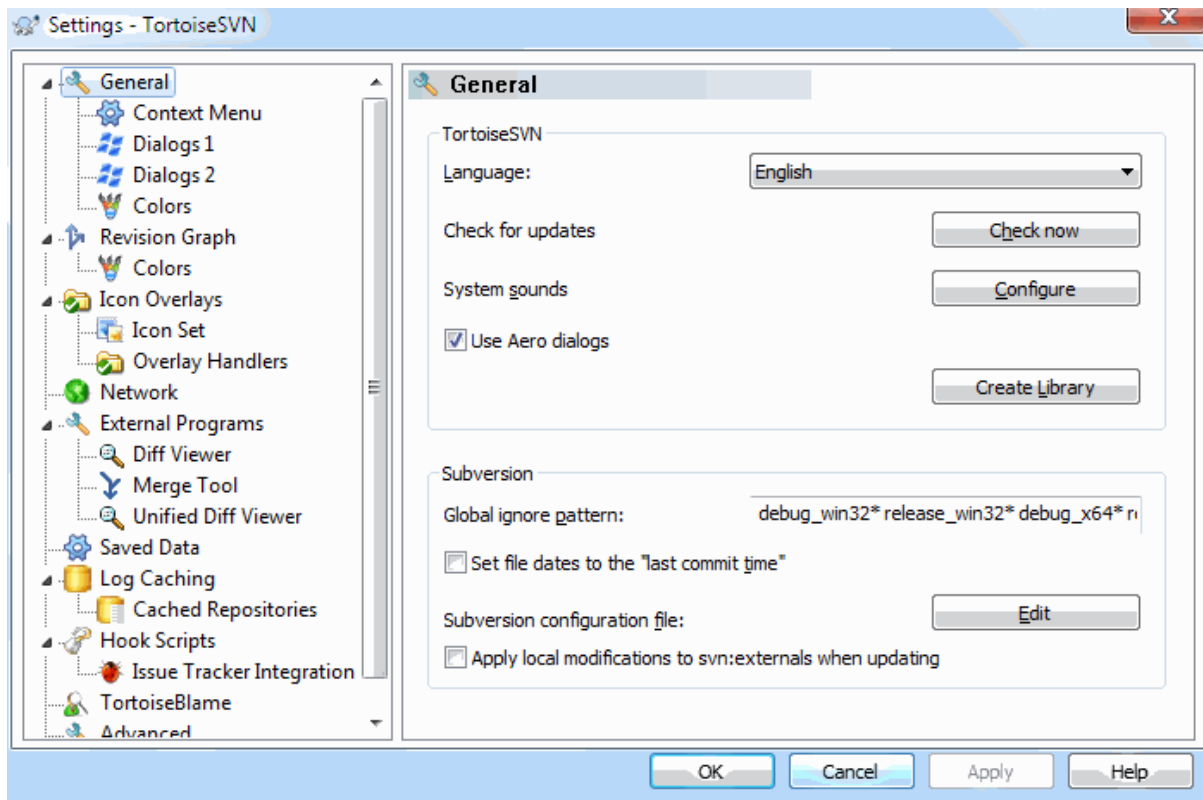
Because the repo viewer integration depends upon accessing Subversion properties, you will only see the results when using a checked out working copy. Fetching properties remotely is a slow operation, so you will not see this feature in action from the repo browser unless you started the repo browser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

Por la misma razón, las propiedades del proyecto no se propagarán automáticamente cuando un se añade una carpeta hija usando el navegador de repositorios.

## 4.31. Configuración de TortoiseSVN

Para averiguar para qué sirven cada una de las diferentes opciones, deje el puntero del ratón un segundo sobre el cuadro de texto/casilla... y aparecerá un útil texto de ayuda.

### 4.31.1. Configuración general



**Figura 4.72. El diálogo Configuración, página General**

Este diálogo le permite especificar su idioma preferido, y las configuraciones específicas de Subversion.

#### Idioma

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.

#### Buscar actualizaciones

TortoiseSVN contactará su sitio de descarga periódicamente para ver si hay disponible una versión más reciente del programa. Si existiera alguna mostrará un vínculo de notificación en el diálogo de actualización. Utilice el botón **Comprobar ahora** si desea una respuesta inmediata. No se descargará la versión más reciente; simplemente recibirá un diálogo de información diciéndole que hay una versión nueva disponible.

#### Sonidos del sistema

TortoiseSVN tiene tres sonidos personalizados que se instalan por defecto.

- Error
- Información
- Atención

Puede seleccionar sonidos diferentes (o quitarlos completamente) utilizando el Panel de Control de Windows. Configurar es un acceso directo al Panel de Control.

#### Utilizar los diálogos de Aero

En Windows Vista y posteriores esto controla si los cuadros de diálogo utilizan el estilo Aero.

#### Crear librería

On Windows 7 you can create a Library in which to group working copies which are scattered in various places on your system.

#### Patrón global de ignorar

Los patrones globales de ignorar se utilizan para evitar que aparezcan archivos no versionados, por ejemplo, en el diálogo de confirmación. Los archivos que concuerden con los patrones también se ignoran en las importaciones. Ignore archivos o directorios escribiendo sus nombres o extensiones. Los patrones se separan por espacios, por ejemplo, `bin obj *.bak *.~?? *.jar *. [Tt]mp`. Estos patrones no deberían incluir ningún separador de rutas. Tenga en cuenta que no hay forma de diferenciar entre archivos y directorios. Lea [Sección 4.14.1, “Concordancia de patrones en las listas de ignorados”](#) para obtener más información sobre la sintaxis de la concordancia de patrones.

Tenga en cuenta que los patrones de ignorar que especifique aquí también afectarán a otros clientes de Subversion que se ejecuten en su PC, incluyendo el cliente de línea de comandos.



## Atención

Si utiliza el archivo de configuración de Subversion para establecer un patrón `global-ignores`, prevalecerá sobre las configuraciones que haga aquí. El archivo de configuración de Subversion se accede utilizando **Editar** como se describe a continuación.

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 [Sección 4.14, “Ignorando archivos y directorios”](#) for more information.

Establecer las fechas de los archivos a la “hora de la última confirmación”

Esta opción le indica a TortoiseSVN que establezca las fechas de los archivos al momento de la última confirmación cuando se hace una obtención o una actualización. Si no, TortoiseSVN utiliza la fecha actual. Si está desarrollando software generalmente es mejor que utilice la fecha actual porque los sistemas de compilación generalmente se fijan en las fechas de los archivos para determinar qué archivos necesitan ser compilados. Si utiliza “hora de la última confirmación” y revierte a una revisión anterior del archivo, su proyecto puede que no se compile como lo espera.

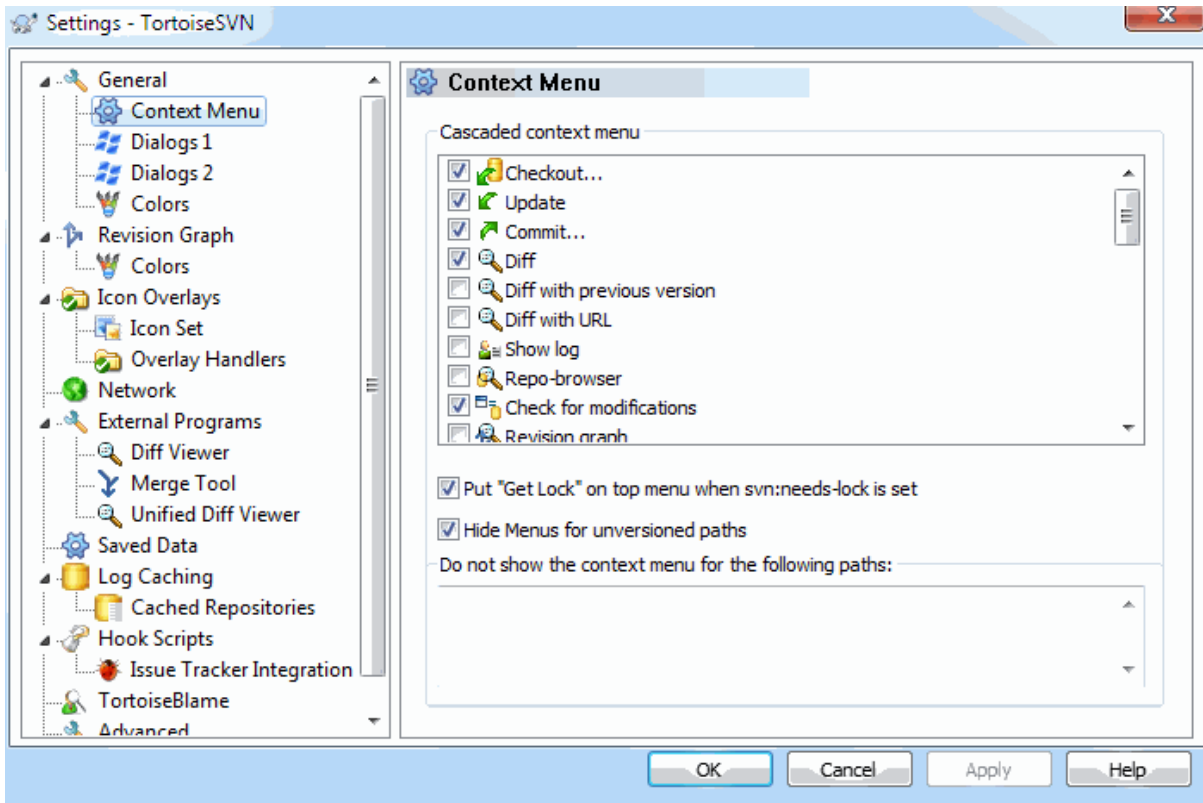
Archivo de configuración de Subversion

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.

Aplicar modificaciones locales a `svn:externals` al actualizar

This option tells TortoiseSVN to always apply local modifications to the `svn:externals` property when updating the working copy.

#### 4.31.1.1. Configuración del menú contextual



**Figura 4.73. El diálogo Configuración, página de Menú contextual**

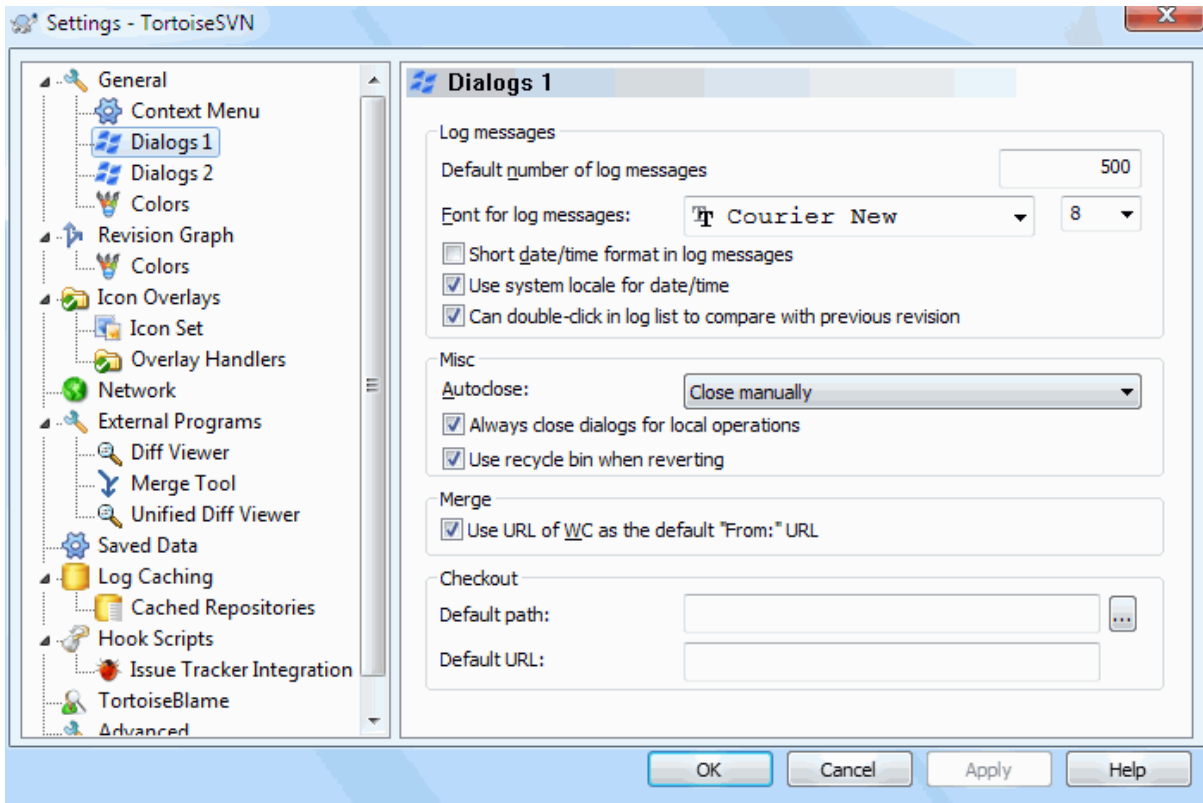
Esta página le permite especificar qué entradas del menú contextual de TortoiseSVN se mostrarán en el menú contextual principal, y cuales aparecerán en el submenú TortoiseSVN. Por defecto, la mayoría de los ítems están desmarcados y aparecen en el submenú.

Hay un caso especial para **Obtener bloqueo**. Por supuesto puede promocionarlo al nivel superior utilizando la lista anterior, pero como la mayoría de archivos no necesitan bloqueos esto molestaría. Sin embargo, un archivo con la propiedad `svn:needs-lock` necesita esta acción cada vez que se modifica, por lo que en ese caso sería útil tener la opción en el nivel superior. Activando esta casilla hace que cuando se selecciona un archivo que tenga establecida la propiedad `svn:needs-lock`, **Obtener bloqueo** se mostrará siempre en el nivel superior.

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.

Si hay algunas rutas en su ordenador en las que simplemente no desea que aparezca el menú contextual de TortoiseSVN en absoluto, puede listarlas en la caja inferior.

### 4.31.1.2. Configuración de diálogos de TortoiseSVN 1



**Figura 4.74. El diálogo Configuración, página de Diálogos 1**

Este diálogo le permite configurar a su gusto algunos de los diálogos de TortoiseSVN.

#### Número por defecto de mensajes de registro

Limita el número de mensajes de registro que TortoiseSVN obtiene la primera vez que seleccione TortoiseSVN → Mostrar registro Útil para conexiones lentas a servidores. Siempre puede utilizar Mostrar todos o Sigüientes 100 para obtener más mensajes.

#### Fuente para los mensajes de registro

Selecciona la fuente y el tamaño de la letra que se utiliza para mostrar el propio mensaje de registro en el panel del medio del diálogo Registro de Revisiones, y cuando se componen los mensajes de registro en el diálogo Confirmar.

#### Formato abreviado de fecha/hora en los mensajes de registro

Si los mensajes estándar largos toman mucho espacio en la pantalla utilice la forma corta.

#### Hacer doble clic en la lista de registro para comparar con la versión anterior

Si con frecuencia te encuentras comparando revisiones en el panel superior del diálogo del log, puedes usar esta opción para permitir dicha acción con un doble clic. No se permite por defecto porque adquirir el diff es con frecuencia un proceso prolongado, y muchos prefieren evitar la espera tras un doble clic, por este motivo esta opción no está activada por defecto.

#### Cerrar automáticamente

TortoiseSVN puede cerrar automáticamente todos los diálogos de progreso cuando la acción se termina sin error. Esta configuración le permite seleccionar las condiciones para el cierre de diálogos. La configuración por defecto (recomendada) es Cerrar manualmente que le permite revisar todos los mensajes y comprobar qué ha pasado. Sin embargo, puede decidir que desea ignorar algunos tipos de mensajes y hacer que el cuadro de diálogo se cierre automáticamente si no hay cambios críticos.

Auto-cerrar si no hay fusiones, adiciones o eliminaciones significa que el diálogo de progreso se cerrará si ha habido actualizaciones simples, pero si algún cambio del repositorio se ha fusionado con los suyos, o si se añadió o borró algún archivo, el diálogo seguirá abierto. También seguirá abierto si hubo algún conflicto o error durante la operación.

Auto-cerrar si no hay conflictos relaja el criterio y cerrará el diálogo incluso si ha habido fusiones, adiciones o borrados. Sin embargo, si hay algún conflicto o error, el diálogo continuará abierto.

Auto-cerrar si no hay errores siempre cierra el diálogo, incluso si hay conflictos. La única condición que deja el diálogo abierto es una condición de error, que ocurre cuando Subversion no puede terminar una tarea. Por ejemplo, una actualización falla si el servidor es inaccesible, o una confirmación falla cuando la copia de trabajo no está actualizada.

#### Siempre cierre los diálogos para las operaciones locales

Operaciones locales como agregar archivos o revertir cambios no tienen necesidad de contactarse con el repositorio y se completan rápidamente, por tanto el diálogo de progreso es en general de poco interés. Seleccione esta opción si quiere que el diálogo de progreso se cierre en forma automática luego de esas operaciones, a no ser que haya errores.

#### Utilizar la papelera de reciclaje al revertir

Cuando revierte las modificaciones locales, sus cambios se descartan. TortoiseSVN le ofrece una red de seguridad adicional al mandar el archivo modificado a la papelera de reciclaje antes de volver a la copia prístina. Si prefiere saltarse la papelera de reciclaje, desmarque esta opción.

#### Utilizar la URL de la copia de trabajo como la URL “Desde:” por defecto

En el diálogo de fusión, el comportamiento por defecto es que la URL Desde: se mantenga entre fusiones. Sin embargo, algunas personas quieren realizar fusiones desde diferentes puntos en su jerarquía, y les resulta más cómodo empezar con la URL de la copia de trabajo actual. Esto puede luego cambiarse para referirse a una ruta paralela en otra rama.

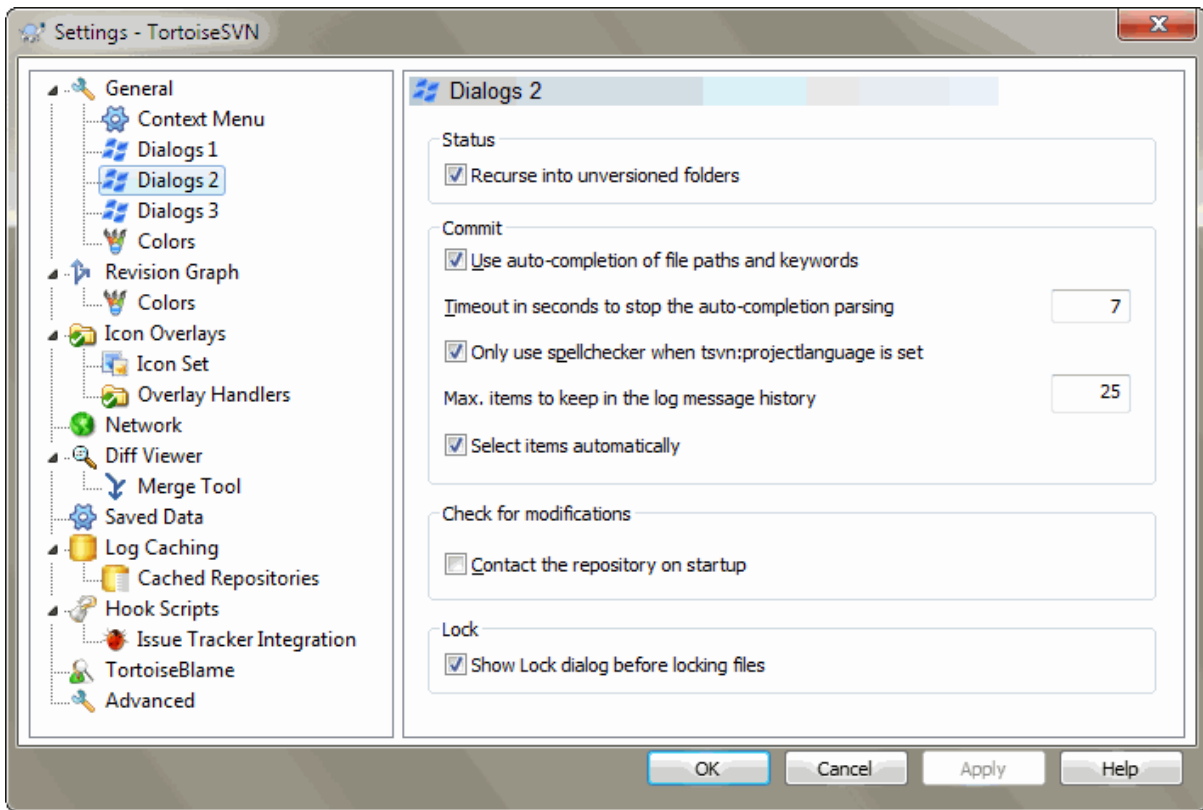
#### Ruta de obtención por defecto

Puede especificar la ruta por defecto para las obtenciones. Si mantiene todas sus obtenciones en un mismo lugar, es útil tener la unidad y la carpeta pre-rellenados para que así sólo tenga que añadir el nombre de la nueva carpeta al final.

#### URL de obtención por defecto

También puede especificar la URL por defecto para las obtenciones. Si obtiene a menudo subproyectos de un proyecto muy grande, puede ser útil tener la URL pre-rellenada de forma que sólo tenga que añadir el nombre del subproyecto al final.

### 4.31.1.3. Configuración de diálogos de TortoiseSVN 2



**Figura 4.75. El diálogo Configuración, página de Diálogos 2**

#### Recursividad en carpetas no versionadas

Si se marca esta casilla (por defecto está marcada), entonces siempre que se muestre el estado de una carpeta sin versionar en los diálogos de Añadir, Confirmar o Comprobar Modificaciones, se muestran también sus archivos y carpetas hijos. Si desmarca esta casilla, sólo se muestra la carpeta padre sin versionar, lo que reduce el desorden en esos diálogos. En ese caso, si selecciona una carpeta sin versionar para Añadir, se añadirá recursivamente.

En el diálogo Comprobar modificaciones puede elegir ver los elementos ignorados. Si esta casilla está marcada, siempre que se encuentre una carpeta ignorada se mostrarán todos los elementos hijos.

#### Utilizar autocompletar en rutas de archivos y palabras clave

El diálogo confirmar incluye una ayuda para parsear la lista de nombres de archivos que se van a confirmar. Cuando teclee las primeras 3 letras de un ítem en la lista, aparecerá la caja de autocompletar, y puede pulsar Enter para completar el nombre del archivo. Seleccione la caja para habilitar esta opción.

#### Tiempo límite en segundos para detener el parseado de autocompletar

El parseador de autocompletar puede ser bastante lento si hay muchos archivos grandes que comprobar. Este tiempo límite evita que el diálogo de confirmación se congele durante mucho tiempo. Si se está perdiendo información de autocompletar importante, puede extender el tiempo límite.

#### Utilizar el corrector ortográfico únicamente cuando se haya establecido `tsvn:projectlanguage`

Si no desea utilizar el corrector ortográfico para todas las confirmaciones, marque esta casilla. Aun así, el corrector ortográfico se activará cuando las propiedades del proyecto así lo indiquen.

#### Número máximo de ítems para mantener en el historial de mensajes de registro.

Cuando escribe un mensaje de registro en el diálogo de confirmación, TortoiseSVN lo almacena para poder reutilizarlo más tarde. Por defecto, almacena los últimos 25 mensajes de registro para cada repositorio, pero puede personalizar el número aquí. Si tiene muchos repositorios diferentes, puede querer reducirlo para evitar llenar su registro.

Tenga en cuenta que esta configuración sólo se aplica a los mensajes que escriba en este ordenador. No tiene nada que ver con la caché de registro.

#### Seleccionar ítems automáticamente

El comportamiento normal en el diálogo de confirmación es que todos los ítems (versionados) modificados se seleccionen automáticamente para la confirmación. Si prefiere empezar sin nada seleccionado y seleccionar los archivos para la confirmación manualmente, desmarque esta casilla.

#### Reabra el cuadro de diálogo después de enviar si los elementos fuesen no enviados.

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 con el repositorio al iniciar

El diálogo Comprobar Modificaciones comprueba la copia de trabajo por defecto, y sólo contacta con el repositorio cuando pulse el botón **Comprobar repositorio**. Si siempre quiere comprobar el repositorios, puede utilizar esta opción para que esa acción siempre ocurra automáticamente.

#### Mostrar el diálogo Bloquear antes de bloquear archivos

Cuando selecciona uno o más archivos y utiliza TortoiseSVN → **Bloquear** para obtener un bloqueo en esos archivos, en algunos proyectos es costumbre escribir un mensaje de bloqueo explicando por qué ha bloqueado esos archivos. Si no utiliza mensajes de bloqueo, puede desmarcar esta casilla para saltar este diálogo y bloquear los archivos inmediatamente.

Si utiliza el comando bloquear en una carpeta, siempre se le mostrará el diálogo de bloqueo ya que también le ofrece la posibilidad de seleccionar qué archivos bloquear.

Si su proyecto utiliza la propiedad `tsvn:lockmsgminsize`, verá el diálogo de bloqueo sin tener en cuenta esta opción ya que el proyecto *necesita* mensajes de bloqueo.

### 4.31.1.4. TortoiseSVN Dialog Settings 3

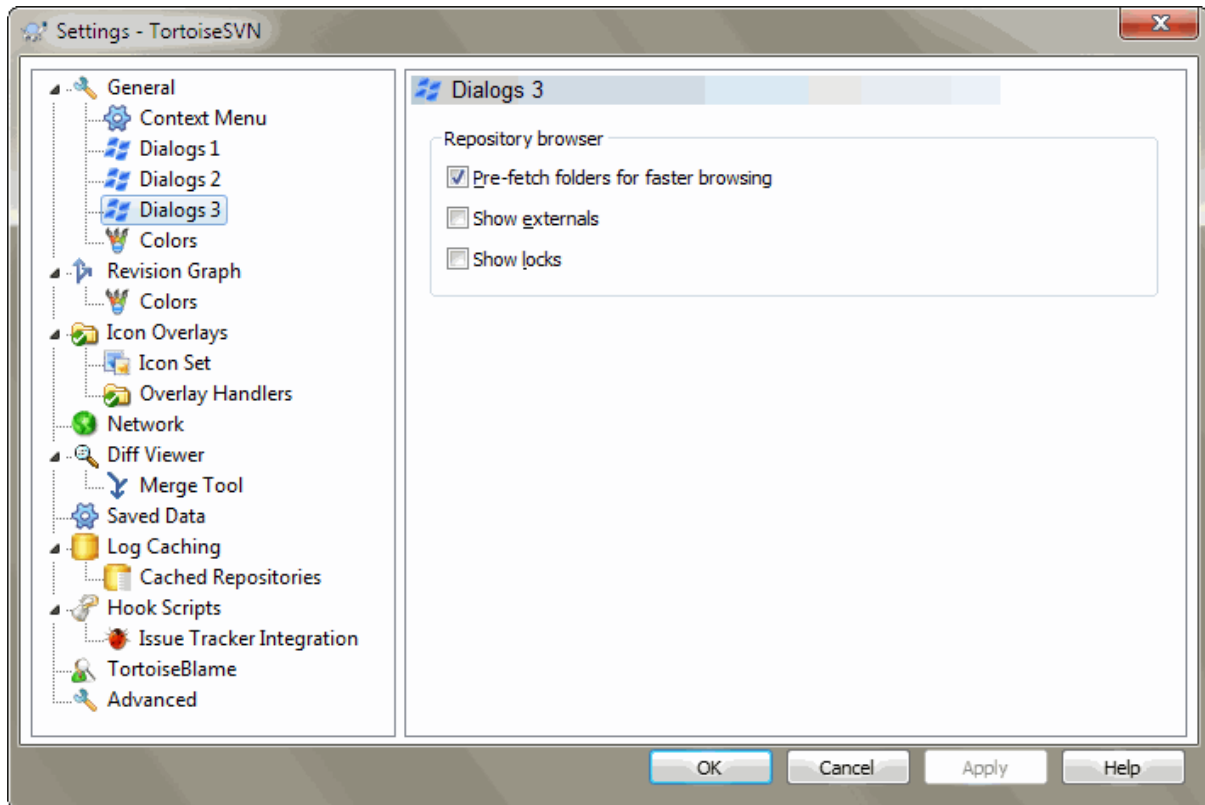


Figura 4.76. The Settings Dialog, Dialogs 3 Page



#### Cargar las carpetas con antelación para acelerar la navegación

If this box is checked (default state), then the repository browser fetches information about shown folders in the background. That way as soon as you browse into one of those folders, the information is already available.

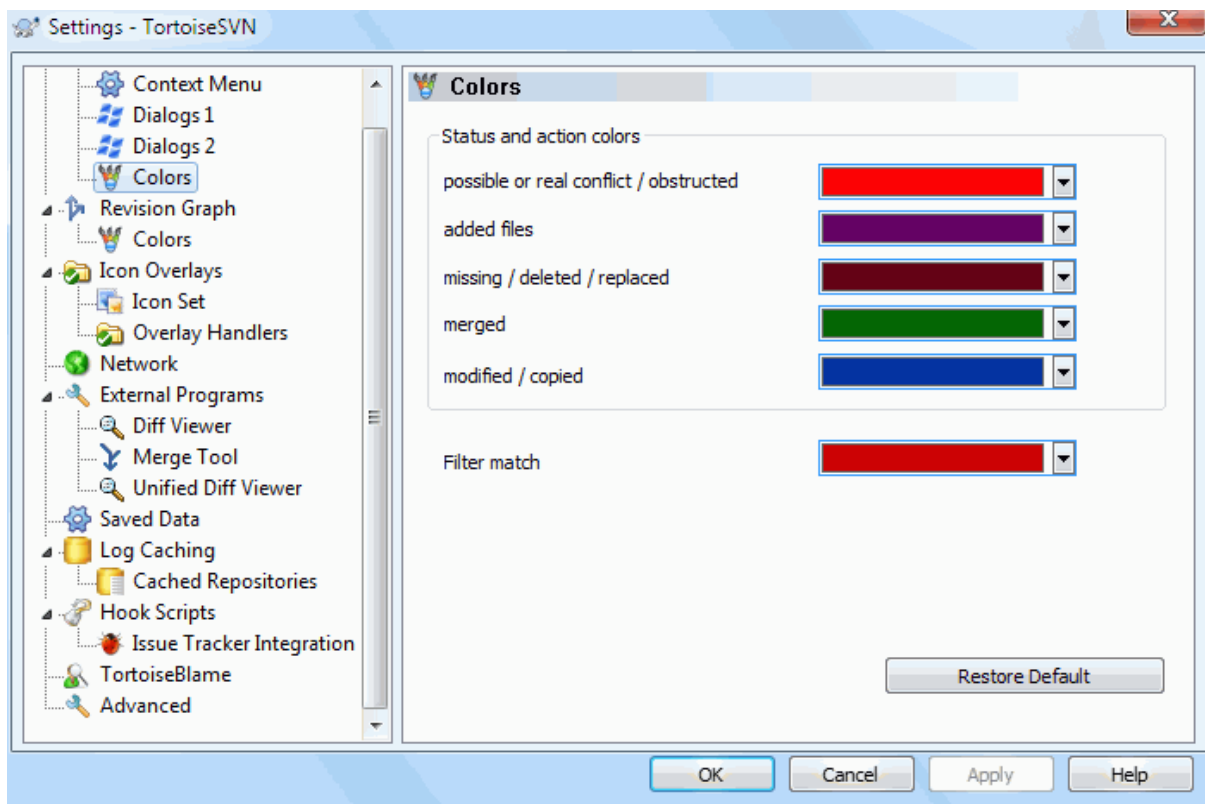
Some servers however can't handle the multiple requests this causes or when not configured correctly treat so many requests as something bad and start blocking them. In this case you can disable the pre-fetching here.

#### Mostrar externos

If this box is checked (default state), then the repository browser shows files and folders that are included with the `svn:externals` property as normal files and folders, but with an overlay icon to mark them as from an external source.

As with the pre-fetch feature explained above, this too can put too much stress on weak servers. In this case you can disable this feature here.

### 4.31.1.5. Configuración de colores de TortoiseSVN



**Figura 4.77. El diálogo Configuración, página de Colores**

Este diálogo le permite configurar a su gusto los colores de los textos utilizados en los diálogos de TortoiseSVN.

#### Conflicto posible o real / obstruido

Ha ocurrido un conflicto durante la actualización, o puede ocurrir durante la fusión. La actualización está obstruida por una carpeta o un archivo sin versionar que ya existe en su copia de trabajo, y tiene el mismo nombre que uno versionado.

Este color se utiliza también para los mensajes de error en los diálogos de progreso.

#### Archivos añadidos

Ítems añadidos al repositorio.

#### Faltante / borrado / reemplazado

Ítems borrados del repositorio, faltantes en la copia de trabajo, o borrados de la copia de trabajo y reemplazados con otro archivo del mismo nombre.

**Fusionado**

Cambios del repositorio que se han fusionado satisfactoriamente con su copia de trabajo sin crear ningún conflicto.

**Modificado / copiado**

Añadido con historia, o rutas copiadas en el repositorio. También se utiliza en el diálogo de registro para las entradas que incluyen ítems copiados.

**Nodo eliminado**

Un ítem que ha sido eliminado del repositorio.

**Nodo añadido**

Un ítem que ha sido añadido al repositorio, mediante una operación añadir, copiar o mover.

**Nodo renombrado**

Un ítem que ha sido renombrado dentro del repositorio.

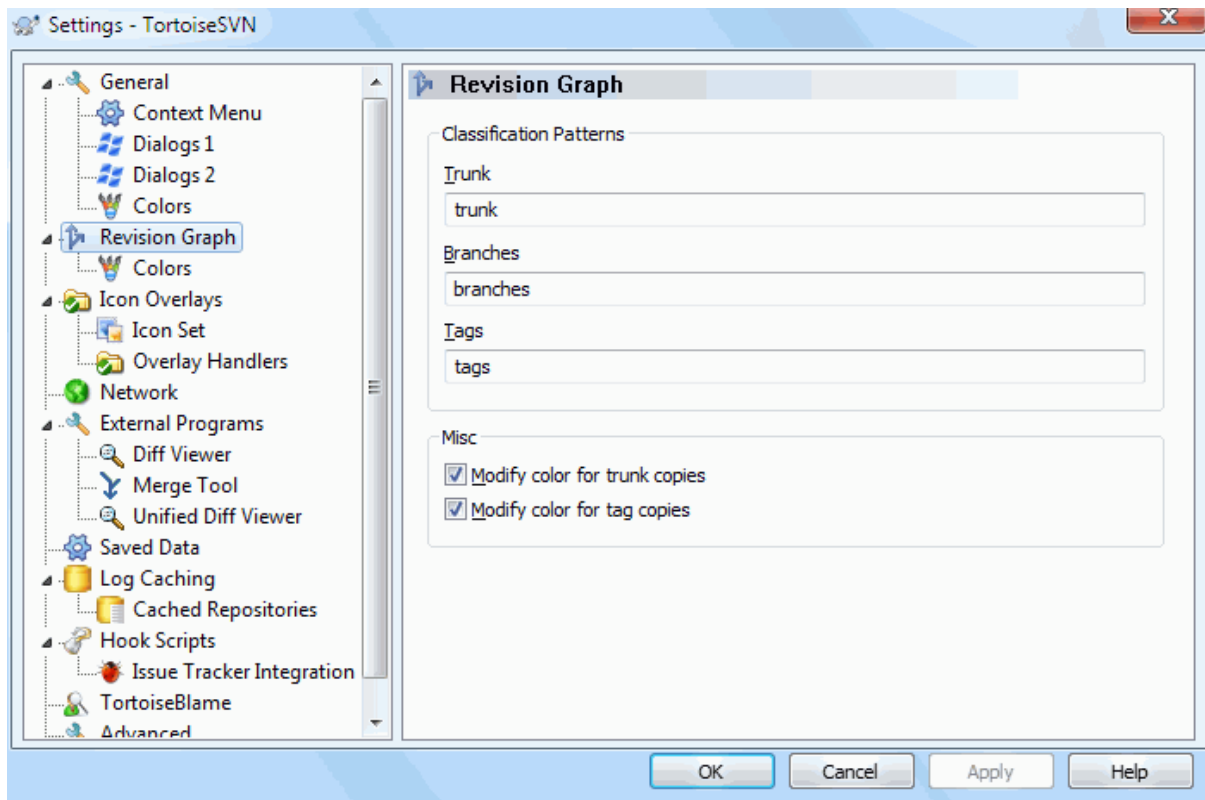
**Nodo reemplazado**

El ítem original ha sido borrado y un nuevo ítem con el mismo nombre le reemplaza.

**Concordancia de filtro**

Cuando use filtrado en el diálogo de registro, los términos buscados son resaltados en los resultados usando este color.

**4.31.2. Configuración del gráfico de revisión**



**Figura 4.78. El diálogo Configuración, página Gráfico de revisión**

**Patrones de clasificación**

El gráfico de revisiones intenta mostrar una imagen más clara de la estructura de su repositorio distinguiendo entre tronco, ramas y etiquetas. Dado que no existe dicha clasificación interna en Subversion, esta información se extrae desde los nombres de las rutas. Las configuraciones por defecto asumen que utiliza los nombres

convencionales en inglés tal y como se sugieren en la documentación de Subversion (trunk, tags y branches), pero por supuesto puede cambiarlo.

Especifique los patrones que se utilizarán para reconocer estos patrones en las tres cajas de texto propuestas. Los patrones se comprobarán sin tener en cuenta mayúsculas y minúsculas, pero debe especificarlas en minúsculas. Los caracteres comodín \* y ? funcionarán como siempre, y puede utilizar ; para separar patrones múltiples. No incluya ningún espacio en blanco extra, porque se incluirá en la especificación de concordancia.



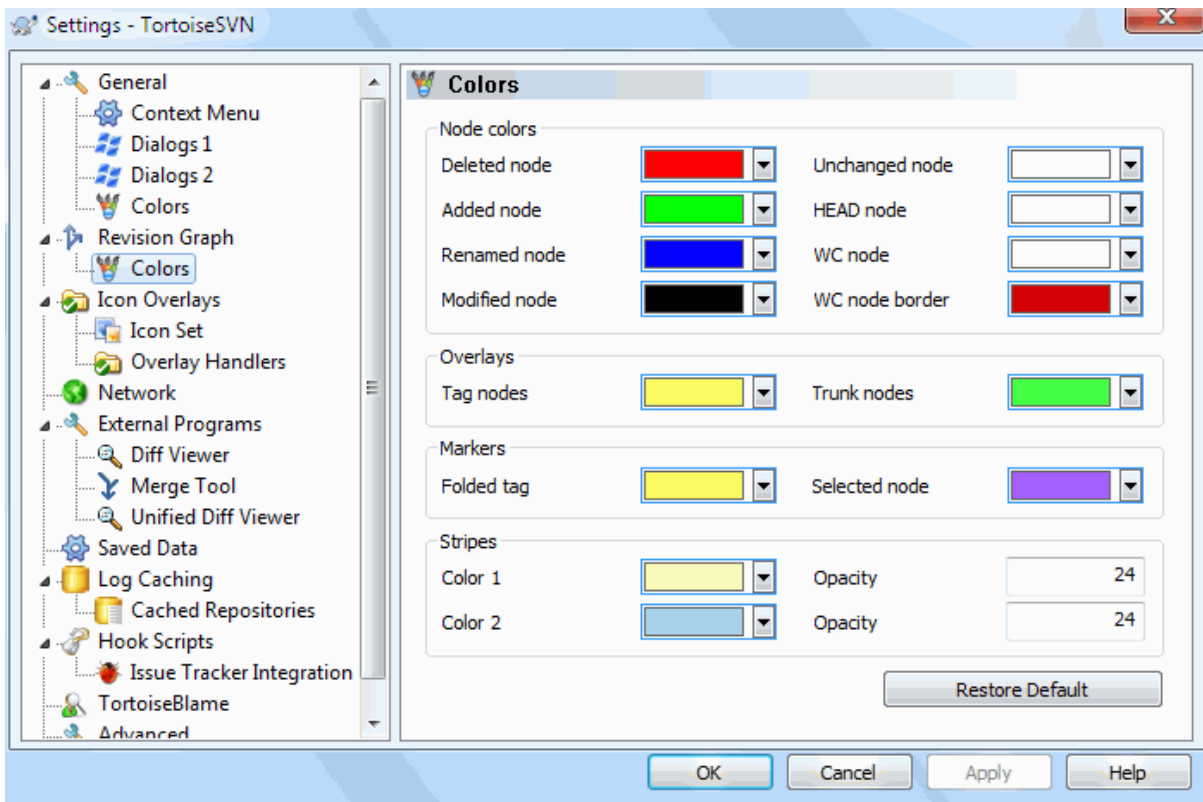
### Commit tag detection

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

#### Modificar colores

Los colores se utilizan en el gráfico de revisiones para indicar el tipo de nodo, es decir, si un nodo se añadió, eliminó, renombró. Para ayudar a distinguir las clasificaciones de los nodos, puede permitir que el gráfico de revisiones fusione colores para dar una indicación de tanto el tipo de nodo y la clasificación. Si la casilla está marcada, se utiliza la fusión. Si la casilla se desmarca, el color se utiliza sólo para indicar el tipo de nodo. Utilice el diálogo de selección de colores para especificar los colores específicos usados.

#### 4.31.2.1. Colores del gráfico de revisión



**Figura 4.79. El diálogo Configuración, página Colores del gráfico de revisión**

Esta página le permite configurar los colores utilizados. Tenga en cuenta que el color especificado aquí es el color sólido. la mayoría de los nodos se colorean utilizando una fusión del color del tipo de nodo, el color de fondo y opcionalmente el color de clasificación.

#### Nodo eliminado

Ítems que se han eliminado y no se han copiado a ningún otro lugar en la misma revisión.

**Nodo añadido**

Ítems añadidos nuevos, o copiados (añadidos con historia).

**Nodo renombrado**

Ítems eliminados de un lugar y añadidos en otro en la misma revisión.

**Nodo modificado**

Modificaciones simples sin ningún añadir o eliminar.

**Nodo sin cambios**

Puede utilizarse para mostrar la revisión utilizada como origen de una copia, incluso cuando no hay ningún cambio (sobre el ítem cuyo gráfico se va a mostrar) en esa revisión.

**Nodo HEAD**

HEAD actual en el repositorio.

**Nodo WC**

Si opta por mostrar un nodo extra para su copia de trabajo adjuntado a la revisión de su última confirmación en el gráfico, utilice este color.

**Borde del nodo WC**

Si opta por mostrar si la copia de trabajo está modificada, utilice este borde de color en el nodo WC cuando se encuentren modificaciones.

**Nodos etiqueta**

Los nodos clasificados como etiquetas pueden fundirse con este color.

**Nodos tronco**

Los nodos clasificados como tronco pueden fundirse con este color.

**Marcadores de etiquetas plegadas**

Si utiliza plegado de etiquetas para ahorrar espacio, las etiquetas se marcan en la copia origen utilizando un bloque de este color.

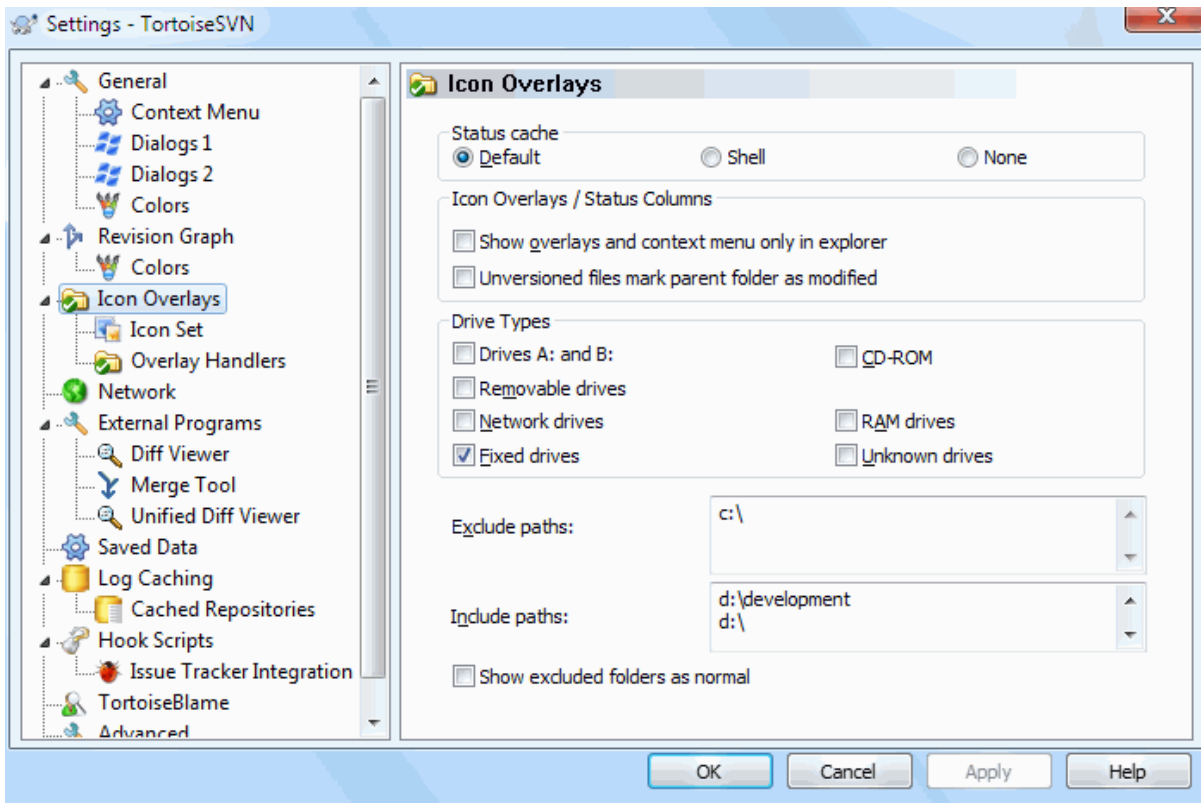
**Marcadores del nodo seleccionado**

Cuando hace click con el botón izquierdo en un nodo para seleccionarlo, el marcador utilizado para indicar la selección es un bloque de este color.

**Bandas**

Estos colores se utilizan cuando el gráfico se divide en sub-árboles y el fondo se colorea en bandas alternadas para ayudar a distinguir los distintos árboles.

### 4.31.3. Configuración de los íconos superpuestos



**Figura 4.80. El diálogo Configuración, página de íconos superpuestos**

Esta página le permite seleccionar los ítems para los que TortoiseSVN mostrará íconos superpuestos.

Dado que lleva bastante tiempo obtener el estado de una copia de trabajo, TortoiseSVN utiliza una caché para almacenar el estado de forma que el explorador no se vea demasiado atosigado cuando se muestren los íconos superpuestos. Puede elegir qué tipo de caché deberá utilizar TortoiseSVN de acuerdo a sus sistema y al tamaño de las copias de trabajo aquí:

Por defecto

Cachea toda la información de estado en un proceso separado (TSVNCache.exe). Ese proceso observa todas las unidades en busca de cambios y obtiene el estado de nuevo si se modifican archivos dentro de una copia de trabajo. El proceso se ejecuta con la menor prioridad posible para que el rendimiento de otros programas no se resientan por su culpa. Eso también significa que la información de estado no se actualiza en *tiempo real* sino que puede llevar unos pocos segundos hasta que los íconos superpuestos cambien.

Ventaja: los íconos superpuestos muestran el estado de forma recursiva, es decir, si un archivo muy dentro de su copia de trabajo se modifica, todas las carpetas padre hasta la raíz de su copia de trabajo mostrarán el ícono superpuesto de modificación. Y dado que el proceso puede enviar notificaciones al shell, los íconos superpuestos del árbol de la izquierda normalmente también cambiarán.

Desventaja: el proceso se ejecuta constantemente, incluso si no está trabajando en sus proyectos. Además utiliza alrededor de 10-50 MB de RAM dependiendo del número y tamaño de sus copias de trabajo.

Shell

El cacheo se realiza directamente dentro de la dll de la extensión del shell, pero sólo para la carpeta actualmente visible. Cada vez que navega a otra carpeta, se obtiene de nuevo la información de estado.

Ventaja: necesita únicamente muy poca memoria (alrededor de 1 MB de RAM) y puede mostrar el estado en *tiempo real*.

Desventaja: Dado que sólo se hace caché de una carpeta, las superposiciones no muestran el estado recursivamente. Para copias de trabajo grandes, mostrar una carpeta en el explorador puede llevar más tiempo que con la caché por defecto. Además, la columna tipo-mime no está disponible.

#### Ninguno

Con esta configuración, TortoiseSVN no obtiene ningún estado en el Explorador. Por esa causa, los archivos no tendrán íconos superpuestos y las carpetas sólo tendrán un ícono superpuesto 'normal' si están versionadas. No se mostrará ningún otro ícono superpuesto, y tampoco estarán disponibles las columnas extra.

Ventaja: no utiliza memoria adicional en absoluto y no ralentiza de ninguna forma el Explorador mientras se navega.

Desventaja: La información de estado de los archivos y carpetas no se muestra en el Explorador. Para ver si sus copias de trabajo están modificadas, tendrá que utilizar el diálogo “Comprobar modificaciones”.

Por defecto, los íconos superpuestos y los menús contextuales aparecerán tanto en todos los diálogos abrir/guardar como en el Explorador de Windows. Si desea que aparezcan *únicamente* en el Explorador de Windows, marque **Mostrar íconos superpuestos y menú contextual sólo en el explorador box**.

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.

También puede elegir marcar las carpetas como modificadas si contienen ítems sin versionar. Esto puede ser útil para recordarle que ha creado nuevos archivos que aún no están versionados. Esta opción sólo está disponible cuando utiliza la opción de la caché de estados *por defecto* (más abajo).

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.

El siguiente grupo le permite seleccionar qué tipos de almacenamiento deberían mostrar íconos superpuestos. Por defecto, sólo los discos duros están seleccionados. Incluso puede deshabilitar todos los íconos superpuestos, pero ¿qué gracia tendría eso?

Las unidades de red pueden ser muy lentas, así que por defecto no se muestran los íconos para las copias de trabajo que se encuentren en unidades de red.

Las unidades USB Flash parecen ser un caso especial en el que el tipo de unidad lo identifica el propio dispositivo. Algunas aparecen como discos duros, y otros como discos extraíbles.

**Excluir Rutas** se usa para decirle a TortoiseSVN en qué rutas *no* debe mostrar íconos superpuestos ni columnas de estado. Esto es útil si tiene algunas copias de trabajo muy grandes que contienen únicamente librerías que no cambian en absoluto y por tanto no necesitan los íconos superpuestos, o si solo desea que TortoiseSVN tome en cuenta carpetas específicas.

Se asume que cualquier ruta que especifique aquí se aplicará recursivamente, por tanto ninguna de las carpetas hijas mostrará íconos superpuestos. Si quiere excluir *solo* la carpeta nombrada, agregue ? después de la ruta.

Lo mismo se aplica a **Incluir rutas**. Excepto que para esas rutas los íconos superpuestos se muestran incluso si están deshabilitadas para ese tipo de unidad en concreto, o por una ruta de exclusión especificada más arriba.

Los usuarios preguntan a veces como interactúan estas tres configuraciones. Para cualquier ruta verifique las listas de incluidos y excluidos, buscando hacia arriba en la estructura de directorio hasta que se encuentra una concordancia. Cuando se encuentre la primera concordancia, se obedece la regla de inclusión o exclusión. Si existe un conflicto, una especificación para un directorio simple tiene precedencia sobre una especificación recursiva, luego la inclusión tiene precedencia sobre la exclusión.

An example will help here:

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

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

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

TSVNCache.exe también utiliza estas rutas para restringir su escaneo. Si desea que sólo mire en determinadas carpetas, deshabilite todos los tipos de unidades e incluya sólo las carpetas que específicamente quiere que sean escaneadas.



### 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
```

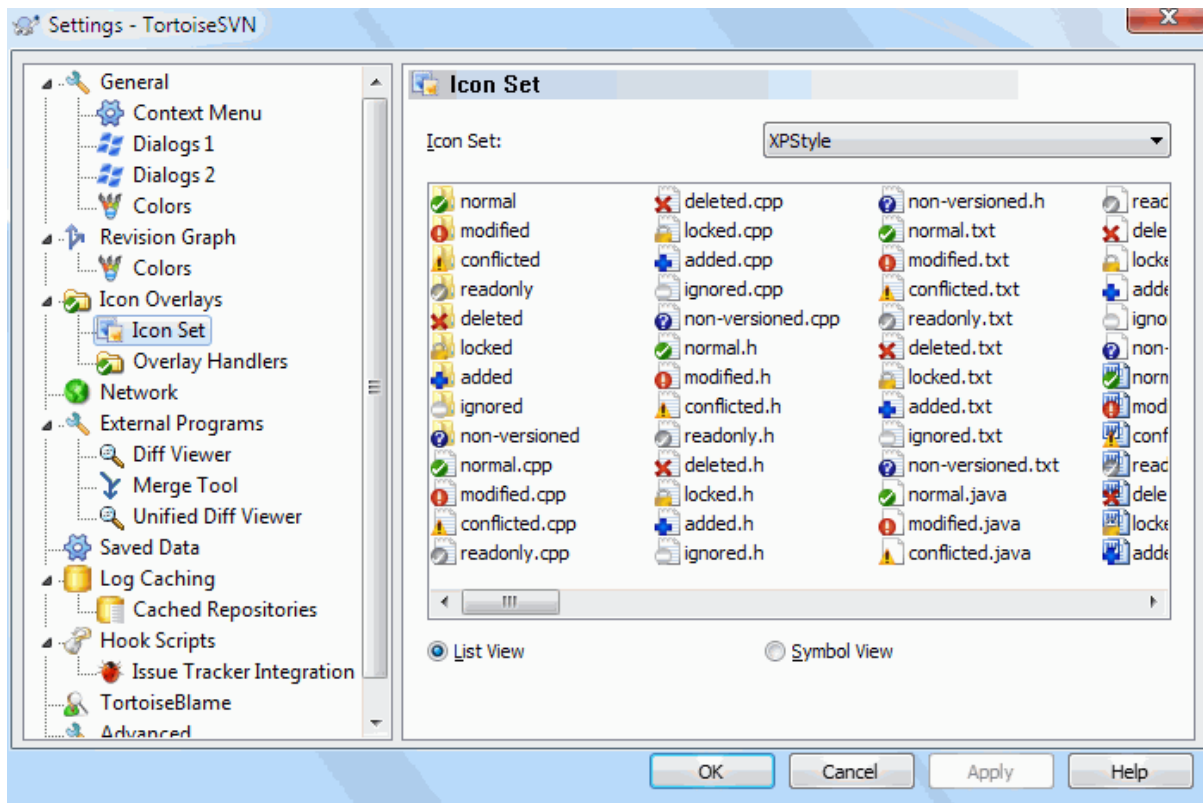
However this can cause the overlays not to update, as TSVNCache will only receive one notification when a file changes, and that is normally for the original path. This means that your overlays on the subst path may never be updated.

Una forma fácil de evitar esto es excluir de los íconos superpuestos la ruta original, de forma que los íconos superpuestos aparezcan ahora en la ruta subst.

Sometimes you will exclude areas that contain working copies, which saves TSVNCache from scanning and monitoring for changes, but you still want a visual indication that a folder contains a working copy. The **Show excluded root folders as 'normal'** checkbox allows you to do this. With this option, working copy root folders in any excluded area (drive type not checked, or specifically excluded) will show up as normal and up-to-date, with a green check mark. This reminds you that you are looking at a working copy, even though the folder overlays may not be correct. Files do not get an overlay at all. Note that the context menus still work, even though the overlays are not shown.

Como una excepción especial a esto, las unidades A: y B: nunca se consideran para la opción **Mostrar carpetas excluidas como normales**. El motivo es que Windows se ve obligado a mirar en la unidad, lo que puede resultar en un retraso de varios segundos cuando se inicia el Explorador, incluso si su PC no tiene una unidad de disquete.

### 4.31.3.1. Selección del conjunto de íconos

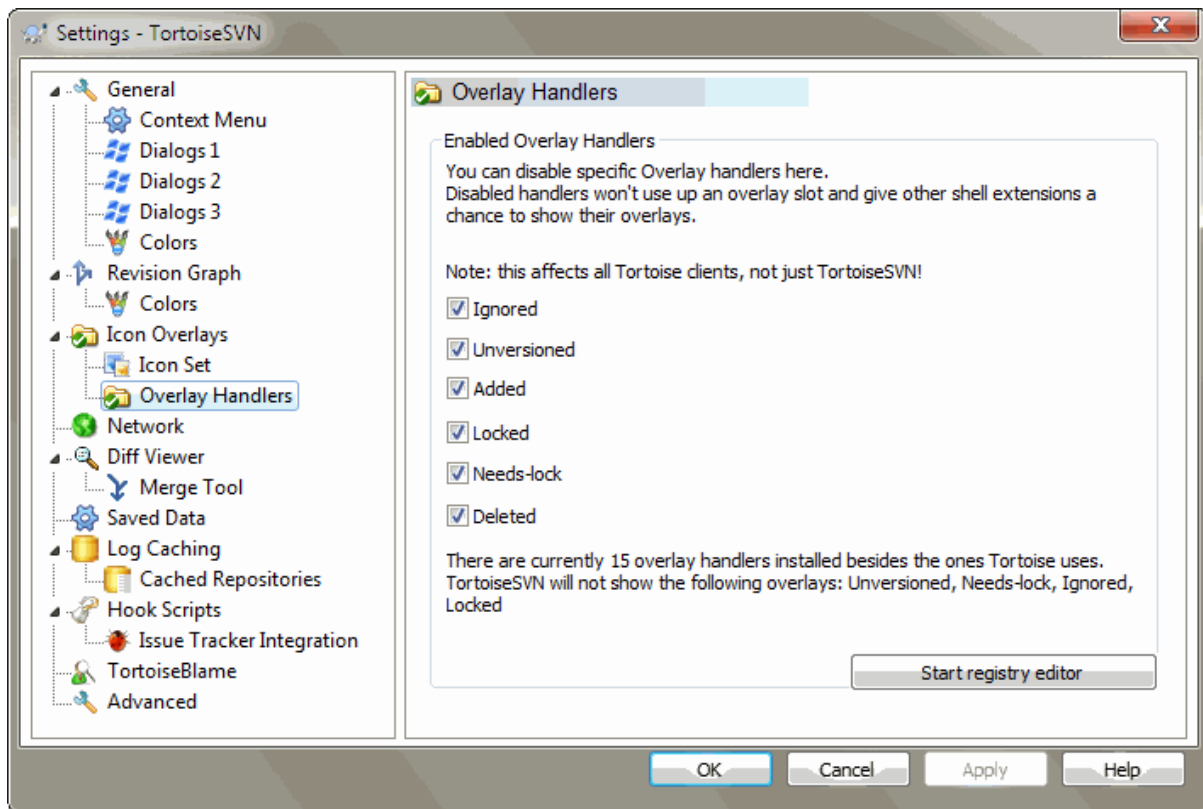


**Figura 4.81.** El diálogo Configuración, página de Conjunto de íconos

También puede cambiar el conjunto de íconos superpuestos al que más le guste. Tenga en cuenta que si cambia el conjunto de superposiciones, puede tener que reiniciar su ordenador para que estos cambios surtan efecto.



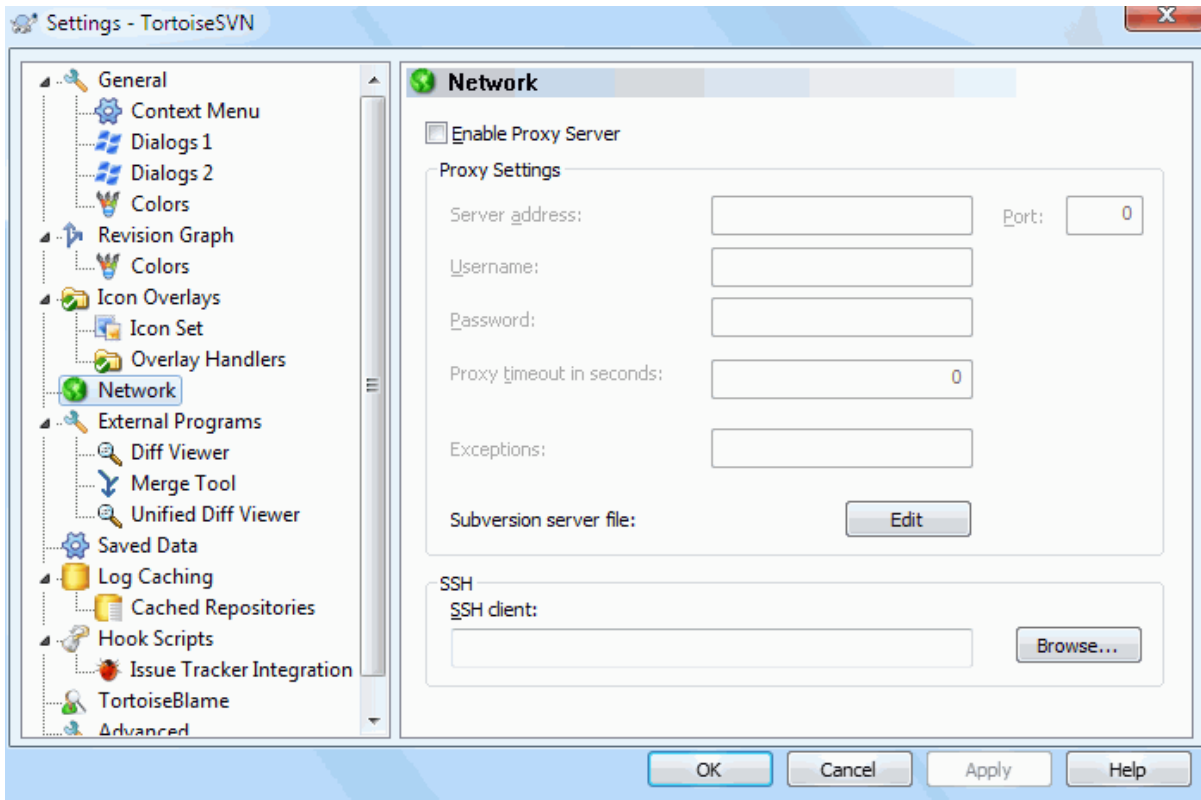
### 4.31.3.2. Manejadores de superpuestos habilitados



**Figura 4.82. El Diálogo de Configuración, Página de Manejadores de Íconos**

Because the number of overlays available is severely restricted, you can choose to disable some handlers to ensure that the ones you want will be loaded. Because TortoiseSVN uses the common TortoiseOverlays component which is shared with other Tortoise clients (e.g. TortoiseCVS, TortoiseHg) this setting will affect those clients too.

### 4.31.4. Configuración de red



**Figura 4.83. El diálogo Configuración, página de Red**

Aquí puede configurar su servidor proxy, si necesita uno para atravesar el cortafuegos de su compañía.

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.

También puede especificar qué programa debe utilizar TortoiseSVN para establecer una conexión segura a un repositorio svn+ssh. Le recomendamos que utilice TortoisePlink.exe. Ésta es una versión del popular programa Plink, y se incluye con TortoiseSVN, pero está compilado como una aplicación sin ventanas, por lo que no verá una ventana DOS molestando cada vez que se autentifica.

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"
```

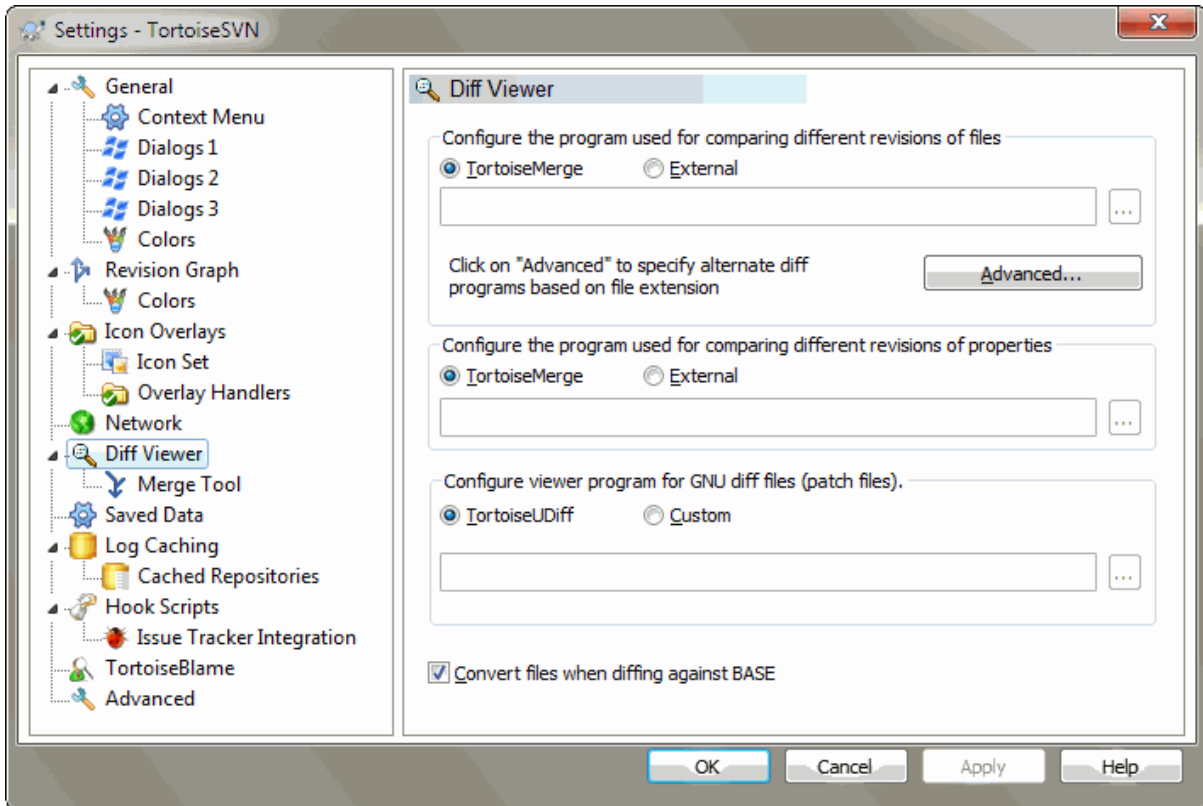
Un efecto colateral de no tener una ventana es que no hay sitio para que vayan los mensajes de error, por lo que si la autenticación falla simplemente obtendrá un mensaje diciendo algo como "Unable to write to standard output" ("No se puede escribir en la salida estándar"). Por esta razón le recomendamos que primero pruebe su configuración con el Plink estándar. Cuando todo funcione, puede utilizar TortoisePlink con exactamente los mismos parámetros.

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 evitar ser preguntado por una contraseña repetidamente, puede considerar utilizar una herramienta de caché de contraseñas como Pageant. También está disponible desde el sitio web de 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. Configuración de programas externos



**Figura 4.84. El diálogo Configuración, página de Visor de diferencias**

Aquí puede definir sus propios programas de diferencias/fusión que TortoiseSVN debería utilizar. La configuración básica es utilizar TortoiseMerge que se instala junto con TortoiseSVN.

En [Sección 4.11.6, “Herramientas externas de diferencias/fusión”](#) encontrará una lista de algunos de los programas externos de diferencias / fusión que la gente está utilizando con TortoiseSVN.

#### 4.31.5.1. Visor de diferencias

Se puede utilizar un programa externo de diferencias para comparar diferentes revisiones de archivos. El programa externo necesitará obtener los nombres de los archivos de la línea de comando, junto con otras opciones de la línea de comandos. TortoiseSVN usa sustitución de parámetros con prefijo %. Cuando se encuentra uno de estos se sustituirá por el valor apropiado. El orden de los parámetros dependerá del programa de Diferencias que utilice.

%base

El archivo original sin sus cambios

%bname

La ventana de título para el archivo base

%nqname

La ventana de título para el archivo base, sin comillas

%mine

Su propio archivo, con sus cambios

%yname

El título de la ventana para su archivo

`%nqynname`  
La ventana de título para tu archivo, sin comillas

`%burl`  
La dirección URL del archivo original, si está disponible

`%nqburl`  
La URL del archivo original, si está disponible, sin comillas

`%yurl`  
La dirección URL del segundo archivo, si está disponible

`%nqyurl`  
La URL del segundo archivo, si está disponible, sin comillas

`%brev`  
La revisión del archivo original, si está disponible

`%nqbrev`  
La revisión del archivo original, si está disponible, sin comillas

`%yrev`  
La revisión del segundo archivo, si está disponible

`%nqyrev`  
La revisión del segundo archivo, si está disponible, sin comillas

`%peg`  
La revisión de paridad, si está disponible

`%nqpeg`  
La marca (señal) de la revisión, si está disponible, sin comillas

`%fname`  
El nombre del archivo. Éste es una cadena vacía si dos diferentes archivos son diferenciados, en lugar de dos estados del mismo archivo.

`%nqfname`  
El nombre del archivo, sin comillas

Los títulos de las ventanas no son exactamente los nombres de los ficheros. TortoiseSVN los trata como un nombre para mostrar y crea los nombres en consecuencia. Así que, por ejemplo, si usted está haciendo una diferencia de un fichero en la revisión 123 con un fichero de su copia de trabajo, los nombres serán nombre de fichero: revisión 123 y nombre de fichero : copia de trabajo.

For example, with ExamDiff Pro:

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

or with KDiff3:

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

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:%yname
                        %base %mine
```

or with UltraCompare:

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

or with DiffMerge:

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

Si utiliza la propiedad `svn:keywords` para expandir palabras clave, y en particular la `revision` de un archivo, entonces puede haber una diferencia entre archivos que se debe únicamente al valor actual de esta palabra clave. También puede ocurrir que si utiliza `svn:eol-style = native` el archivo BASE tenga finales de líneas formados únicamente por LF mientras que sus archivos tendrán finales de líneas CR-LF. TSVN normalmente oculta esas diferencias automáticamente ya que primero parsea el archivo BASE para expandir las palabras clave y los finales de líneas antes de realizar la operación de diferenciar. Sin embargo, esto puede tomar bastante tiempo con archivos grandes. Si se desmarca **Convertir archivos cuando se diferencian contra BASE**, TortoiseSVN se saltará el preprocesado de los archivos.

También puede especificar una herramienta distinta de diferencias para utilizarla en las propiedades de Subversion. Dado que estas tienden a ser cadenas cortas de texto simple, puede que desee utilizar un visor más compacto.

Si ha configurado una herramienta de diferencias distinta, puede acceder a TortoiseMerge y a la herramienta de terceros desde los menús contextuales. **Menú contextual** → **Diff** utiliza la herramienta de diferencias primaria, y **Mayúsculas+ Menú contextual** → **Diff** utiliza la herramienta de diferencias secundaria.

At the bottom of the dialog you can configure a viewer program for unified-diff files (patch files). No parameters are required. The Default setting is to use TortoiseUDiff which is installed alongside TortoiseSVN, and colour-codes the added and removed lines.

Ya que Unified Diff es solo un formato de texto, usted puede utilizar su editor de texto favorito si lo prefiere.

#### 4.31.5.2. Herramienta de fusión

Un programa de fusión externo que se utiliza para resolver archivos en conflicto. La sustitución de parámetros se utiliza de la misma forma que el Programa de Diferencias.

`%base`

El archivo original sin ningún cambio, ni suyo ni de otros

`%bname`

La ventana de título para el archivo base

`%nqbname`

La ventana de título para el archivo base, sin comillas

`%mine`

Su propio archivo, con sus cambios

`%yname`

El título de la ventana para su archivo

`%nqyname`

La ventana de título para tu archivo, sin comillas

%theirs

El archivo tal cual estaba en el repositorio

%tname

El título de la ventana del archivo en el repositorio

%nqtname

La ventana de título para el archivo en el repositorio, sin comillas

%merged

El archivo en conflicto, el resultado de la operación de fusión

%mname

El título de la ventana para el archivo fusionado

%nqmname

La ventana de título para el archivo unido, sin comillas

%fname

El nombre del archivo con conflicto

%nqfname

El nombre del archivo con conflicto, sin comillas

For example, with Perforce Merge:

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

or with KDiff3:

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

or with Araxis:

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

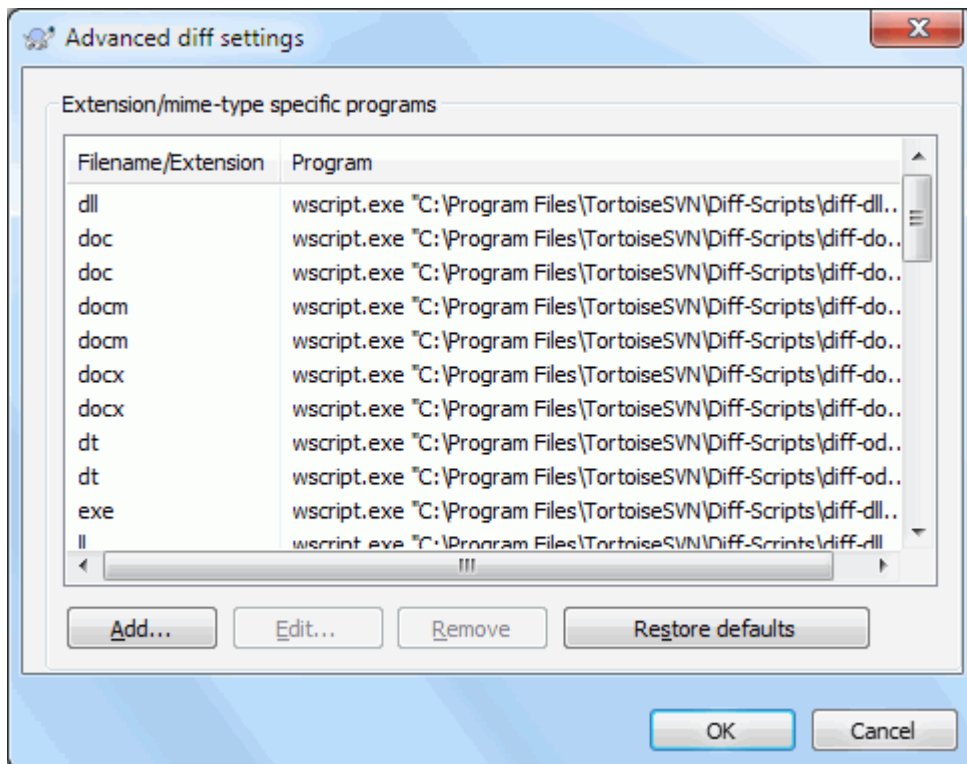
or with WinMerge (2.8 or later):

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

or with DiffMerge:

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

### 4.31.5.3. Configuración avanzada de diferencias / fusiones

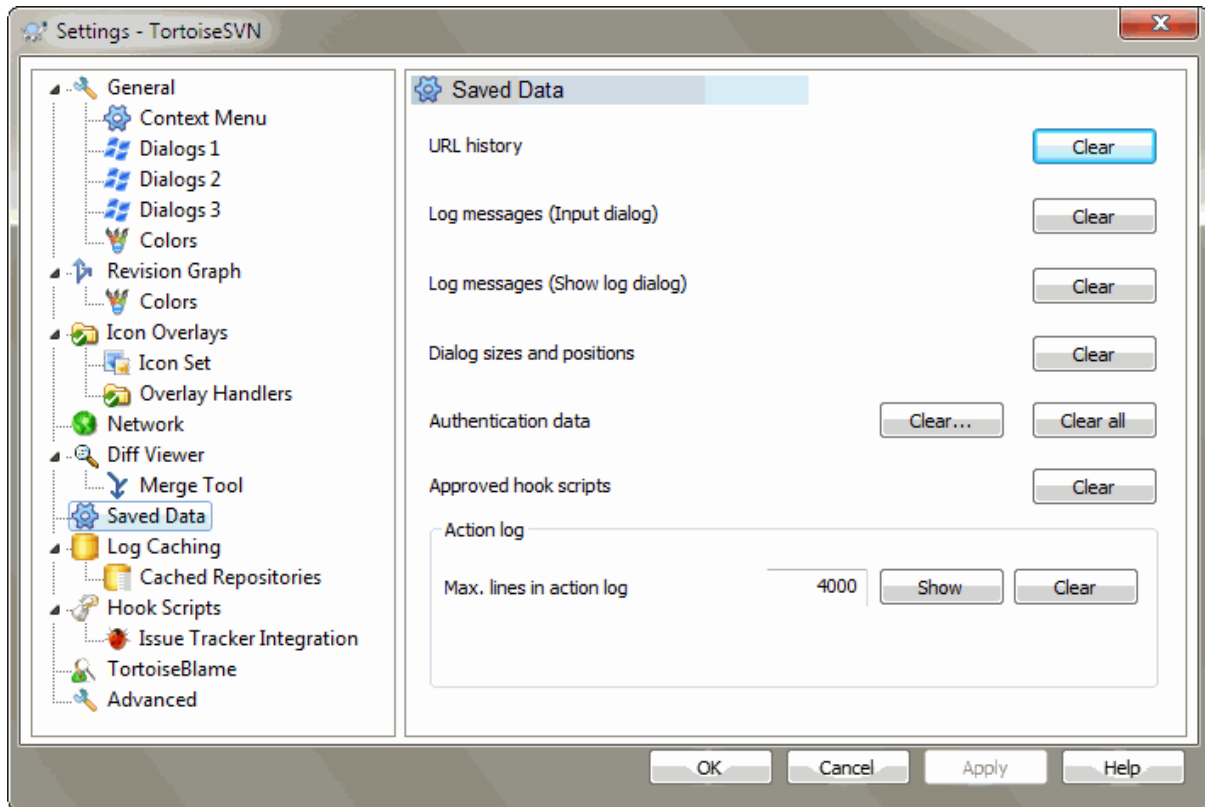


**Figura 4.85. El diálogo Configuración, diálogo de Diferencias/Fusión avanzadas**

En las configuraciones avanzadas, puede definir un programa de diferencias y fusión diferente por cada extensión de archivo. Por ejemplo, podría asociar Photoshop como el programa de “Diferencias” para archivos .jpg :-). También puede asociar la propiedad `svn:mime-type` con un programa de diferencias o fusión.

Para asociar utilizando una extensión de archivo, necesita especificar la extensión. Utilice `.bmp` para describir archivos de bitmaps de Windows. Para asociar utilizando la propiedad `svn:mime-type`, especifique el tipo MIME, incluyendo una barra, por ejemplo `text/xml`.

### 4.31.6. Datos de configuración almacenados



**Figura 4.86. El diálogo Configuración, página de Datos almacenados**

Por su comodidad, TortoiseSVN almacena muchas de las opciones que utiliza, y recuerda dónde ha estado últimamente. Si desea limpiar esa caché de datos, puede hacerlo aquí.

#### Historia de URL

Siempre que obtiene una copia de trabajo, fusione cambios o utilice el navegador de repositorios, TortoiseSVN guarda un registro de las últimas URLs utilizadas y las ofrece en una caja desplegable. A veces esa lista se llena de URLs desfasadas por lo que puede ser útil limpiarla periódicamente.

Si quiere eliminar un único ítem de uno de los cuadro de texto desplegables puede hacerlo allí mismo. Simplemente haga click en la flecha para desplegar el cuadro, mueva el ratón sobre el ítem que quiere eliminar y pulse Mayúsculas+Supr.

#### Mensajes de registro (diálogo de entrada)

TortoiseSVN almacena los mensajes de registro de confirmación que introduce. Estos se almacenan por cada repositorio, por lo que si accede a muchos repositorios la lista puede crecer bastante.

#### Mensajes de registro (diálogo Mostrar registro)

TortoiseSVN almacena en caché los mensajes de registro obtenidos por el diálogo Mostrar registro para ahorrar tiempo la próxima vez que muestre el registro. Si alguien edita un mensaje de registro que ya estuviera en su caché, no verá el cambio hasta que limpie la caché. La caché de mensajes de registro se habilita en la pestaña Caché de registros.

#### Tamaños y posiciones de los diálogos

Muchos diálogos almacenan el tamaño y la posición en pantalla que tenían la última vez.

#### Datos de autenticación

Cuando se autentifica con un servidor de Subversion, el usuario y la contraseña se cachean localmente para que no tenga que seguir introduciéndolos. Puede querer limpiar esto por razones de seguridad, o porque quiere acceder al repositorio bajo un nombre de usuario diferente ... ¿sabe John que está utilizando su PC?



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

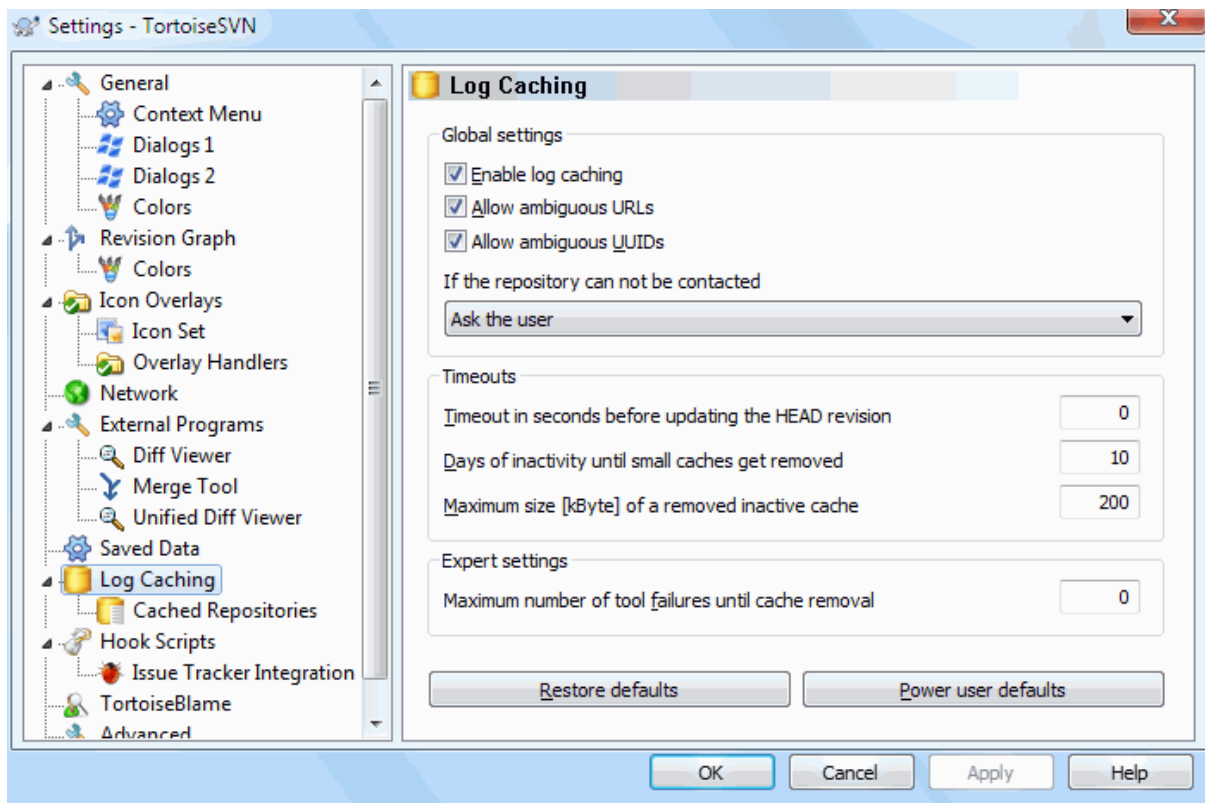
#### Registro de acciones

TortoiseSVN almacena un registro de todo lo que ha escrito en sus diálogos de progreso. Esto puede ser útil si, por ejemplo, desea comprobar qué ocurrió en un comando de actualización reciente.

El archivo de registro tiene un tamaño límite y cuando crece demasiado el contenido más antiguo se descarta. Por defecto se guardan 4000 líneas, pero puede personalizar este número.

Desde aquí puede ver el contenido del archivo de registro, y también limpiarlo.

### 4.31.7. Caché de registro



**Figura 4.87. El diálogo Configuración, página Caché de registro**

Este diálogo le permite configurar la característica de caché de registro de TortoiseSVN, que retiene una copia local de los mensajes de registro y rutas cambiadas para evitar descargas desde el servidor que llevan tiempo. Al utilizar la caché de registro acelerará dramáticamente el diálogo de registro y el diálogo de revisiones. Otra característica útil es que los mensajes de registro serán accesibles cuando esté desconectado.

#### Habilitar la caché de registro

Habilita la caché de registro siempre que se pidan datos de registro. Si se marca, los datos se recuperarán de la caché cuando estén disponibles allí, y cualquier mensaje que no esté en la caché se recuperará del servidor y se añadirá a la caché.

Si la caché se deshabilita, los datos siempre se recuperarán directamente del servidor y no se almacenarán localmente.

#### Permitir URLs ambiguas

Ocasionalmente puede tener que conectarse a un servidor que utiliza la misma URL para todos los repositorios. Las versiones antiguas de `svnbridge` harían esto. Si necesita acceder a este tipo de repositorios necesitará marcar esta opción. Si no lo necesita, deje esta casilla desmarcada para mejorar el rendimiento.

#### Permitir UUIDs ambiguas

Algunos proveedores de hospedaje dan a todos sus repositorios el mismo UUID. Puede que incluso le haya pasado si ha copiado una carpeta de repositorio para crear otro. Esto es una mala idea por toda clase de motivos - un UUID debería ser *único*. Sin embargo, la caché de registro todavía podrá trabajar en esta situación si marca esta casilla. Si no lo necesita, déjela desmarcada para mejorar el rendimiento.

#### Si no se puede contactar con el repositorio

Si está trabajando sin conexión, o si el servidor del repositorio está caído, la caché de registro aún puede utilizarse para suministrar mensajes de registro que estén ya almacenados en la caché. Por supuesto la caché puede no estar actualizada, por lo que hay opciones que le permiten seleccionar cuándo debería utilizarse esta característica.

Cuando los datos de registro se toman de la caché sin contactar con el servidor, el diálogo que utiliza dichos mensajes mostrará el estado desconectado en su barra de título.

#### Tiempo límite antes de actualizar la revisión HEAD

Cuando invoca al diálogo de registro normalmente querrá contactar con el servidor para comprobar si hay mensajes de registro nuevos. Si el tiempo límite establecido aquí es distinto de cero, sólo se contactará con el servidor cuando el tiempo límite haya pasado desde el último contacto. Esto puede reducir los viajes al servidor si abre el diálogo de registro frecuentemente y el servidor es lento, pero los datos mostrados pueden no estar completamente actualizados. Si desea utilizar esta característica le sugerimos un valor de compromiso de 300 (5 minutos).

#### Días de inactividad hasta que las cachés pequeñas se eliminan

Si navega por muchos repositorios acumulará un montón de cachés de registro. Si no las utiliza de forma activa, las cachés no crecerán mucho, por lo que TortoiseSVN por defecto las purgará tras un periodo de tiempo. Utilice este ítem para controlar la purga de la caché.

#### Tamaño máximo de las cachés inactivas eliminadas

Las cachés más grandes son más costosas de readquirir, por lo que TortoiseSVN sólo purga cachés pequeñas. Ajuste el límite con este valor.

#### Máximo número de fallos de herramienta hasta la eliminación de la caché

Ocasionalmente algo puede ir mal con la caché y provocar un fallo. Si esto ocurre la caché normalmente se elimina automáticamente para evitar la recurrencia del problema. Si utiliza la versión nocturna y menos estable puede optar por mantener la caché de todas formas.

### 4.31.7.1. Repositorios cacheados

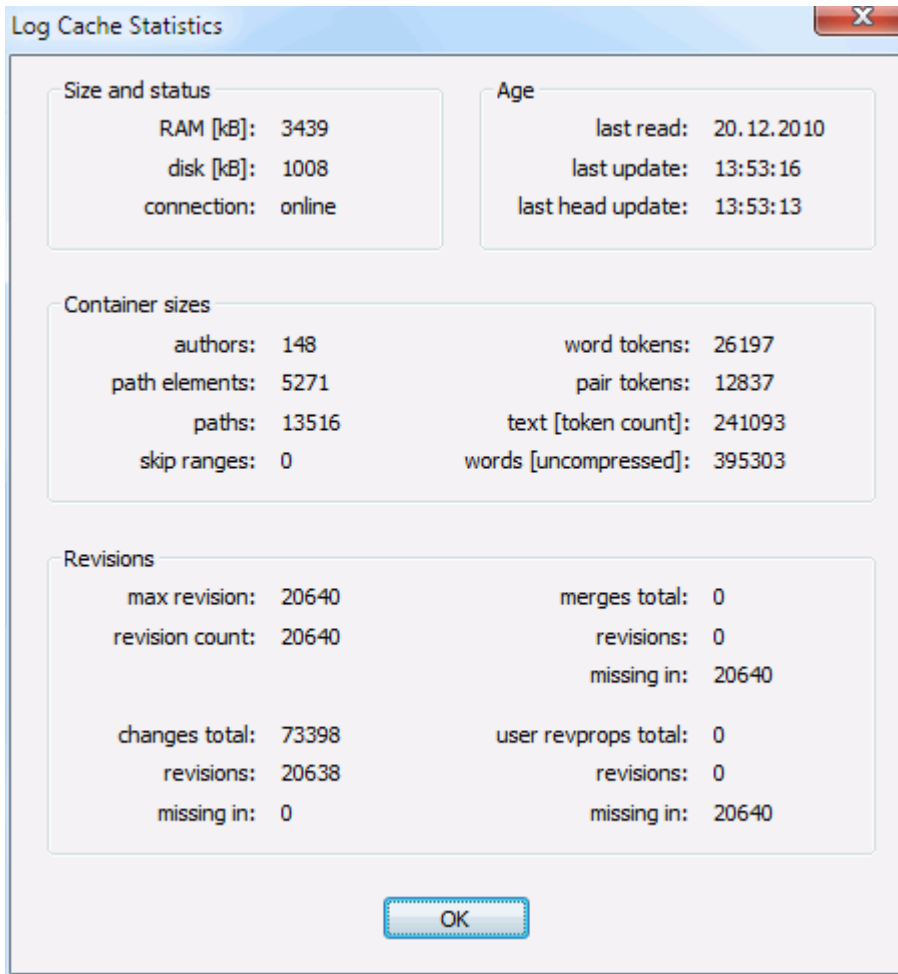
En esta página puede ver una lista de repositorios que están almacenados en la caché local, y el espacio utilizado para la caché. Si selecciona uno de los repositorios podrá utilizar los botones inferiores.

Pulse en **Actualizar** para refrescar completamente la caché y rellenar cualquier hueco. Para un repositorio grande esto puede llevar mucho tiempo, pero es útil si va a desconectarse y quiere tener la mejor caché disponible.

Pulse en **Exportar** para exportar la caché completa como un conjunto de archivos CSV. Esto puede ser útil si desea procesar los datos de registro utilizando un programa externo, aunque es útil principalmente para los desarrolladores.

Pulse en **Eliminar** para eliminar todos los datos de la caché de los repositorios seleccionados. Esto no deshabilita la caché para el repositorio, por lo que la próxima vez que pida datos de registro se creará una nueva caché.

### 4.31.7.2. Estadísticas de la Caché de Registro



**Figura 4.88.** El diálogo Configuración, página Estadísticas de la caché de registro

Pulse en el botón **Detalles** para ver estadísticas detalladas de una caché en concreto. Muchos de los campos mostrados aquí son de interés principalmente para los desarrolladores de TortoiseSVN, así que no se describen todos en detalle aquí.

#### RAM

La cantidad de memoria necesaria para dar servicio a esta caché.

#### Disco

El espacio de disco utilizado para la caché. Los datos están comprimidos, así que el uso de disco generalmente es bastante modesto.

#### Conexión

Muestra si el repositorio estaba disponible la última vez que se utilizó la caché.

#### Última actualización

La última fecha y hora en la que se modificó el contenido de la caché.

#### Última actualización head

La última vez que se pidió la revisión HEAD al servidor.

#### Autores

El número de diferentes autores con mensajes almacenados en la caché.

Rutas

El número de rutas mostradas, como las vería utilizando `svn log -v`.

Saltar rangos

El número de rangos de revisiones que no se han obtenido, simplemente por no han sido pedidos. Esta es una medida del número de huecos en la caché.

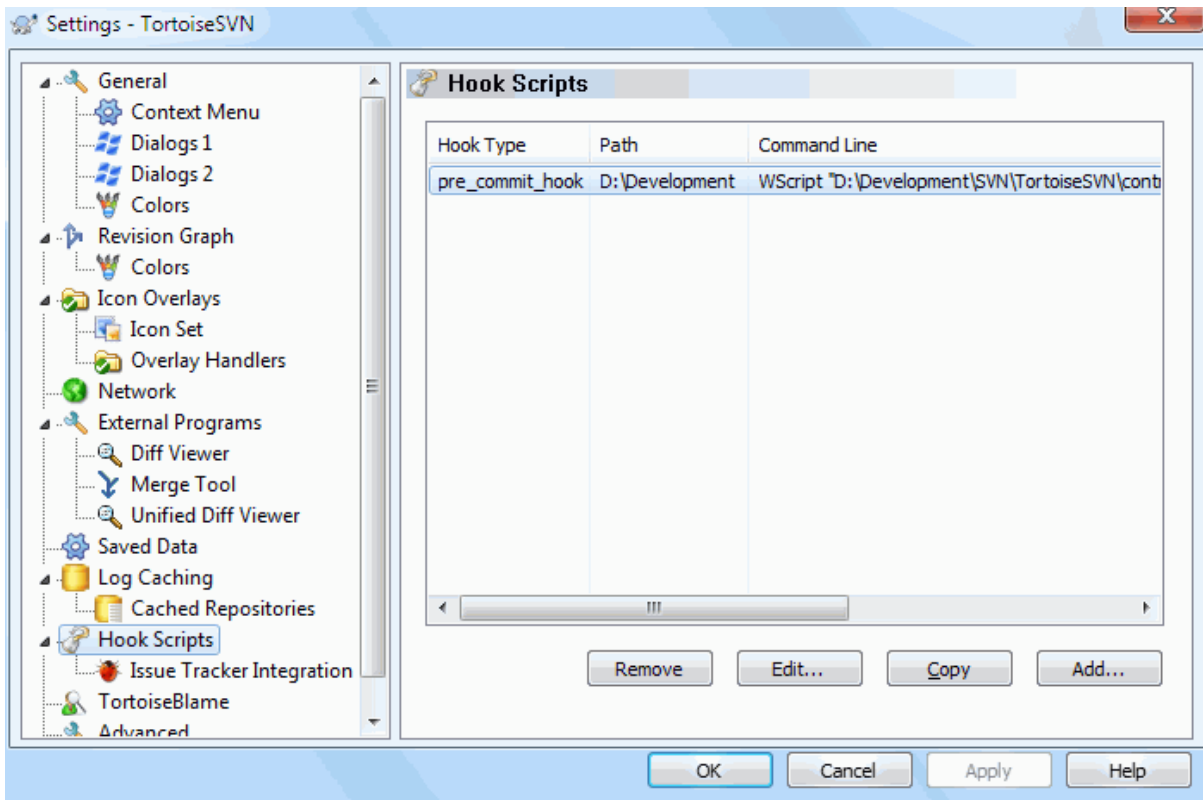
Revisión máx

El número de revisión más alto almacenado en la caché.

Cuenta de revisiones

El número de revisiones almacenados en la caché. Esta es otra medida de la completitud de la caché.

### 4.31.8. Scripts gancho del lado del cliente

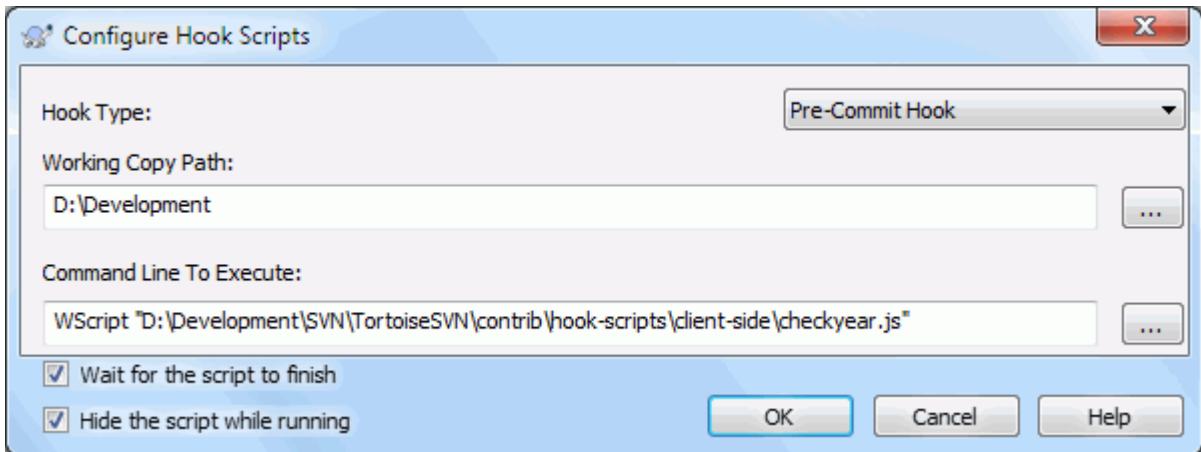


**Figura 4.89. El diálogo Configuración, página de scripts gancho**

Este diálogo le permite establecer scripts gancho que se ejecutarán automáticamente cuando se realicen ciertas acciones de Subversion. Al contrario que los scripts gancho explicados en [Sección 3.3, “Scripts gancho en el lado del servidor”](#), estos scripts se ejecutan localmente en el cliente.

Una aplicación para este tipo de ganchos podría ser una llamada a un programa como `SubWCRev.exe` para actualizar los números de versión tras una confirmación, y quizás para provocar una recompilación.

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



**Figura 4.90. El diálogo Configuración, configurar scripts gancho**

Para añadir un nuevo script gancho, simplemente pulse Añadir y rellene los detalles.

There are currently these types of hook script available

#### Inicio-confirmación

Llamado antes que se muestre el diálogo de confirmación. Puede querer utilizarlo si el gancho modifica un archivo versionado y afecta a la lista de archivos que necesitan ser confirmados y/o al mensaje de confirmación. Sin embargo, debería tener en cuenta que dado que el gancho se llama en una fase temprana, la lista completa de objetos seleccionados para confirmar no está disponible.

#### Manual Pre-commit

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

#### Check-commit

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

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

#### Pre-confirmación

Llamado después que el usuario hace click en el botón Aceptar en el diálogo de confirmación, y antes que comience el proceso de confirmación. Este gancho tiene una lista exacta de lo que se va a confirmar.

#### Post-confirmación

Called after the commit finishes successfully.

#### Inicio-actualización

Llamado antes que se muestre el diálogo actualizar-a-la-revisión.

#### Pre-actualización

Llamado antes que realmente comience la actualización o cambio de Subversion.

#### Post-actualización

Llamado después que termine la actualización, cambio u obtención (sea satisfactoria o no).

#### Pre-conexión

Llamado antes de un intento de contactar al repositorio. Invocado al menos una vez en cinco minutos.

#### Pre-lock

Called before an attempt to lock a file.

#### Post-lock

Called after a file has been locked.

Un gancho se define para una ruta de copia de trabajo en particular. Sólo necesita especificar la ruta de más alto nivel; si realiza una operación en una subcarpeta, TortoiseSVN automáticamente buscará hacia arriba una ruta que concuerde.

Luego, usted debe especificar la línea de comando a ejecutar, empezando con la ruta al script gancho o ejecutable. Este podría ser un archivo por lotes, un archivo ejecutable o cualquier otro archivo que tenga una asociación de fichero válida para Windows, por ejemplo, un script en perl. Debe saber que el script no debe de ser especificado usando una ruta UNC ya que la consola de Windows no permite a estos archivos ejecutarse debido a restricciones de seguridad.

La línea de comandos incluye diversos parámetros que se rellenarán por TortoiseSVN. Los parámetros que se pasan dependen de qué gancho se ejecuta. Cada gancho tiene sus propios parámetros que se pasan en el siguiente orden:

#### Inicio-confirmación

PATHMESSAGEFILECWD

#### Manual Pre-commit

PATHMESSAGEFILECWD

#### Check-commit

PATHMESSAGEFILECWD

#### Pre-confirmación

PATHDEPTHMESSAGEFILECWD

#### Post-confirmación

PATHDEPTHMESSAGEFILEREVISIONERRORCWD

#### Inicio-actualización

PATHCWD

#### Pre-actualización

PATHDEPTHREVISIONCWD

#### Post-actualización

PATHDEPTHREVISIONERRORCWDRESULTPATH

#### Pre-conexión

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

El significado de cada una de estos parámetros se describe aquí:

#### PATH

Una ruta a un archivo temporal que contiene todas las rutas para la operación que se ha iniciado. Cada ruta aparece en una línea distinta en el archivo temporal.

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

La profundidad a la que se realiza la confirmación o actualización.

Los valores posibles son:

-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

Ruta a un archivo que contiene el mensaje de registro de la confirmación. El archivo contiene el texto con la codificación UTF-8. Tras la ejecución con éxito del gancho start-commit, el mensaje de registro se vuelve a leer, dándole al gancho una oportunidad para modificarlo.

#### REVISION

La revisión del repositorio a la que se debe actualizar o después que se completa una confirmación.

#### LOCK

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

#### FORCE

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

#### ERROR

Ruta a un archivo que contiene el mensaje de error. Si no hubo error, el archivo estará vacío.

#### CWD

El directorio de trabajo actual desde el que se ejecuta el script. Este directorio se establece al directorio raíz común de todas las rutas afectadas.

#### 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.

Tenga en cuenta que, aunque hemos dado estos nombres de parámetros por conveniencia, no tiene que referirse a estos nombres en la configuración de los ganchos. Todos los parámetros listados para un gancho en concreto se pasan siempre, los quiera o no ;-)

Si quiere que la operación de Subversion espere a que el gancho termine, marque **Esperar a que el script termine**.

Normalmente querrá ocultar esas feas ventanas DOS cuando se ejecute el script, por lo que la casilla **Ocultar el script mientras se ejecuta** está marcada por defecto.

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>]. ([Sección 3, "Licencia"](#) explains how to access the repository.)

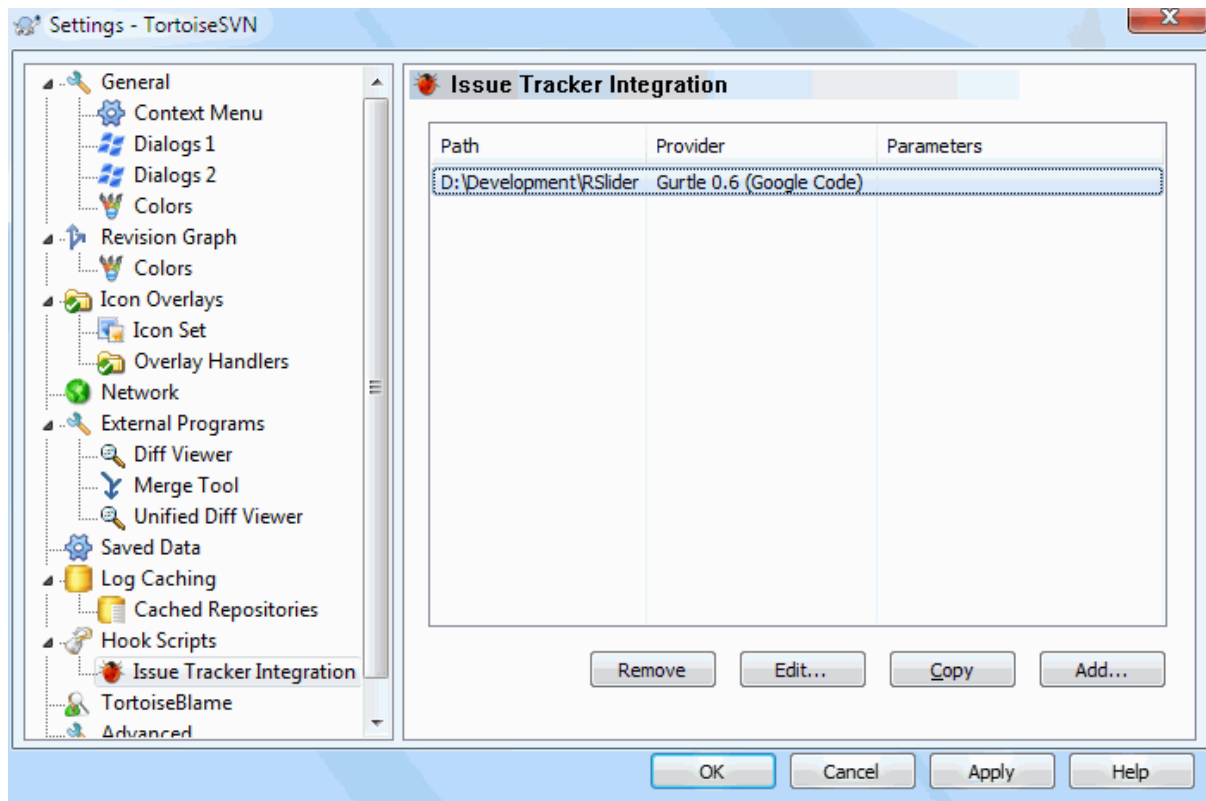
When debugging hook scripts you may want to echo progress lines to the DOS console, or insert a pause to stop the console window disappearing when the script completes. Because I/O is redirected this will not normally work. However you can redirect input and output explicitly to CON to overcome this.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

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

Una pequeña herramienta está incluida en la carpeta de instalación de TortoiseSVN llamada `ConnectVPN.exe`. Puede utilizar esta herramienta configurada como un nexo de pre-conexión para conectarse automáticamente a su VPN antes de que TortoiseSVN intente conectarse al repositorio. Simplemente pase el nombre de la conexión VPN como primer parámetro de la herramienta.

#### 4.31.8.1. Integración con el gestor de incidencias

TortoiseSVN puede utilizar un plugin COM para interrogar a los gestores de incidencias dentro del diálogo de confirmación. El uso de estos plugins se describe en [Sección 4.29.2, “Obteniendo información desde el gestor de incidencias”](#). Si el administrador de su sistema le ha proporcionado uno de estos plugins, y ya está instalado y registrado, este es el lugar para especificar cómo se integra con su copia de trabajo.



**Figura 4.91. El diálogo Configuración, página Integración con control de incidencias**

Pulse en `Añadir...` para utilizar el plugin en una copia de trabajo en concreto. Aquí puede especificar la ruta de la copia de trabajo, elegir qué plugin desea utilizar desde la lista desplegable con todos los plugins de gestores de incidencia registrados, y cualquier parámetro que necesite. Los parámetros son específicos de cada plugin, pero pueden incluir su nombre de usuario en el gestor de incidencias, para que el plugin pueda preguntar sólo aquellas incidencias que tenga asignadas.

Si desea que todos los usuarios utilicen el mismo plugin COM para su proyecto, puede especificar el plugin también con las propiedades `bugtraq:provideruuid`, `bugtraq:provideruuid64` y `bugtraq:providerparams`.

`bugtraq:provideruuid`

Esta propiedad especifica el COM UUID del `IBugtraqProvider`, por ejemplo `{91974081-2DC7-4FB1-B3BE-0DE1C8D6CE4E}`. (Este ejemplo es el UUID del *proveedor de seguimiento de errores Gurtle* [<http://code.google.com/p/gurtle/>], el cual es un proveedor para *Código de Google* [<http://code.google.com/hosting/>] seguimiento de incidencias.)

`bugtraq:provideruuid64`

Esta es como `bugtraq:provideruuid`, pero para la versión de 64 bits de `IBugtraqProvider`.

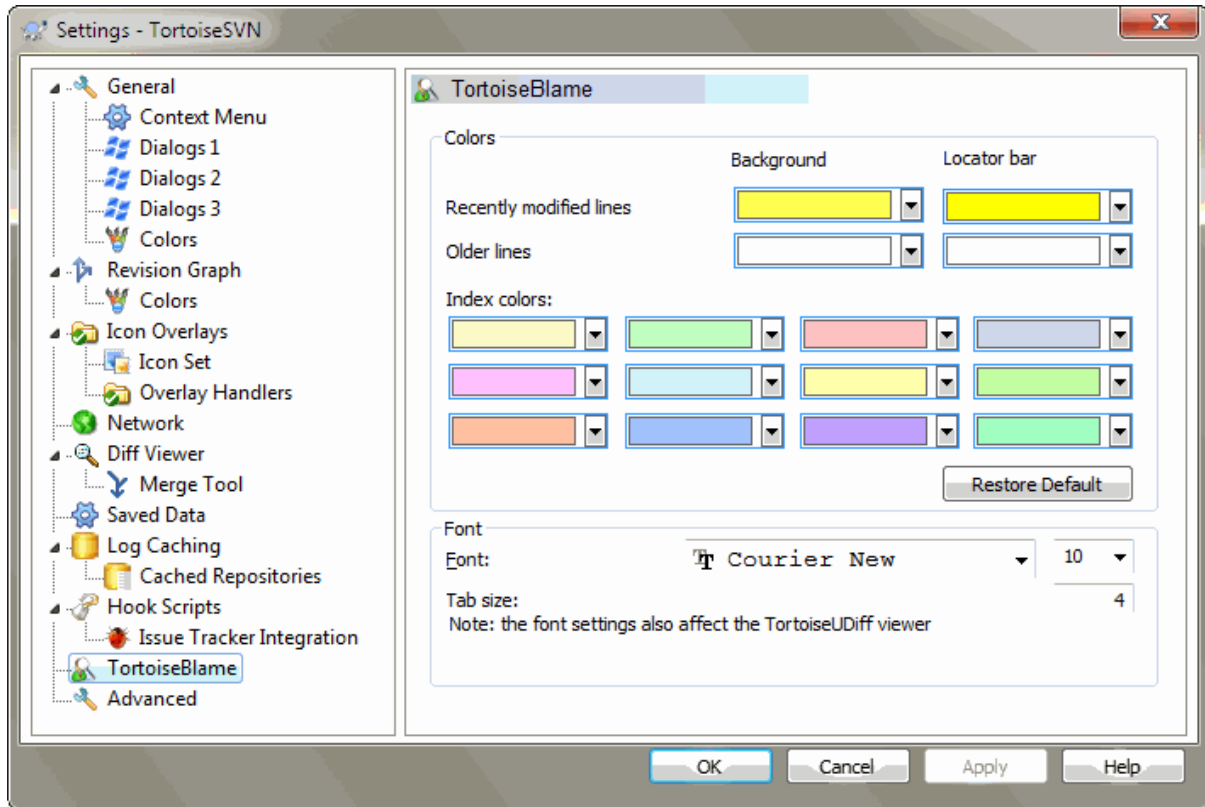


bugtraq:providerparams

Esta propiedad especifica los parámetros pasados al IBUGTraqProvider.

Por favor compruebe la documentación de su plugin IBUGTraqProvider para averiguar qué debe especificar en estas dos propiedades.

#### 4.31.9. Configuración de TortoiseBlame



**Figura 4.92. El diálogo Configuración, página TortoiseBlame**

Las configuraciones utilizadas por TortoiseBlame se controlan desde el menú contextual principal, no directamente con el propio TortoiseBlame.

##### Colores

TortoiseBlame puede utilizar el color de fondo para indicar la antigüedad de las líneas en un archivo. Proponga los colores de los extremos especificando los colores para la revisión más nueva y más antigua, y TortoiseBlame utiliza una interpolación lineal entre esos colores de acuerdo a la revisión del repositorio indicada en cada línea.

Puedes especificar los colores que quieres utilizar para la barra localizadora.

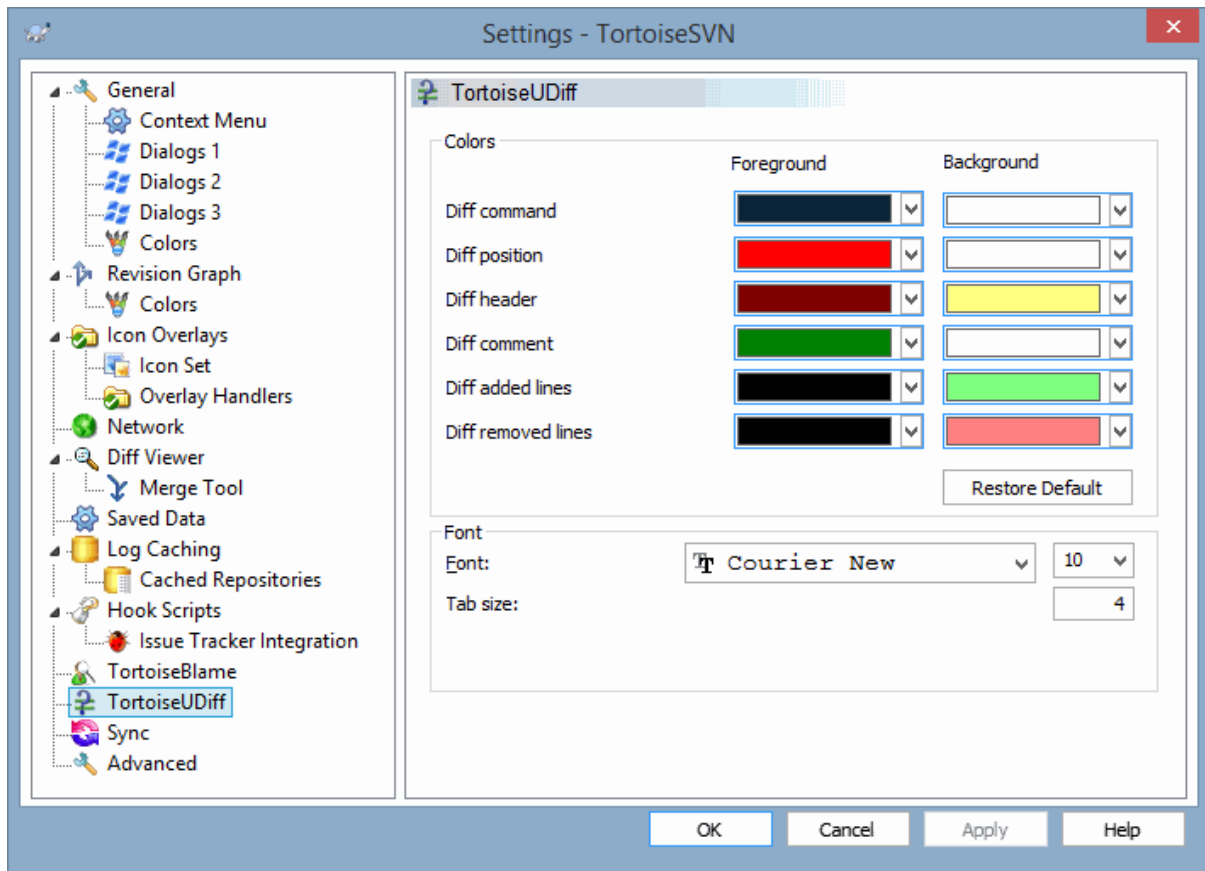
##### Fuente

Puede seleccionar la fuente utilizada para mostrar el texto, y el tamaño de punto a utilizar. Esto se aplica tanto al contenido del archivo como al autor y la información de la revisión mostrados en el panel izquierdo.

##### Tabulaciones

Define cuántos espacios se utilizarán para la expansión cuando se encuentre un carácter de tabulación en el contenido del archivo.

### 4.31.10. TortoiseUDiff Settings



**Figura 4.93. The Settings Dialog, TortoiseUDiff Page**

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

#### Colores

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

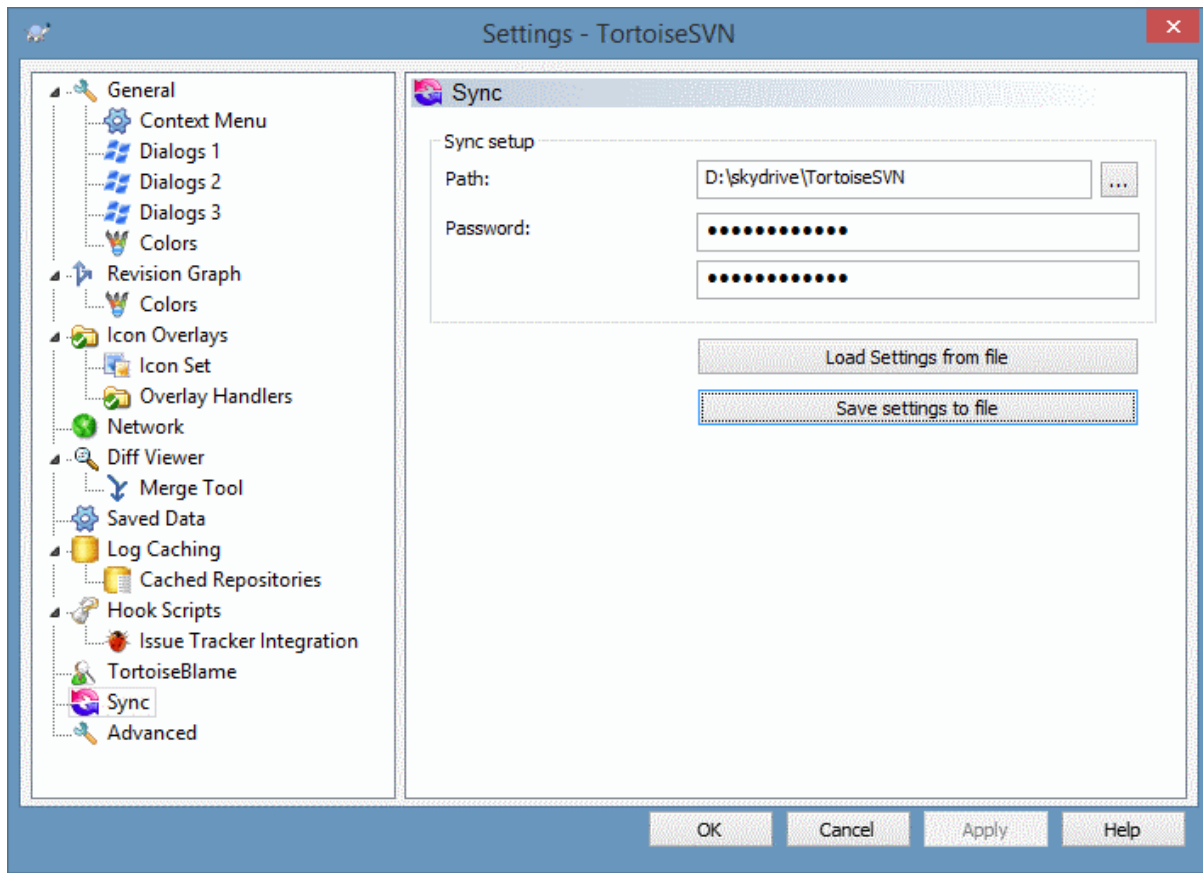
#### Fuente

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

#### Tabulaciones

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

### 4.31.11. Configuración de exportación TSVN



**Figura 4.94. 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. Configuración Avanzada

Sólo en la página avanzada del diálogo de configuración se pueden encontrar Unas opciones poco usadas. Estas opciones modifican el registro directamente y tiene que saber qué hace y para qué sirve cada una de estas opciones. No modifique estas opciones a no ser que esté seguro que necesita cambiarlas.

#### AllowAuthSave

A veces múltiples usuarios utilizan la misma cuenta en la misma computadora. En tales situaciones no se desea almacenar los datos de autenticación. Configurando este valor en `false` deshabilita el botón guardar autenticación del diálogo de autenticación.

#### AllowUnversionedObstruction

Si una actualización añade un nuevo archivo desde el repositorio que ya existe en la copia de trabajo como un archivo no versionado, la acción por defecto es la de mantener el archivo local, mostrándolo como una (posible) versión modificada del nuevo archivo del repositorio. Si prefiere que TortoiseSVN cree un conflicto en tales casos, establezca este valor a `false`.

#### AlwaysExtendedMenu

Al igual que con el explorador, TortoiseSVN muestra comandos adicionales si se mantiene pulsada la tecla **Mayúsculas** mientras se accede al menú contextual. Para forzar que estos comandos extendidos se muestren siempre, establezca 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

La lista de auto-completar mostrada en el editor de mensajes de confirmación muestra los nombres de los archivos que se proponen para confirmar. Para incluir también esos nombres sin sus extensiones, establezca esta 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

Si no desea que el explorador actualice los íconos de estado mientras otro comando TortoiseSVN se esté ejecutando (por ejemplo Actualizar, Confirmar, ...), establezca este valor a `true`.

#### CacheTrayIcon

Para añadir en la barra de tareas un ícono de la caché para el programa TSVNCache, establezca este valor a `true`. Esto realmente sólo es útil para desarrolladores dado que le permite terminar el programa de forma normal.

#### 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

Aquí puede especificar un lugar diferente para el archivo de configuración de Subversion. Esto afectará a todas las operaciones de TortoiseSVN.

#### CtrlEnter

En la mayoría de los diálogos de TortoiseSVN puede usar **Ctrl+Enter** para deshacerse del diálogo como si hubiera hecho click en el botón OK. Si no quiere esto, establezca este valor a `false`.

#### Debug

Ponga esto en `true` si quiere que un diálogo aparezca en una ventana emergente para cada comando mostrado en la línea de comandos utilizada para iniciar TortoiseProc.exe.

#### DebugOutputString

Set this to `true` if you want TortoiseSVN to print out debug messages during execution. The messages can be captured with special debugging tools only.

#### DialogTitles

The default format (value of 0) of dialog titles is `url/path - name of dialog - TortoiseSVN`. If you set this value to 1, the format changes to `name of dialog - url/path - TortoiseSVN`.

#### DiffBlamesWithTortoiseMerge

TortoiseSVN le permite asignar visores de diferencias externos. La mayoría de estos visores, sin embargo, no están preparados para ver la autoría de los cambios ([Sección 4.24.2, "Autoría de las diferencias"](#)), por lo que quizás desee volver a utilizar TortoiseMerge en este caso. Para hacerlo, establezca 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

Some apps change the case of filenames without notice but those changes aren't really necessary nor wanted. For example a change from `file.txt` to `FILE.TXT` wouldn't bother normal Windows applications, but Subversion is case sensitive in these situations. So TortoiseSVN automatically fixes such case changes.

If you don't want TortoiseSVN to automatically fix such case changes for you, you can set this value to `false`.

#### FullRowSelect

La lista de control de estado que es usada en varios diálogos (por ejemplo: confirmar, comprobar modificaciones, añadir, retroceder, ...) usa la selección de línea completa (p.ej: si selecciona una entrada, toda la línea es seleccionada y no sólo la primera columna). Esto está bien, pero entonces la línea seleccionada tapa la imagen de fondo en la parte inferior derecha, lo que puede resultar feo. Para desactivar la selección de la línea completa, establezca 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. El valor predeterminado es 0. Con este ajuste, los íconos se agrupan por tipo de aplicación. Todos los diálogos de TortoiseSVN se agrupan, todas las ventanas de TortoiseMerge se agrupan, ...



**Figura 4.95. Barra de tareas con agrupamiento por defecto**

2. Si se establece en 1, entonces en vez de todos los diálogos en un solo grupo por Application, están agrupados por repositorio. Por ejemplo, si usted tiene abierto un diálogo de registro y uno de confirmación para el repositorio A, y un cuadro de diálogo de verificación para la modificación y uno de registro para el repositorio B, entonces hay dos iconos de la aplicación agrupados que se muestran en la barra de tareas de Win7, uno para cada repositorio. Pero las ventanas TortoiseMerge no se agrupan junto con los diálogos de TortoiseSVN.



**Figura 4.96. Barra de tareas con agrupamiento por repositorio**

3. Si se establece en 2, el grupo funciona como si la configuración se establece en 1, excepto que las ventanas de TortoiseSVN, TortoiseMerge, TortoiseBlame, TortoiseIDiff y TortoiseUDiff están todas agrupadas. Por ejemplo, si usted tiene el cuadro de diálogo de confirmación abierto y luego hace doble clic sobre un archivo modificado, la ventana de diferencia de TortoiseMerge abierta será puesta en el mismo icono agrupado en la barra de tareas que el diálogo de confirmación.



**Figura 4.97. Barra de tareas con agrupamiento por repositorio**

4. If set to 3, then the grouping works as with the setting set to 1, but the grouping isn't done according to the repository but according to the working copy. This is useful if you have all your projects in the same repository but different working copies for each project.
5. If set to 4, then the grouping works as with the setting set to 2, but the grouping isn't done according to the repository but according to the working copy.

#### HideExternalInfo

If this is set to `false`, then every `svn:externals` is shown during an update separately.

If it is set to `true` (the default), then update information for externals is only shown if the externals are affected by the update, i.e. changed in some way. Otherwise nothing is shown as with normal files and folders.

#### GroupTaskbarIconsPerRepoOverlay

Esto no tiene efecto si la opción `GroupTaskbarIconsPerRepo` se establece en 0 (ver arriba).

If this option is set to `true`, then every icon on the Win7 taskbar shows a small colored rectangle overlay, indicating the repository the dialogs/windows are used for.



**Figura 4.98. Agrupamiento de barra de tareas con superposición de colores de repositorio**

#### IncludeExternals

Por defecto, TortoiseSVN siempre lanza una actualización con las referencias externas incluidas. Esto evita problemas con copias de trabajo inconsistentes. Si por el contrario tiene muchas referencias externas establecidas, una actualización puede llevar cierto tiempo en realizarse. Establezca este valor a `false` para lanzar la actualización por defecto excluyendo las referencias externas. Para actualizar con las referencias externas incluidas, puede lanzar el diálogo `Actualizar a revisión...` o puede establecer este valor a `true` de nuevo.

#### 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

Una cadena de formato para los mensajes de registro, cuando múltiples revisiones sean seleccionadas en el registro del diálogo.

Puedes utilizar los siguientes marcadores de posición en tu cadena de formato:

`%1!ld!`

es reemplazado con el texto del número de revisión

`%2!s!`

es reemplazado con el mensaje corto del registro de la revisión

#### LogStatusCheck

El diálogo de registro muestra la ruta de la revisión de la copia de trabajo en negrita. Pero esto requiere que el diálogo de registro obtenga el estado de esa ruta. Dado que esto puede tardar para copias de trabajo grandes, puede establecer este valor a `false` para desactivar esta característica.

#### 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

Cuando fusiona revisiones desde otra rama, y hay información de registro de fusión disponible, los mensajes de las revisiones que fusiona se reunirán para crear un mensaje de registro de confirmación. Se utilizará una cadena de texto predefinida para separar los mensajes de registro individuales de las revisiones fusionadas. Si lo desea, puede establecer este valor con una cadena de separación de su elección.

#### NumDiffWarning

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

#### OldVersionCheck

TortoiseSVN comprueba si hay una nueva version una vez por semana. Si se encuentra una actualización, el diálogo de confirmar muestra un control de enlace con esa información. Si prefiere usar el comportamiento antiguo de mostrar un diálogo con la notificación de la actualización, establezca este valor a `true`.

#### RepoBrowserTrySVNParentPath

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

#### ScintillaBidirectional

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

#### ScintillaDirect2D

This option enables the use of Direct2D accelerated drawing in the Scintilla control which is used as the edit box in e.g. the commit dialog, and also for the unified diff viewer. With some graphic cards however this sometimes doesn't work properly so that the cursor to enter text isn't always visible. If that happens, you can turn this feature off by setting this value to `false`.

#### OutOfDateRetry

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

0

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

1

This is the default. The user is asked whether to update the working copy or not, and the commit dialog is reopened after the update so the user can proceed with the commit right away.

2

Similar to 1, but instead of updating only the paths selected for a commit, the update is done on the working copy root. This helps to avoid inconsistent working copies.

3

The user is not asked to update the working copy. The commit simply fails with the out-of-date error message.

#### PlaySound

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

#### ShellMenuAccelerators

TortoiseSVN usa aceleradores para sus entradas del menú contextual del explorador. Esto puede provocar que algunos aceleradores aparezcan dos veces (p.ej. el `Confirmación SVN` tiene el acelerador **Alt-C**, pero también lo tiene la entrada `Copy` del explorador). Si no quiere o no necesita los aceleradores de las entradas de TortoiseSVN, establezca este valor a `false`.

#### ShowContextMenuIcons

Esto puede ser útil si utiliza un reemplazo del Explorador de Windows o si tiene problemas para mostrar el menú contextual de forma correcta. Establezca este valor a `false` si no desea que TortoiseSVN muestre íconos en el menú contextual del shell. Establezca este valor `true` para que se muestren los íconos.

#### ShowAppContextMenuIcons

Si no quiere que TortoiseSVN muestre íconos en los diálogos de los menús contextuales, establezca este valor a `false`.

#### StyleCommitMessages

El diálogo de confirmación y registro usa estilos (p.ej. negrita, cursiva) en los mensajes de confirmación (ver [Sección 4.4.5, “Mensajes de registro de confirmación”](#) para más detalles). Si no quiere que haga esto, establezca el valor a `false`.

#### UpdateCheckURL

Este valor contiene la URL que TortoiseSVN utiliza para intentar descargar un archivo de texto para averiguar si hay actualizaciones disponibles. Esto puede ser útil para administradores de empresas que no desean que sus usuarios actualicen TortoiseSVN hasta que ellos lo aprueben.

#### UseCustomWordBreak

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

#### VersionCheck

TortoiseSVN comprueba si hay una nueva versión una vez por semana. Si no quiere que TortoiseSVN realice esta comprobación, establezca este valor a `false`.

## 4.32. Último paso

### ¡Donar!

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.

The screenshot shows a dialog box titled "Edit Project to monitor". It has a close button in the top right corner. The dialog is divided into three sections: "Project", "Authentication", and "Misc". In the "Project" section, the "Name" field contains "TortoiseSVN" and the "Path or URL" field contains "D:\Development\SVN\TortoiseSVN". The "Authentication" section has a note: "leave username and password empty if the repository allows anonymous access, or if the authentication is stored already". Below this note are empty fields for "Username" and "Password". The "Misc" section has a field for "Monitor interval in minutes" with the value "5". At the bottom of the dialog are "OK" and "Cancel" buttons.

**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.



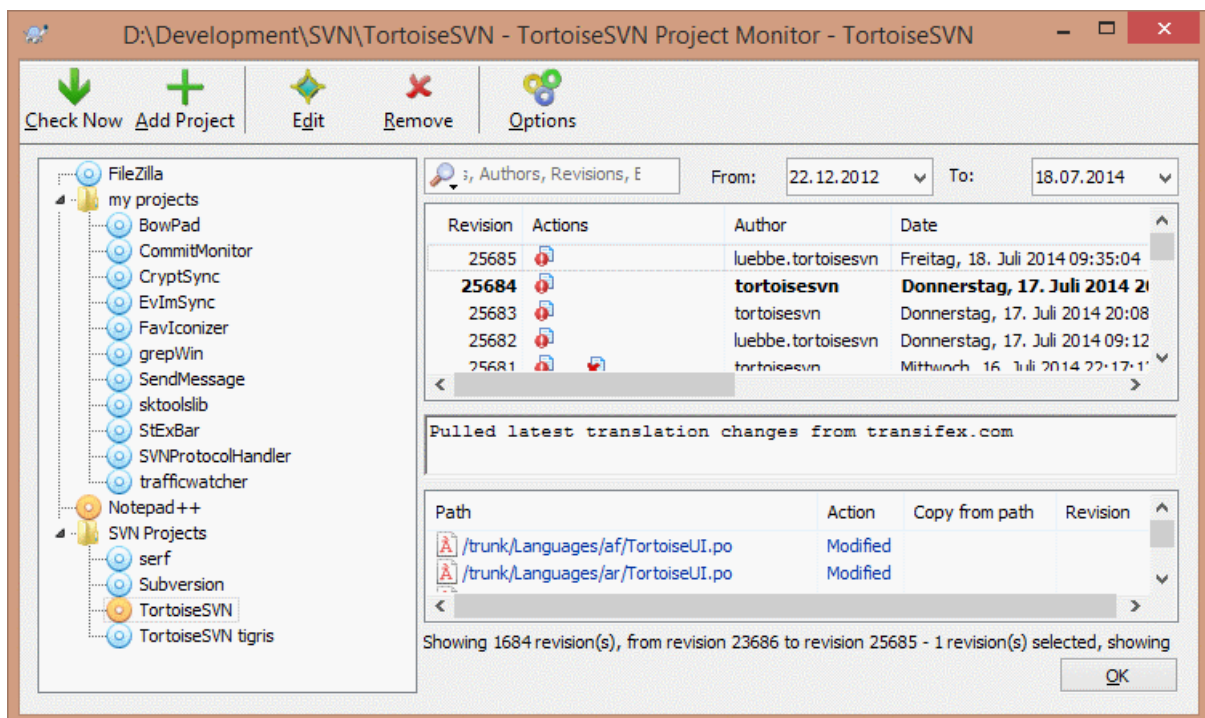
## check interval

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



**Figure 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.

**Eliminar**

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.

**Opciones**

Shows a dialog to configure the behavior of the project monitor.

---

# Capítulo 6. El programa SubWCRev

SubWCRev es un programa de consola para Windows que puede utilizarse para leer el estado de una copia de trabajo local y opcionalmente realizar sustituciones de palabras clave en un archivo plantilla. A menudo se utiliza como parte del proceso de compilación como una forma de incorporar información de la copia de trabajo en el objeto que está construyendo. Típicamente, se puede utilizar para incluir el número de revisión en un diálogo “Acerca de”.

## 6.1. La línea de comandos de SubWCRev

SubWCRev lee el estado de la Subversion de todos los ficheros de la copia de trabajo, excluyendo los externos por defecto. Guarda el número de la revisión enviada más alto encontrado, y la fecha del envío de esa revisión, también guarda si hay modificaciones locales en la copia de trabajo, o revisiones actualizadas y mezcladas. El número de revisión, el rango de la revisión de actualización y el estado de la modificación son mostrados en stdout.

SubWCRev.exe is called from the command line or a script, and is controlled using the command line parameters.

```
SubWCRev WorkingCopyPath [SrcVersionFile DstVersionFile] [-nmdfe]
```

WorkingCopyPath es la ruta a la copia de trabajo que se va a comprobar. Sólo puede utilizar SubWCRev en copias de trabajo, no directamente en el repositorio. La ruta puede ser absoluta o relativa al directorio actual.

Si quiere que SubWCRev realice sustituciones de palabras clave, para que se graben los campos como la revisión del repositorio y la URL a un archivo de texto, debe proporcionar un archivo plantilla SrcVersionFile y un archivo de salida DstVersionFile que contendrá la versión sustituida de la plantilla.

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.



## Sugerencia

To create a file with a starting dot in the Windows explorer, enter `.subwcrevignore..` Note the trailing dot.

Hay un número de selecciones opcionales que afectan la manera en que SubWCRev trabaja. Si usted usa más de una, deben especificarse como un único grupo, por ejemplo `-nm`, no `-n -m`.

Cambiar	Descripción
-n	Si se especifica esta opción, SubWCRev terminará con <code>ERRORLEVEL 7</code> si la copia de trabajo tiene modificaciones locales. Esto puede utilizarse para evitar compilaciones cuando hay presentes cambios sin confirmar.
-N	If this switch is given, SubWCRev will exit with <code>ERRORLEVEL 11</code> if the working copy contains unversioned items that are not ignored.
-m	Si se especifica esta opción, SubWCRev terminará con <code>ERRORLEVEL 8</code> si la copia de trabajo contiene revisiones mezcladas. Esto puede utilizarse para evitar compilaciones de una copia de trabajo parcialmente actualizada.
-d	Si se especifica esta opción, SubWCRev terminará con <code>ERRORLEVEL 9</code> si el archivo de destino ya existe.
-f	Si se especifica esta opción, SubWCRev incluirá la revisión del último cambio de las carpetas. El comportamiento por defecto es utilizar únicamente los archivos cuando se obtienen los números de revisión.
-e	Si se especifica esta opción, SubWCRev examinará los directorios que se han incluido con <code>svn:externals</code> , pero sólo si son del mismo repositorio. El comportamiento por defecto es ignorar los 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	Si se especifica esta opción, SubWCRev mostrará los números de revisión en HEX.
-X	Si se especifica esta opción, SubWCRev mostrará los números de revisión en HEX, anteponiendo '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.

**Tabla 6.1. Lista de opciones de línea de comandos disponible**

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:

Error Code	Descripción
1	Syntax error. One or more command line parameters are invalid.
2	The file or folder specified on the command line was not found.
3	The input file could not be opened, or the target file could not be created.
4	Could not allocate memory. This could happen if e.g. the source file is too big.
5	The source file can not be scanned properly.
6	SVN error: Subversion returned with an error when SubWCRev tried to find the information from the working copy.
7	The working copy has local modifications. This requires the <code>-n</code> switch.

Error Code	Descripción
8	The working copy has mixed revisions. This requires the <code>-m</code> switch.
9	The output file already exists. This requires the <code>-d</code> switch.
10	The specified path is not a working copy or part of one.
11	The working copy has unversioned files or folders in it. This requires the <code>-N</code> switch.

**Tabla 6.2. List of SubWCRev error codes**

## 6.2. Sustitución de palabras clave

Si se especifican archivos de origen y destino, SubWCRev copia el archivo origen al destino, realizando las siguientes sustituciones de palabras clave:

Palabra clave	Descripción
\$WCREV\$	Se reemplaza con la revisión de confirmación más alta de la copia de trabajo.
\$WCREV&\$	Replaced with the highest commit revision in the working copy, ANDed with the value after the <code>&amp;</code> char. For example: <code>\$WCREV&amp;0xFFFF\$</code>
\$WCREV-\$, \$WCREV+\$	Replaced with the highest commit revision in the working copy, with the value after the <code>+</code> or <code>-</code> char added or subtracted. For example: <code>\$WCREV-1000\$</code>
\$WCDATES\$, \$WCDATEUTC\$	Replaced with the commit date/time of the highest commit revision. By default, international format is used: <code>yyyy-mm-dd hh:mm:ss</code> . Alternatively, you can specify a custom format which will be used with <code>strftime()</code> , for example: <code>\$WCDATE=%a %b %d %I:%M:%S %p\$</code> . For a list of available formatting characters, look at the <a href="http://msdn.microsoft.com/en-us/library/fe06s4ak.aspx">referencia en línea</a> [http://msdn.microsoft.com/en-us/library/fe06s4ak.aspx].
\$WCNOW\$, \$WCNOWUTC\$	Reemplazado por la fecha/hora actual del sistema. Esto puede utilizarse para indicar la fecha de compilación. El formato del tiempo es el descrito por <code>\$WCDATE\$</code> .
\$WCRANGES\$	Sustituido por el rango de la revisión actualizada en la copia de trabajo. Si la copia de trabajo presenta un estado consistente, esta será una única revisión. Si la copia de trabajo contiene revisiones mezcladas, tanto si es debido a que están fuera de fecha, o a una deliberada actualización-a-revisión, entonces el rango será mostrado de la manera <code>100:200</code> .
\$WCMIXED\$	<code>\$WCMIXED?TextoSI:TextoNO\$</code> se reemplaza por <code>TextoSI</code> si hay revisiones de actualización mezcladas, o por <code>TextoNO</code> si no es así.
\$WCMODS\$	<code>\$WCMODS?TextoSI:TextoNO\$</code> se reemplaza por <code>TextoSI</code> si hay modificaciones locales, o por <code>TextoNO</code> si no es así.
\$WCUNVER\$	<code>\$WCUNVER?TText:FText\$</code> is replaced with <code>TextoSI</code> if there are unversioned items in the working copy, or <code>TextoNO</code> if not.
\$WCEXTALLFIXED\$	<code>\$WCEXTALLFIXED?TText:FText\$</code> is replaced with <code>TextoSI</code> if all externals are fixed to an explicit revision, or <code>TextoNO</code> if not.
\$WCISTAGGED\$	<code>\$WCISTAGGED?TText:FText\$</code> is replaced with <code>TextoSI</code> if the repository URL contains the tags classification pattern, or <code>TextoNO</code> if not.
\$WCURL\$	Se reemplaza con la URL del repositorio de la copia de trabajo pasada a SubWCRev.
\$WCINSVNS\$	<code>\$WCINSVN?TextoSI:TextoNO\$</code> se reemplaza por <code>TextoSI</code> si la entrada está versionada, o por <code>TextoNO</code> si no es así.
\$WCNEEDSLOCK\$	<code>\$WCNEEDSLOCK?TextoSI:TextoNO\$</code> se reemplaza por <code>TextoSI</code> si la entrada tiene establecida la propiedad <code>svn:needs-lock</code> , o por <code>TextoNO</code> si no es así.

Palabra clave	Descripción
\$WCISLOCKED\$	\$WCISLOCKED?TextoSI:TextoNO\$ se reemplaza por TextoSI si la entrada está bloqueada, o por TextoNO si no es así.
\$WCLOCKDATE\$, \$WCLOCKDATEUTC\$	Se reemplaza con la fecha del bloqueo. El formato del tiempo se puede utilizar como se describe en \$WCDATE\$.
\$WCLOCKOWNER\$	Reemplazado con el nombre del propietario del bloqueo.
\$WCLOCKCOMMENT\$	Reemplazado por el comentario del bloqueo.
\$WCUNVER\$	\$WCUNVER?TText:FText\$ is replaced with TextoSI if there are unversioned files or folders in the working copy, or TextoNO if not.

**Tabla 6.3. List of available keywords**

SubWCRev does not directly support nesting of expressions, so for example you cannot use an expression like:

```
#define SVN_REVISION      "$WCMIXED?$WCRANGE$: $WCREV$$"
```

But you can usually work around it by other means, for example:

```
#define SVN_RANGE        $WCRANGE$
#define SVN_REV          $WCREV$
#define SVN_REVISION     "$WCMIXED?SVN_RANGE:SVN_REV$"
```



### Sugerencia

Algunas de estas palabras clave se aplican a ficheros únicos en lugar de a una copia de trabajo completa, así que solo tiene sentido utilizar estas palabras clave cuando SubWCRev es invocado para escanear un solo fichero. Esto se aplica a \$WCINSVN\$, \$WCNEEDSLOCK\$, \$WCISLOCKED\$, \$WCLOCKDATE\$, \$WCLOCKOWNER\$ and \$WCLOCKCOMMENT\$.

## 6.3. Ejemplo de palabras clave

El siguiente ejemplo le muestra cómo se sustituyen las palabras clave de un archivo plantilla en el archivo resultado.

```
// Test file for SubWCRev

char *Revision          = "$WCREV$";
char *Revision16        = "$WCREV&0xFF$";
char *Revisionp100      = "$WCREV+100$";
char *Revisionm100      = "$WCREV-100$";
char *Modified          = "$WCMODS?Modified:Not modified$";
char *Unversioned       = "$WCUNVER?Unversioned items found:no unversioned items$";
char *Date              = "$WCDATE$";
char *CustDate          = "$WCDATE=%a, %d %B %Y$";
char *DateUTC           = "$WCDATEUTC$";
char *CustDateUTC       = "$WCDATEUTC=%a, %d %B %Y$";
char *TimeNow           = "$WCNOW$";
char *TimeNowUTC        = "$WCNOWUTC$";
char *RevRange          = "$WCRANGE$";
char *Mixed              = "$WCMIXED?Mixed revision WC:Not mixed$";
```

```

char *ExtAllFixed    = "$WCEXTALLFIXED?All externals fixed:Not all externals fixed$";
char *IsTagged      = "$WCISTAGGED?Tagged:Not tagged$";
char *URL           = "$WCURL$";
char *isInSVN      = "$WCINSVN?versioned:not versioned$";
char *needslock    = "$WCNEEDSLOCK?TRUE:FALSE$";
char *islocked     = "$WCISLOCKED?locked:not locked$";
char *lockdateutc  = "$WCLOCKDATEUTC$";
char *lockdate     = "$WCLOCKDATE$";
char *lockcustutc  = "$WCLOCKDATEUTC=%a, %d %B %Y$";
char *lockcust     = "$WCLOCKDATE=%a, %d %B %Y$";
char *lockown      = "$WCLOCKOWNER$";
char *lockcmt      = "$WCLOCKCOMMENT$";

```

```

#if $WCMODS?1:0$
#error Source is modified
#endif

```

```
// End of file
```

Después de ejecutar SubWCRev.exe ruta\a\la\copia\de\trabajo fichprueba.tmpl  
fichprueba.txt, el archivo de salida fichprueba.txt se parecería a esto:

```
// Test file for SubWCRev
```

```

char *Revision      = "22837";
char *Revision16   = "53";
char *Revisionpl100 = "22937";
char *Revisionml100 = "22737";
char *Modified     = "Modified";
char *Unversioned  = "no unversioned items";
char *Date         = "2012/04/26 18:47:57";
char *CustDate     = "Thu, 26 April 2012";
char *DateUTC      = "2012/04/26 16:47:57";
char *CustDateUTC  = "Thu, 26 April 2012";
char *TimeNow      = "2012/04/26 20:51:17";
char *TimeNowUTC   = "2012/04/26 18:51:17";
char *RevRange     = "22836:22837";
char *Mixed        = "Mixed revision WC";
char *ExtAllFixed  = "All externals fixed";
char *IsTagged     = "Not tagged";
char *URL          = "https://svn.code.sf.net/p/tortoisesvn/code/trunk";
char *isInSVN     = "versioned";
char *needslock   = "FALSE";
char *islocked    = "not locked";
char *lockdateutc = "1970/01/01 00:00:00";
char *lockdate    = "1970/01/01 01:00:00";
char *lockcustutc = "Thu, 01 January 1970";
char *lockcust    = "Thu, 01 January 1970";
char *lockown     = "";
char *lockcmt    = "";

```

```

#if 1
#error Source is modified
#endif

```

```
// End of file
```





## Sugerencia

Un archivo como este se incluirá en la compilación por lo que puede esperar que esté versionado. Asegúrese que versiona el archivo de plantilla, no el archivo generado, porque si no cada vez que regenere el archivo de versión deberá confirmar el cambio, lo que a su vez significaría que el archivo de versión necesita ser actualizado.

## 6.4. Interfaz COM

Si necesita acceder a la información de las revisiones de Subversion desde otros programas, puede utilizar el interfaz COM de SubWCRev. El objeto a crear es `SubWCRev.object`, y se soportan los siguientes métodos:

Método	Descripción
<code>.GetWCInfo</code>	Este método atraviesa la copia de trabajo obteniendo la información de revisión. Naturalmente debe llamarlo antes que pueda acceder a la información utilizando los métodos restantes. El primer parámetro es la ruta. El segundo parámetro debe ser true si desea incluir las revisiones de las carpetas; equivalente al parámetro de la línea de comandos <code>-f</code> . El tercer parámetro debería ser true si desea incluir <code>svn:externals</code> ; equivalente al parámetro de la línea de comandos <code>-e</code> .
<code>.GetWCInfo2</code>	The same as <code>GetWCInfo()</code> but with a fourth parameter that sets the equivalent to the <code>-E</code> command line switch.
<code>.Revision</code>	La revisión de confirmación más alta de la copia de trabajo. Equivalente a <code>\$WCREV\$</code> .
<code>.Date</code>	La fecha/hora de la revisión de confirmación más alta. Equivalente a <code>\$WCDATE\$</code> .
<code>.Author</code>	El autor de la revisión de confirmación más alta, esto es, la última persona que confirmó cambios en la copia de trabajo.
<code>.MinRev</code>	La revisión mínima de actualización, como se muestra en <code>\$WCRANGE\$</code>
<code>.MaxRev</code>	La revisión máxima de actualización, como se muestra en <code>\$WCRANGE\$</code>
<code>.HasModifications</code>	True si hay modificaciones locales
<code>.HasUnversioned</code>	True if there are unversioned items
<code>.Url</code>	Reemplaza con la URL del repositorio de la copia de trabajo utilizados en <code>GetWCInfo</code> . Equivalente a <code>\$WCURL\$</code> .
<code>.IsSvnItem</code>	True si el ítem está versionado.
<code>.NeedsLocking</code>	True si el ítem tiene la propiedad <code>svn:needs-lock</code> establecida.
<code>.IsLocked</code>	True si el ítem está bloqueado.
<code>.LockCreationDate</code>	Cadena representando la fecha cuando el bloqueo se creó, o una cadena vacía si el ítem no está bloqueado.
<code>.LockOwner</code>	Cadena representando el propietario del bloqueo, o una cadena vacía si el ítem no está bloqueado.
<code>.LockComment</code>	El mensaje que se introdujo cuando se creó el bloqueo.

**Tabla 6.4. Métodos de automatización/COM soportados**

El siguiente ejemplo muestra cómo se podría usar el interfaz.

```
// testCOM.js - javascript file
// test script for the SubWCRev COM/Automation-object
```

```
filesystem = new ActiveXObject("Scripting.FileSystemObject");

revObject1 = new ActiveXObject("SubWCRev.object");
revObject2 = new ActiveXObject("SubWCRev.object");
revObject3 = new ActiveXObject("SubWCRev.object");
revObject4 = new ActiveXObject("SubWCRev.object");

revObject1.GetWCInfo(
    filesystem.GetAbsolutePathName("."), 1, 1);
revObject2.GetWCInfo(
    filesystem.GetAbsolutePathName(".."), 1, 1);
revObject3.GetWCInfo(
    filesystem.GetAbsolutePathName("SubWCRev.cpp"), 1, 1);
revObject4.GetWCInfo2(
    filesystem.GetAbsolutePathName("../.."), 1, 1, 1);

wcInfoString1 = "Revision = " + revObject1.Revision +
    "\nMin Revision = " + revObject1.MinRev +
    "\nMax Revision = " + revObject1.MaxRev +
    "\nDate = " + revObject1.Date +
    "\nURL = " + revObject1.Url + "\nAuthor = " +
    revObject1.Author + "\nHasMods = " +
    revObject1.HasModifications + "\nIsSvnItem = " +
    revObject1.IsSvnItem + "\nNeedsLocking = " +
    revObject1.NeedsLocking + "\nIsLocked = " +
    revObject1.IsLocked + "\nLockCreationDate = " +
    revObject1.LockCreationDate + "\nLockOwner = " +
    revObject1.LockOwner + "\nLockComment = " +
    revObject1.LockComment;

wcInfoString2 = "Revision = " + revObject2.Revision +
    "\nMin Revision = " + revObject2.MinRev +
    "\nMax Revision = " + revObject2.MaxRev +
    "\nDate = " + revObject2.Date +
    "\nURL = " + revObject2.Url + "\nAuthor = " +
    revObject2.Author + "\nHasMods = " +
    revObject2.HasModifications + "\nIsSvnItem = " +
    revObject2.IsSvnItem + "\nNeedsLocking = " +
    revObject2.NeedsLocking + "\nIsLocked = " +
    revObject2.IsLocked + "\nLockCreationDate = " +
    revObject2.LockCreationDate + "\nLockOwner = " +
    revObject2.LockOwner + "\nLockComment = " +
    revObject2.LockComment;

wcInfoString3 = "Revision = " + revObject3.Revision +
    "\nMin Revision = " + revObject3.MinRev +
    "\nMax Revision = " + revObject3.MaxRev +
    "\nDate = " + revObject3.Date +
    "\nURL = " + revObject3.Url + "\nAuthor = " +
    revObject3.Author + "\nHasMods = " +
    revObject3.HasModifications + "\nIsSvnItem = " +
    revObject3.IsSvnItem + "\nNeedsLocking = " +
    revObject3.NeedsLocking + "\nIsLocked = " +
    revObject3.IsLocked + "\nLockCreationDate = " +
    revObject3.LockCreationDate + "\nLockOwner = " +
    revObject3.LockOwner + "\nLockComment = " +
    revObject3.LockComment;

wcInfoString4 = "Revision = " + revObject4.Revision +
    "\nMin Revision = " + revObject4.MinRev +
    "\nMax Revision = " + revObject4.MaxRev +
```

```
"\nDate = " + revObject4.Date +
"\nURL = " + revObject4.Url + "\nAuthor = " +
revObject4.Author + "\nHasMods = " +
revObject4.HasModifications + "\nIsSvnItem = " +
revObject4.IsSvnItem + "\nNeedsLocking = " +
revObject4.NeedsLocking + "\nIsLocked = " +
revObject4.IsLocked + "\nLockCreationDate = " +
revObject4.LockCreationDate + "\nLockOwner = " +
revObject4.LockOwner + "\nLockComment = " +
revObject4.LockComment;
```

```
WScript.Echo(wcInfoString1);
WScript.Echo(wcInfoString2);
WScript.Echo(wcInfoString3);
WScript.Echo(wcInfoString4);
```

El siguiente listado es un ejemplo de como usar el objeto COM SubWCRev desde C#:

```
using LibSubWCRev;
SubWCRev sub = new SubWCRev();
sub.GetWCInfo("C:\\PathToMyFile\\MyFile.cc", true, true);
if (sub.IsSvnItem == true)
{
    MessageBox.Show("versioned");
}
else
{
    MessageBox.Show("not versioned");
}
```

---

# Capítulo 7. Interfaz IBugtraqProvider

Para lograr una integración más compacta con herramientas de seguimiento de problemas, que la lograda simplemente usando las propiedades `bugtraq:`, TortoiseSVN puede usar extensiones COM. Con estas extensiones es posible acceder a información directamente en el seguidor de problemas, interactuar con el usuario y proveer información a TortoiseSVN acerca de problemas no resueltos, verificar mensajes de registro introducidos por el usuario y hasta ejecutar acciones luego de una confirmación exitosa para, por ejemplo, cerrar un problema.

No podemos proporcionar información y tutoriales acerca de cómo implementar un objeto COM en tu lenguaje de programación favorito, pero tenemos ejemplos de plugins en C++/ALT y C# en nuestro repositorio en la carpeta `contrib/issue-tracker-plugins`. En esa carpeta también puede encontrar los ficheros de include necesarios para construir tu plugin. (Sección 3, "Licencia" explica cómo acceder al repositorio.)



## Importante

Se debería proveer una versión de 32-bit y una de 64-bit de su extensión. Esto es debido a que la versión de 64-bit de TortoiseSVN no puede usar una extensión de 32-bit y vice-versa.

## 7.1. Convenciones de nombres

Si usted libera una extensión del gestor de incidencias para TortoiseSVN, por favor *no* lo nombre *Tortoise*<XXXXX>. Nos gustaría reservar el prefijo *Tortoise* para el cliente de control de versión integrado a windows. Por ejemplo: TortoiseCVS, TortoiseSVN, TortoiseHg, TortoiseGit y TortoiseBzr son todos clientes de control de versión.

Por favor, nombre su extensión para un cliente Tortoise *Turtle*<XXXXX>, donde <XXXXX> se refiere al gestor de incidencias al que se está usted conectando. Como alternativa puede elegir un nombre que suene como *Turtle* pero que comience por otra letra. Buenos ejemplos son:

- Gurtle - Un plugin de seguimiento de incidencias para el código de Google
- TurtleMine - Un plugin de seguimiento de incidencias para Redmine
- VurtleOne - Un plugin de seguimiento de incidencias de VersionOne

## 7.2. La interfaz de IBugtraqProvider

TortoiseSVN 1.5 y posteriores puede usar extensiones que implementan la interface IBugtraqPorvider. La interface provee algunos métodos que las extensiones pueden usar para interactuar con la herramienta de seguimiento.

```
HRESULT ValidateParameters (  
    // Ventana primaria para cualquier interfaz de usuario // que se debe mostrar durante  
    [in] HWND hParentWnd,  
  
    // La cadena de caracteres que necesita ser validada.  
    [in] BSTR parameters,  
  
    // ¿Es valida la cadena de caracteres?  
    [out, retval] VARIANT_BOOL *valid  
);
```

Este método es llamado desde el diálogo de configuración en el cual el usuario puede agregar y configurar la extensión. Los parámetros "cadena de caracteres" pueden ser utilizados por la extensión para obtener información adicional requerida, ej., la URL al seguimiento de tareas, información de identificación, etc. La extensión debería verificar los parámetros "cadena de caracteres" y mostrar un diálogo de error si la cadena de caracteres no es válida. El parámetro `hParentWnd` debería ser utilizado para cualquier diálogo que la extensión muestre como si fuese la ventana principal. La extensión debe retornar `TRUE` si la validación de los parámetros

"cadena de caracteres" es exitosa. Si la extensión retorna FALSE, el diálogo de configuración no le permitirá al usuario agregar la extensión a una ruta de una copia de trabajo.

```
HRESULT GetLinkText (  
    // Ventana principal para cualquier (error) interfaz de usuario que necesite ser mostrada.  
    [in] HWND hParentWnd,  
  
    // Una cadena de caracteres, por si acaso necesitas conversar con tu  
    // servicio web (ej.) para saber cual es el texto correcto.  
    [in] BSTR parameters,  
  
    // ¿Que texto deseas mostrar?  
    // Use el hilo actual localmente.  
    [out, retval] BSTR *linkText  
);
```

La extensión puede proveer un string aquí que sera utilizado en el diálogo de actualización del TortoiseSVN para el botón que invoca a la extensión, ej., "Elija tema" o "Seleccione tiquet". Asegúrese que el string no sea demasiado largo, de lo contrario podría no caber en el botón. Si el método devuelve un error (ej., E\_NOTIMPL), un texto por defecto será utilizado para el botón.

```
HRESULT GetCommitMessage (  
    // Ventana principal de su interfaz proveedor.  
    [in] HWND hParentWnd,  
  
    // Parámetros para su proveedor.  
    [in] BSTR parameters,  
    [in] BSTR commonRoot,  
    [in] SAFEARRAY(BSTR) pathList,  
  
    // El texto ya presente en el mensaje de actualización.  
    // Su proveedor debería incluir este texto en el nuevo mensaje,  
    // donde sea apropiado.  
    [in] BSTR originalMessage,  
  
    // El nuevo texto para el mensaje de actualización.  
    // Reemplaza el mensaje original.  
    [out, retval] BSTR *newMessage  
);
```

Este es el método principal de la extensión. Este método es llamado desde el diálogo de actualización del TortoiseSVN cuando el usuario pulsa el botón de la extensión.

El string `parameters` es el string que el usuario debe ingresar al diálogo de configuración al configurar la extensión. Normalmente la extensión utilizaría esto para encontrar la URL del gestor de incidencias y/o información de identificación o más.

El string `RaízComún` contiene la ruta padre de todos los ítems seleccionados para desplegar el diálogo de confirmación. Note que esta *no* es la ruta raíz de todos los ítems que el usuario seleccionó en el diálogo de confirmación. Para el diálogo rama/etiqueta, éste es el directorio que ha de ser copiado.

El parámetro `pathList` contiene un arreglo de rutas (en strings) que el usuario ha seleccionado para confirmar.

El parámetro `originalMessage` contiene el texto introducido en la caja de mensaje de registro en el diálogo de confirmación. Si no se ha introducido todavía ningún texto, el string estará vacío.

El string retornado `newMessage` es copiado en la caja de edición del mensaje de registro en el diálogo de confirmación, reemplazando lo que sea que se encuentre allí. Si alguna extensión no modifica el string

originalMessage, debe retornar nuevamente el mismo string aquí, de otra manera cualquier texto introducido por el usuario se perderá.

## 7.3. La interfaz IBugtraqProvider2

En TortoiseSVN 1.6 se agrego una nueva interfaz de software que provee más funcionalidades para extensiones. Esta interfaz IBugtraqProvider2 hereda de IBugtraqProvider.

```
HRESULT GetCommitMessage2 (
    // Ventana padre de su proveedor de Interfaz.
    [in] HWND hParentWnd,

    // Parametros para su proveedor.
    [in] BSTR parameters,
    // La URL comun para la confirmación
    [in] BSTR commonURL,
    [in] BSTR commonRoot,
    [in] SAFEARRAY(BSTR) pathList,

    // El texto ya presente en el mensaje de confirmación.
    // Su proveedor debería incluir este texto en el mensaje nuevo,
    // donde corresponda.
    [in] BSTR originalMessage,

    // Usted le puede asignar propiedades personalizadas a una confirmación
    // estableciendo los siguientes dos parametros.
    // nota: Ambos arreglos deben ser del mismo largo.
    ; //      Para cada nombre de propiedad debe haber un valor de propiedad.

    // El contenido del campo bugID (si se muestra)
    [in] BSTR bugID,

    // El Contenido modificado del campo bugID.
    [out] BSTR * bugIDOut,

    // La lista de nombres de propiedades de revision.
    [out] SAFEARRAY(BSTR) * revPropNames,

    // La lista de valores de propiedades de revision.
    [out] SAFEARRAY(BSTR) * revPropValues,

    // Texto nuevo para el mensaje de confirmación.
    // Reempaza al mensaje original.
    [out, retval] BSTR * newMessage
);
```

Este método es llamado desde el diálogo de confirmación de TortoiseSVN cuando el usuario presiona el boton de la extensión. Este método es llamado en lugar de `GetCommitMessage()`. Por favor refierase a la documentación de `GetCommitMessage()` para los parámetros que también se usan allí.

El parámetro `commonURL` es la URL padre de todos los elementos seleccionados para mostrar en el diálogo de confirmación. Esta es básicamente la URL de la ruta `commonURL`.

El parámetro `bugID` contiene el contenido del campo bug-ID (si es mostrado, configurable con la propiedad `bugtraq:message`).

El parámetro de retorno `bugIDOut` es utilizado para completar el campo bug-ID cuando el método retorna.

Los parámetros de retorno `revPropNames` y `revPropValues` pueden contener pares nombre/valor para propiedades de revisión que la confirmación debería establecer. Una extensión debe asegurarse que ámbos arreglos tengan el mismo tamaño de retorno. Cada propiedad nombre en `revPropNames` también tiene un valor correspondiente en `revPropValues`. Si no hay propiedades de revisión para establecer, la extensión debe retornar arreglos vacíos.

```
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 es llamado justo antes que el diálogo de confirmación se cierre y la confirmación comience. Una extensión puede utilizar este método para validar los archivos/carpetas seleccionadas para la confirmación y/o el mensaje de confirmación ingresado por el usuario. Los parámetros son los mismos que para `GetCommitMessage2()`, con la diferencia que `commonURL` es ahora la URL común de todos los elementos *seleccionados*, y `commonRoot` la ruta raíz de todo los elementos seleccionados.

Para el diálogo rama/etiqueta, el `commonURL` es la URL fuente de la copia, y `commonRoot` es la URL de destino de la copia.

El parámetro de retorno `errorMessage` debe contener o bien un mensaje de error el cual el TortoiseSVN muestra al usuario o estar vacío para que inicie la confirmación. Si retorna un mensaje de error, TortoiseSVN muestra este string de error en un diálogo y mantiene el diálogo de confirmación abierto para que el usuario pueda corregir lo que sea que esta mal. Una extensión debería por lo tanto retornar un string de error que informe al usuario *que* esta mal y como corregirlo.

```
HRESULT OnCommitFinished (  
    // Ventana padre para cualquier (error) interfaz que necesite  
    // ser mostrada.  
    [in] HWND hParentWnd,  
  
    // La raiz comun de todas las rutas que se han confirmado.  
    [in] BSTR commonRoot,  
  
    // Todas las rutas que han sido confirmadas.  
    [in] SAFEARRAY(BSTR) pathList,  
  
    // El texto ya presente en el mensaje de confirmación.  
    [in] BSTR logMessage,  
  
    // La revisión de la confirmación.  
    [in] ULONG revision,  
  
    // Un error para mostrarle al usuario si esta función  
    // retorna otra cosa que S_OK.  
    [out, retval] BSTR * error  
);
```

Este método es llamado después de una confirmación exitosa. Una extensión puede utilizar este método para por ejemplo, cerrar la incidencia seleccionada o agregar información sobre la confirmación a la incidencia. Los parámetros son los mismos que para `GetCommitMessage2`.

```
HRESULT HasOptions(  
    // Si el proveedor provee opciones o no  
    [out, retval] VARIANT_BOOL *ret  
);
```

Este método es llamado desde el diálogo de configuración donde el usuario puede configurar la extensión. Si una extensión provee su propio diálogo de configuración con `ShowOptionsDialog`, debe retornar `TRUE` aquí, o `FALSE` de lo contrario.

```
HRESULT ShowOptionsDialog(  
    // Ventana padre para el diálogo de opciones.  
    [in] HWND hParentWnd,  
  
    // Parámetros para su proveedor.  
    [in] BSTR parameters,  
  
    // Los parámetros string.  
    [out, retval] BSTR * newparameters  
);
```

Este método es llamado desde el diálogo de configuración cuando el usuario oprime el botón "Opciones" que se muestra si `HasOptions` retorna `TRUE`. Una extensión puede mostrar un diálogo de opciones para facilitarle al usuario la configuración de la extensión.

Los parámetros string contienen los parámetros string de la extensión que ya están establecidos/ingresados.

El parámetro de retorno `newparameters` debe contener los parámetros string que la extensión construyó de la información que recopiló en su diálogo de opciones. Esos parámetro string es pasado a todos los otros métodos `IBugtraqProvider` y `IBugtraqProvider2`.



---

# Apéndice A. Preguntas más frecuentes (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.

Si tiene una pregunta que no está respondida en ningún otro lado, el mejor lugar para preguntar es en alguna de las siguientes listas de correo:

- <users@tortoisesvn.tigris.org> para preguntas relacionadas con el uso de TortoiseSVN.
- Si desea ayudar con el desarrollo de TortoiseSVN, entonces debería tomar parte en las discusiones en <dev@tortoisesvn.tigris.org>.
- Si desea ayudar con las traducciones de la documentación de la interfaz de usuario de TortoiseSVN, envíe un mail a <translators@tortoisesvn.tigris.org>.

---

# Apéndice B. ¿Cómo...?

Este apéndice contiene soluciones a los problemas o preguntas que puede tener mientras utiliza TortoiseSVN.

## B.1. Mover/copiar muchos archivos de golpe

Se pueden mover/copiar archivos de uno en uno utilizando TortoiseSVN → Renombrar.... Pero si desea mover/copiar muchos archivos, esta forma es muy lenta y trabajosa.

La manera recomendada es a través de right dragging los archivos a la nueva ubicación. Simplemente right click en los archivos que desea mover o copiar sin soltar el botón del mouse. Luego arrastra los archivos a su nueva ubicación y suelta el botón del mouse. Un menu contextual aparecerá donde podrás elegir Context Menu → SVN Copy versioned files here. o Context Menu → SVN Move versioned files here.

## B.2. Obligar a los usuarios a introducir un mensaje de registro

Hay dos formas de evitar que los usuarios puedan confirmar con un mensaje de registro vacío. Uno es específico de TortoiseSVN, el otro funciona con todos los clientes de Subversion, pero necesita acceso directo al servidor.

### B.2.1. Script gancho en el servidor

Si tiene acceso directo al servidor del repositorio, puede instalar un script gancho pre-commit que rechace todas las confirmaciones con mensajes de registro cortos o vacíos.

En la carpeta del repositorio en el servidor, hay una subcarpeta `hooks` que contiene algunos scripts ganchos de ejemplo que puede utilizar. El archivo `pre-commit.tmpl` contiene un script de ejemplo que rechaza las confirmaciones si no se ha introducido un mensaje de registro, o si ese mensaje es demasiado corto. El archivo también tiene comentarios sobre cómo instalar/utilizar este script. Simplemente siga las instrucciones de ese archivo.

Este método es la forma recomendada si sus usuarios también utilizan otros clientes de Subversion además de TortoiseSVN. La parte negativa es que la confirmación se rechaza por el servidor, y por tanto los usuarios obtendrán un mensaje de error. El cliente no puede saber antes de realizar la confirmación que se rechazará. Si desea que TortoiseSVN deshabilite el botón **Aceptar** mientras el mensaje de registro sea demasiado corto, utilice el método descrito a continuación.

### B.2.2. Propiedades del proyecto

TortoiseSVN utiliza propiedades para controlar algunas de sus funciones. Una de esas propiedades es la propiedad `tsvn:logminsize`.

Si establece esa propiedad en una carpeta, TortoiseSVN deshabilitará el botón **Aceptar** en todos los diálogos de confirmación hasta que el usuario introduzca un mensaje de registro con al menos la longitud especificada en la propiedad.

Para obtener información detallada sobre esas propiedades del proyecto, por favor consulte [Sección 4.18](#), “Configuración del proyecto”.

## B.3. Actualizar los archivos seleccionados desde el repositorio

Normalmente actualizará su copia de trabajo utilizando TortoiseSVN → Actualizar. Pero si sólo desea obtener algunos archivos nuevos que un colega ha añadido sin fusionar ningún cambio en los demás archivos a la vez, necesitará otra forma de actuación.

Utilice TortoiseSVN → Comprobar Modificaciones. y pulse en Comprobar repositorio para ver lo que se cambió en el repositorio. Seleccione los archivos que desea actualizar localmente, y utilice el menú contextual para actualizar sólo esos archivos.

## B.4. Deshacer revisiones en el repositorio

### B.4.1. Utilice el diálogo Registro de revisiones

La manera más simple para revertir los cambios desde una o varias versiones es usar el diálogo de registro de revisiones.

1. Seleccione el archivo o la carpeta en la que desea revertir los cambios. Si desea revertir todos los cambios, esta debería ser la carpeta más alta.
2. Seleccione TortoiseSVN → Mostrar registro para mostrar una lista de revisiones. Puede que necesite utilizar Obtener Todo o Siguientes 100 para mostrar la o las revisiones en las que está interesado.
3. Elija la revisión que desea revertir. Si desea deshacer un rango de revisiones, elija la primera y pulse la tecla **Mayús** mientras selecciona la última. Si quiere elegir revisiones sueltas y rangos, use la tecla **Ctrl** mientras selecciona revisiones. Haga click con el botón derecho en las selección y elija Menú Contextual Revertir cambios de esta revisión
4. O si desea hacer que una revisión antigua se convierta en la nueva revisión HEAD, haga click con el botón derecho en la revisión seleccionada, luego seleccione Menú Contextual → Revertir a esta revisión. Esto descartará *todos* los cambios que se hicieron después de la revisión seleccionada.

Ha revertido los cambios dentro de su copia de trabajo. Compruebe los resultados, y luego confirme los cambios.

### B.4.2. Utilice el diálogo Fusionar

If you want to enter revision numbers as a list, you can use the Merge dialog. The previous method uses merging behind the scenes; this method uses it explicitly.

1. En su copia de trabajo seleccione TortoiseSVN → Fusionar.
2. In the Merge Type dialog select Merge a range of revisions.
3. In the From: field enter the full repository URL of your working copy folder. This should come up as the default URL.
4. In the Revision range to merge field enter the list of revisions to roll back (or use the log dialog to select them as described above).
5. Make sure the Reverse merge checkbox is checked.
6. In the Merge options dialog accept the defaults.
7. Click Merge to complete the merge.

Ha revertido los cambios dentro de su copia de trabajo. Compruebe los resultados son los esperados, y luego confirme los cambios.

### B.4.3. Utilice svndumpfilter

Dado que TortoiseSVN nunca pierde datos, sus revisiones “deshechas” aún existen como revisiones intermedias en el repositorio. Sólo se ha cambiado la revisión HEAD a su estado previo. Si desea hacer que las revisiones desaparezcan completamente de su repositorio, borrando toda traza que indique que alguna vez existieron, deberá utilizar medidas más extremas. A menos que haya una buena razón para hacer esto, *no se recomienda*. Una razón posible sería que alguien haya confirmado un documento confidencial en un repositorio 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>].

## B.5. Comparar dos revisiones de un archivo o carpeta

Si desea comparar dos revisiones en la historia de un ítem, por ejemplo las revisiones 100 y 200 del mismo archivo, utilice TortoiseSVN → Mostrar Registro para ver la historia de revisiones de ese archivo. Seleccione las dos revisiones que desea comparar y luego utilice Menú Contextual → Comparar Revisiones.

Si desea comparar el mismo ítem en dos árboles diferentes, por ejemplo el tronco y una rama, puede utilizar el navegador de repositorios para abrir ambos árboles, seleccionar el archivo en los dos lugares, y luego utilizar Menú Contextual → Comparar Revisiones.

Si desea comparar dos árboles para ver lo que ha cambiado, por ejemplo el tronco y una versión etiquetada, puede utilizar TortoiseSVN → Gráfico de Revisiones. Seleccione los dos nodos a comparar, y luego utilice Menú Contextual → Comparar Revisiones HEAD. Esto le mostrará una lista de archivos cambiados, y luego podrá seleccionar los archivos individuales para ver los cambios en detalle. También puede exportar una estructura de árbol conteniendo todos los archivos cambiados, o simplemente una lista de todos los archivos cambiados. Puede obtener más información en [Sección 4.11.3, “Comparando carpetas”](#). Alternativamente, utilice Menú contextual → Diff unificado de las revisiones HEAD para ver un resumen de todas las diferencias, con un contexto mínimo.

## B.6. Incluir un sub-proyecto común

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. Utilice `svn:externals`

Establece la propiedad `svn:externals` para una carpeta en su proyecto. Esta propiedad contiene una o más líneas; cada línea tiene el nombre de una subcarpeta que quiere utilizar como la carpeta de obtención del código común, y la URL del repositorio que desea obtener ahí. Para saber todos los detalles lea [Sección 4.19, “Ítems externos”](#).

Confirme la nueva carpeta. Ahora, cuando actualice, Subversion traerá una copia de ese proyecto desde su repositorio en su copia de trabajo. Las subcarpetas se crearán automáticamente si es necesario. Cada vez que actualice su copia de trabajo, también recibirá la última versión de todos los proyectos externos.

Si el proyecto externo está en el mismo repositorio, cualquier cambio que haga allí se incluirá en la lista de confirmación cuando confirme su proyecto 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.

De los tres métodos descritos, éste es el único que no necesita preparación en el lado del cliente. Una vez que los externos se especifican en las propiedades de la carpeta, todos los clientes obtendrán carpetas rellenas cuando se actualicen.

### B.6.2. Utilice una copia de trabajo anidada

Cree una nueva carpeta dentro de su proyecto para contener el código común, pero no la añada a Subversion.

Seleccione TortoiseSVN → Obtener en la nueva carpeta, y obtenga una copia del código común en ella. Ahora tiene anidada una copia de trabajo separada dentro de su copia de trabajo principal.

Las dos copias de trabajo son independientes. Cuando confirme los cambios de la padre, los cambios en la copia de trabajo anidada se ignoran. De la misma forma, cuando actualice el padre, la copia de trabajo anidada no se actualiza.

### B.6.3. Utilice una ruta 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 sólo funcionará en un entorno de todo-PCs, y necesitará documentar los mapeos de unidades requeridos para que su equipo sepa dónde están esos archivos misteriosos. Este método debería utilizarse estrictamente en entornos de desarrollo cerrados, y no se recomienda para su uso generalizado.

### B.6.4. Add the project to the repository

The maybe easiest way is to simply add the project in a subfolder to your own project working copy. However this has the disadvantage that you have to update and upgrade this external project manually.

To help with the upgrade, TortoiseSVN provides a command in the explorer right-drag context menu. Simply right-drag the folder where you unzipped the new version of the external library to the folder in your working copy, and then select Context Menu → SVN Vendorbranch here. This will then copy the new files over to the target folder while automatically adding new files and removing files that aren't in the new version anymore.

## B.7. Crear un acceso directo a un repositorio

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 archivos que ya están versionados

Si accidentalmente añadió algunos archivos que deberían haber sido ignorados, ¿cómo puede sacarlos del control de versiones sin perderlos? Quizás tiene su propio archivo de configuración IDE que no es parte del proyecto, pero que le costó bastante tiempo ajustar a su gusto.

Si aún no has subido lo añadido, entonces todo lo que tienes que hacer es usar TortoiseSVN → Undo Add... para deshacer el añadir. Deberías entonces añadir el/los fichero/s a la lista de ignorar para que no vuelvan a añadirse por error posteriormente.

Si los archivos ya están en el repositorio, deben ser borrados del repositorio y agregados a la lista de ignorados. Afortunadamente TortoiseSVN tiene un acelerador conveniente para hacer esto. TortoiseSVN → Desversionar y agregar a la lista de ignorados marcará primero al archivo/carpeta para borrado del repositorio, conservando la copia local. También agrega este elemento a la lista de ignorados para que no sea agregado nuevamente a la Subversion por error. Una vez que esto está hecho sólo necesita confirmar a la carpeta contenedora.

## B.9. Desversionar una copia de trabajo

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 [Sección 4.27.1, “Eliminando una copia de trabajo del control de versiones”](#) to find out how.

## B.10. Eliminar una copia de trabajo

Si tienes una copia de trabajo que ya no necesitas, ¿cómo te deshaces limpiamente de ella? Fácil - ¡simplemente bórrala con el Explorador de Windows! Las copias de trabajo son entidades locales, y son auto-contenidas. Borrar una copia de trabajo en el Explorador de Windows no afecta en absoluto los datos en el repositorio.

---

# Apéndice C. Trucos útiles para los administradores

Este apéndice contiene soluciones a los problemas o preguntas que pueda tener cuando es responsable de distribuir TortoiseSVN a múltiples ordenadores clientes.

## C.1. Instalar TortoiseSVN utilizando políticas de grupo

El instalador de TortoiseSVN viene en un archivo MSI, lo que significa que no debería tener problemas para añadir ese archivo MSI a las políticas de grupo de su controlador de dominio.

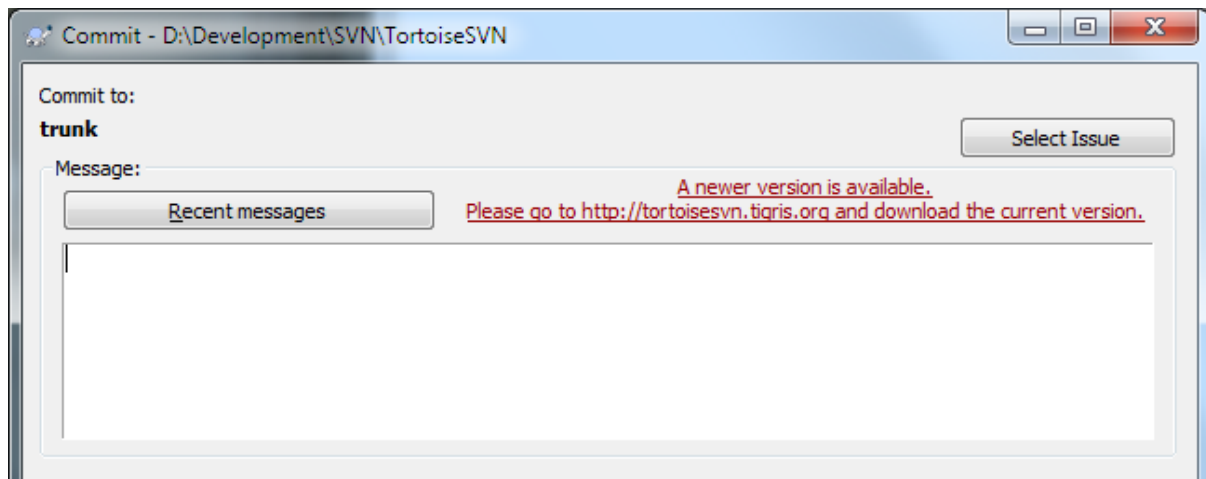
Puede encontrar una buena guía sobre cómo hacerlo en el artículo de la base de datos de conocimiento 314934 de Microsoft: <http://support.microsoft.com/?kbid=314934>.

TortoiseSVN debe instalarse bajo *Configuración de Equipo* y no bajo *Configuración de Usuario*. Esto es así porque TortoiseSVN necesita las DLL CRT y MFC, que solo pueden instalarse *por equipo* y no *por usuario*. Si necesita desplegar TortoiseSVN obligatoriamente en modo por usuario, entonces debe instalar los paquetes MFC y CRT versión 12 de Microsoft en cada equipo en que desee instalar TortoiseSVN también por usuario.

Si lo desea, puede personalizar el fichero MSI par que todos sus usuarios tengan la misma configuración. La configuración de TSVN se almacena en el registro en HKEY\_CURRENT\_USER\Software\TortoiseSVN y los parámetros generales de Subversion (que afectan a todos los clientes Subversion) se guardan en ficheros de configuración en %APPDATA%\Subversion. Si necesita ayuda con la personalización del MSI, pruebe en algún foro de MSI transform o busque en la web “MSI transform”.

## C.2. Redirigir la comprobación de actualización

TortoiseSVN comprueba si hay una nueva versión disponible cada pocos días. Si hay una nueva versión disponible, se muestra una notificación en el diálogo de commit.



**Figura C.1. El diálogo de confirmación, mostrando la notificación de actualización**

Si eres responsable de muchos usuarios, puede que quieras que tus usuarios sólo utilicen versiones que hayas aprobado, y no que intalen siempre la última versión. Probablemente no desees que se muestre la notificación de actualización, de modo que tus usuarios no actualicen de manera inmediata.

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. Estableciendo la variable de entorno SVN\_ASP\_DOT\_NET\_HACK

Desde la versión 1.4.0, el instalador de TortoiseSVN ya no ofrece al usuario la opción de establecer la variable de entorno SVN\_ASP\_DOT\_NET\_HACK, dado que eso ha causado muchos problemas y confusiones a algunos usuarios que siempre instalan *todo* sin importar si saben para lo que sirve.

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

Por favor ten en cuenta que este hack es sólo necesario si todavía estás utilizando VS.NET2002. ¡Ninguna de las versiones posteriores de Visual Studio requieren que este hack esté activado! Así que, a menos que estés usando esta herramienta antigua, ¡NO UTILICES ESTO!

### C.4. Deshabilitar entradas del menú contextual

Empezando en la versión 1.5.0, TortoiseSVN le permite deshabilitar (realmente, ocultar) entradas del menú contextual. Dado que esta característica no debería usarse a la ligera sino únicamente cuando hay una razón necesaria, no hay interfaz de usuario para hacerlo y debe modificarse directamente el registro de Windows. Esto puede utilizarse para deshabilitar ciertos comandos que los usuarios no deberían usar. Pero por favor tenga en cuenta que sólo se ocultan las entradas del menú contextual en el *Explorador*, pero los comandos aún están disponibles por otros medios, por ejemplo la línea de comandos ¡o incluso mediante otros diálogos en el propio TortoiseSVN!

Las claves de registro que almacenan la información sobre qué menús contextuales mostrar son HKEY\_CURRENT\_USER\Software\TortoiseSVN\ContextMenuEntriesMaskLow y HKEY\_CURRENT\_USER\Software\TortoiseSVN\ContextMenuEntriesMaskHigh.

Cada una de estas entradas de registro es un valor DWORD, en el que cada bit corresponde a una entrada de menú específica. Un bit a 1 significa que la entrada de menú correspondiente se desactiva.

Valor	Entrada de menú
0x0000000000000001	Obtener
0x0000000000000002	Actualizar
0x0000000000000004	Confirmar
0x0000000000000008	Añadir
0x0000000000000010	Revertir
0x0000000000000020	Limpieza
0x0000000000000040	Resolver



Valor	Entrada de menú
0x0000000000000080	Cambiar
0x0000000000000100	Importar
0x0000000000000200	Exportar
0x0000000000000400	Crear repositorio aquí
0x0000000000000800	Rama/Etiqueta
0x0000000000001000	Fusionar
0x0000000000002000	Eliminar
0x0000000000004000	Renombrar
0x0000000000008000	Actualizar a la revisión
0x0000000000010000	Diff
0x0000000000020000	Mostrar registro
0x0000000000040000	Editar conflictos
0x0000000000080000	Relocalizar
0x0000000000100000	Comprobar modificaciones
0x0000000000200000	Ignorar
0x0000000000400000	Navegador de repositorios
0x0000000000800000	Autoría
0x0000000001000000	Crear parche
0x0000000002000000	Aplicar parche
0x0000000004000000	Gráfico de revisiones
0x0000000008000000	Bloqueo
0x0000000010000000	Eliminar bloqueo
0x0000000020000000	Propiedades
0x0000000040000000	Diferenciar con URL
0x0000000080000000	Eliminar ítems no versionados
0x0000000100000000	Fusionar todo
0x0000000200000000	Dif con version anterior
0x0000000400000000	Pegar
0x0000000800000000	Actualizar copia de trabajo
0x0000001000000000	Comparar más tarde
0x0000002000000000	Diff with 'filename'
0x0000004000000000	Unified diff
0x2000000000000000	Configuración
0x4000000000000000	Ayuda
0x8000000000000000	Acerca de

**Tabla C.1. Entradas de menú y sus valores**

Example: to disable the “Relocate” the “Delete unversioned items” and the “Settings” menu entries, add the values assigned to the entries like this:

```
0x00000000000080000
+ 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 habilitar las entradas de menú de nuevo, simplemente elimine las dos entradas del registro.

---

# Apéndice D. Automatizando TortoiseSVN

Dado que todos los comandos de TortoiseSVN se controlan a través de parámetros de línea de comandos, puede automatizarlo con scripts batch o iniciando comandos y diálogos específicos desde otros programas (por ejemplo, desde su editor de textos favorito).



## Importante

Recuerde que TortoiseSVN es un cliente GUI, y esta guía de automatización le muestra cómo conseguir que aparezcan los diálogos de TortoiseSVN para recolectar información del usuario. Si desea escribir un script que no requiera entradas del usuario, debería utilizar el cliente de línea de comandos oficial de Subversion.

## D.1. Comandos de TortoiseSVN

El programa de interfaz gráfico de TortoiseSVN se llama `TortoiseProc.exe`. Todos los comandos se especifican con el parámetro `/command:abcd` donde `abcd` es el nombre obligatorio del comando. La mayoría de estos comando necesitan al menos un argumento de ruta, que se proporciona con `/path:"alguna\ruta"`. En la siguiente tabla el comando se refiere al parámetro `/command:abcd` y la ruta se refiere al parámetro `/path:"alguna\ruta"`.

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.

Como algunos comandos pueden tomar una lista de rutas de destino (por ejemplo, si se confirman varios archivos específicos) el parámetro `/path` puede admitir varias rutas, separadas con un caracter `*`.

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.

El diálogo de progreso que se utiliza para las confirmaciones, actualizaciones y muchos más comandos normalmente queda abierto después que el comando termina, hasta que el usuario pulsa el botón **Aceptar**. Esto puede cambiarse marcando la opción correspondiente en el diálogo de configuración. Pero utilizando esa opción se cerrará el diálogo de progreso, sin importar si se inició el comando desde el archivo batch o desde el menú contextual de TortoiseSVN.

Para especificar una localización diferente del archivo de configuración, utilice el parámetro `/configdir:"ruta\al\directorio\de\configuración"`. Esto tomará preferencia sobre la ruta por defecto, incluyendo cualquier configuración en el registro.

Para cerrar el diálogo de progreso al final de un comando automáticamente sin utilizar la configuración permanente, puede pasar el parámetro `/closeonend`.

- `/closeonend:0` no cierra el diálogo automáticamente
- `/closeonend:1` auto cerrar si no hay errores
- `/closeonend:2` cierra automáticamente si no hay errores ni conflictos
- `/closeonend:3` cierra automáticamente si no hay errores, conflictos ni fusiones

Para cerrar el diálogo de progreso para operaciones locales si no hubiera errores ni conflictos, pase el parámetro `/closeforlocal`.

La siguiente tabla lista todos los comandos a los que se puede acceder utilizando la línea de comando de TortoiseProc.exe. Como se describe más arriba, se deben utilizar en la forma `/command:abcd`. En la tabla, el prefijo `/command` se omite para ahorrar espacio.

Comando	Descripción
<code>:about</code>	Muestra el diálogo Acerca de. También se muestra si no se pasa ningún comando.
<code>:log</code>	<p>Opens the log dialog. The <code>/path</code> specifies the file or folder for which the log should be shown. Additional options can be set:</p> <ul style="list-style-type: none"> <li>• <code>/startrev:xxx</code>,</li> <li>• <code>/endrev:xxx</code>,</li> <li>• <code>/strict</code> enables the 'stop-on-copy' checkbox,</li> <li>• <code>/merge</code> enables the 'include merged revisions' checkbox,</li> <li>• <code>/datemin: "{datestring}"</code> sets the start date of the filter, and</li> <li>• <code>/datemax: "{datestring}"</code> sets the end date of the filter. The date format is the same as used for svn date revisions.</li> <li>• <code>/findstring: "filterstring"</code> fills in the filter text,</li> <li>• <code>/findtext</code> forces the filter to use text, not regex, or</li> <li>• <code>/findregex</code> forces the filter to use regex, not simple text search, and</li> <li>• <code>/findtype:X</code> with X being a number between 0 and 511. The numbers are the sum of the following options: <ul style="list-style-type: none"> <li>• <code>/findtype:0</code> filtrar por todo</li> <li>• <code>/findtype:1</code> filtrar por mensajes</li> <li>• <code>/findtype:2</code> filtrar por ruta</li> <li>• <code>/findtype:4</code> filtrar por autores</li> <li>• <code>/findtype:8</code> filtrar por revisiones</li> <li>• <code>/findtype:16</code> not used</li> <li>• <code>/findtype:32</code> filter by bug ID</li> <li>• <code>/findtype:64</code> not used</li> <li>• <code>/findtype:128</code> filter by date</li> <li>• <code>/findtype:256</code> filter by date range</li> </ul> </li> <li>• If <code>/outfile:path\to\file</code> is specified, the selected revisions are written to that file when the log dialog is closed. The revisions are written in the same format as is used to specify revisions in the merge dialog.</li> </ul> <p>An svn date revision can be in one of the following formats:</p> <ul style="list-style-type: none"> <li>• <code>{2006-02-17}</code></li> <li>• <code>{15:30}</code></li> </ul>

Comando	Descripción
	<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 el diálogo de obtener. El <code>/path</code> especifica el directorio de destino y el <code>/url</code> especifica la URL de la cual se obtiene el proyecto. Si especifica las clave <code>/blockpathadjustments</code> , se bloquean los ajustes del path de obtención automática. El <code>/revision:XXX</code> especifica la revisión que se obtiene.
:import	Abre el dialogo de importación. <code>&lt;placeholder-1&gt;</code> especifica el directorio con los datos a importar. Puedes especificar <code>&lt;placeholder-2&gt;</code> switch para pasar un mensaje log predefinido al diálogo de importación. O, si no quiere pasar el mensaje de log de la línea de comandos, usa <code>&lt;placeholder-3&gt;</code> , donde <code>&lt;placeholder-4&gt;</code> apunta a un fichero que contiene el mensaje de log. <code>&lt;/placeholder-4&gt;</code> <code>&lt;/placeholder-3&gt;</code> <code>&lt;/placeholder-2&gt;</code> <code>&lt;/placeholder-1&gt;</code>
:update	Updates the working copy in <code>/path</code> to HEAD. If the option <code>/rev</code> is given then a dialog is shown to ask the user to which revision the update should go. To avoid the dialog specify a revision number <code>/rev:1234</code> . Other options are <code>/nonrecursive</code> , <code>/ignoreexternals</code> and <code>/includeexternals</code> . The <code>/stickydepth</code> indicates that the specified depth should be sticky, creating a sparse checkout. The <code>/skipprechecks</code> can be set to skip all checks that are done before an update. If this is specified, then the <b>Mostrar registro</b> button is disabled, and the context menu to show diffs is also disabled after the update.
:commit	Abre el diálogo de confirmación. <code>/path</code> especifica el directorio de destino o la lista de archivos a confirmar. También puede especificar la opción <code>/logmsg</code> para pasar un mensaje de registro predefinido al diálogo de confirmación. O, si no desea pasar el mensaje de registro en la línea de comandos, utilice <code>/logmsgfile:ruta</code> , donde <code>ruta</code> apunta a un archivo que contiene el mensaje de registro. Para rellenar la caja de texto del ID del error (en caso que haya preparado adecuadamente la integración con el sistema de control de errores), puede utilizar <code>/bugid: "el ID del error aquí"</code> .
:add	Añade los archivos en <code>/path</code> al control de versiones.
:revert	Revierte las modificaciones locales de una copia de trabajo. La <code>/path</code> dice qué items deben revertirse.
:cleanup	Cleans up interrupted or aborted operations and unlocks the working copy in <code>/path</code> . You also have to pass the <code>/cleanup</code> to actually do the cleanup. Use <code>/noui</code> to prevent the result dialog from popping up (either telling about the cleanup being finished or showing an error message). <code>/noprogessui</code> also disables the progress dialog. <code>/nodlg</code> disables showing the cleanup dialog where the user can

Comando	Descripción
	choose what exactly should be done in the cleanup. The available actions can be specified with the options <code>/cleanup</code> for status cleanup, <code>/breaklocks</code> to break all locks, <code>/revert</code> to revert uncommitted changes, <code>/delunversioned</code> , <code>/delignored</code> , <code>/refreshshell</code> , <code>/externals</code> , <code>/fixtimestamps</code> and <code>/vacuum</code> .
<code>:resolve</code>	Marca un archivo en conflicto especificado en <code>/path</code> como resuelto. Si se especifica <code>/noquestion</code> , la resolución se realiza sin preguntar al usuario primero si realmente debe hacerse.
<code>:reprocreate</code>	Crea un repositorio en <code>/path</code>
<code>:switch</code>	Opens the switch dialog. The <code>/path</code> specifies the target directory and <code>/url</code> the URL to switch to.
<code>:export</code>	Exporta la copia de trabajo en <code>/path</code> a otra carpeta. Si <code>/path</code> apunta a una carpeta no versionada, un cuadro de diálogo le preguntará por la URL a exportar a la carpeta <code>/path</code> . Si indica la clave <code>/blockpathadjustments</code> , se bloquearán los ajustes automáticos en la ruta de exportación.
<code>:dropexport</code>	Exports the working copy in <code>/path</code> to the directory specified in <code>/droptarget</code> . This exporting does not use the export dialog but executes directly. The option <code>/overwrite</code> specifies that existing files are overwritten without user confirmation, and the option <code>/autorename</code> specifies that if files already exist, the exported files get automatically renamed to avoid overwriting them. The option <code>/extended</code> can specify either <code>localchanges</code> to only export files that got changed locally, or <code>unversioned</code> to also export all unversioned items as well.
<code>:dropvendor</code>	Copies the folder in <code>/path</code> recursively to the directory specified in <code>/droptarget</code> . New files are added automatically, and missing files get removed in the target working copy, basically ensuring that source and destination are exactly the same. Specify <code>/noui</code> to skip the confirmation dialog, and <code>/noprogessui</code> to also disable showing the progress dialog.
<code>:merge</code>	Opens the merge dialog. The <code>/path</code> specifies the target directory. For merging a revision range, the following options are available: <code>/fromurl:URL</code> , <code>/revrange:cadena</code> . For merging two repository trees, the following options are available: <code>/fromurl:URL</code> , <code>/tourl:URL</code> , <code>/fromrev:xxx</code> and <code>/torev:xxx</code> .
<code>:mergeall</code>	Abre el diálogo fusionar todo. <code>/path</code> especifica el directorio de destino.
<code>:copy</code>	Brings up the branch/tag dialog. The <code>/path</code> is the working copy to branch/tag from. And the <code>/url</code> is the target URL. If the url starts with a <code>^</code> it is assumed to be relative to the repository root. To already check the option <code>Switch working copy to new branch/tag</code> you can pass the <code>/switchaftercopy</code> switch. To check the option <code>Crear carpetas intermedias</code> . pass the <code>/makeparents</code> switch. You can also specify the <code>/logmsg</code> switch to pass a predefined log message to the branch/tag dialog. Or, if you don't want to pass the log message on the command line, use <code>/logmsgfile:ruta</code> , where <code>ruta</code> points to a file containing the log message.
<code>:settings</code>	Abre el diálogo de configuración.
<code>:remove</code>	Elimina el/los archivo/s en <code>/path</code> del control de versiones.
<code>:rename</code>	Renombra el archivo en <code>/path</code> . El nuevo nombre del archivo se pregunta con un diálogo. Para evitar la pregunta sobre renombrar archivos similares en un paso, utilice <code>/noquestion</code> .
<code>:diff</code>	Starts the external diff program specified in the TortoiseSVN settings. The <code>/path</code> specifies the first file. If the option <code>/path2</code> is set, then the diff program is started with those two files. If <code>/path2</code> is omitted, then the diff is done between the file in <code>/path</code> and its BASE. If the specified file also has property modifications,

Comando	Descripción
	the external diff tool is also started for each modified property. To prevent that, pass the option <code>/ignoreprops</code> . To explicitly set the revision numbers use <code>/startrev:xxx</code> and <code>/endrev:xxx</code> , and for the optional peg revision use <code>/pegrevision:xxx</code> . If <code>/blame</code> is set and <code>/path2</code> is not set, then the diff is done by first blaming the files with the given revisions. The parameter <code>/line:xxx</code> specifies the line to jump to when the diff is shown.
<code>:shelve</code>	Shelves the specified paths in a new shelf. The option <code>/shelfname:name</code> specifies the name of the shelf. An optional log message can be specified with <code>/logmsg:message</code> . If option <code>/checkpoint</code> is passed, the modifications of the files are kept.
<code>:unshelve</code>	Applies the shelf with the name <code>/shelfname:name</code> to the working copy path. By default the last version of the shelf is applied, but you can specify a version with <code>/version:X</code> .
<code>:showcompare</code>	<p>Dependiendo de las URLs y las revisiones a comparar, esto o bien muestra un diff unificado (si la opción <code>unified</code> está establecida), o un diálogo con una lista de archivos que han cambiado o, en el caso que las URLs apunten a archivos, inicia el visor de diferencias para esos dos archivos.</p> <p>Las opciones <code>url1</code>, <code>url2</code>, <code>revision1</code> y <code>revision2</code> deben especificarse. Las opciones <code>pegrevision</code>, <code>ignoreancestry</code>, <code>blame</code> y <code>unified</code> son opcionales.</p> <p>If the specified url also has property modifications, the external diff tool is also started for each modified property. To prevent that, pass the option <code>/ignoreprops</code>.</p> <p>If a unified diff is requested, an optional <code>prettyprint</code> option can be specified which will show the merge-info properties in a more user readable format.</p>
<code>:conflicteditor</code>	Inicia el editor de conflictos especificado en la configuración de TortoiseSVN con los archivos correctos para el archivo en conflicto de <code>/path</code> .
<code>:relocate</code>	Abre el diálogo relocalizar. <code>/path</code> especifica la ruta de la copia de trabajo a relocalizar.
<code>:help</code>	Abre el archivo de ayuda.
<code>:repostatus</code>	Abre el diálogo comprobar-notificaciones. <code>/path</code> especifica el directorio de trabajo. Si <code>/remote</code> esta especificado, entonces el diálogo contacta al repositorio inmediatamente al inicio, como si el usuario hubiera pulsado sobre el boton <b>Revisar el repositorio</b> .
<code>:repobrowser</code>	<p>Inicia el diálogo del navegador de repositorios, apuntando a la URL de la copia de trabajo dada en <code>/path</code> o <code>/path</code> apunta directamente a una URL.</p> <p>Una opción adicional <code>/rev:xxx</code> puede emplearse para especificar la revisión que deberá mostrar el explorador del repositorio. Si se omite la <code>/rev:xxx</code>, por defecto muestra la revisión HEAD.</p> <p>Si <code>/path</code> apunta a una URL, la <code>/projectpropertiespath:path/to/wc</code> especifica la ruta o path desde el que leer y usar las propiedades del proyecto.</p> <p>Si <code>/outfile:path\to\file</code> se especifica, la URL y revisión seleccionadas se escriben a dicho fichero cuando el explorador del repositorio se cierra. La primera línea en el fichero de texto contiene la URL, la segunda es la revisión en formato de texto.</p>
<code>:ignore</code>	Añade todos los objetivos en <code>/path</code> a la lista de ignorados, es decir, añade la propiedad <code>svn:ignore</code> a esos archivos.

Comando	Descripción
:blame	<p>Abre el diálogo de autoría para el archivo especificado en <code>/path</code>.</p> <p>Si las opciones <code>/startrev</code> y <code>/endrev</code> se establecen, entonces el diálogo preguntando por el rango de la autoría no se muestra y se utilizan los valores de revisión especificados.</p> <p>Si la opción <code>/line:nnn</code> se establece, TortoiseBlame se abrirá mostrando el número de línea especificado.</p> <p>Las opciones <code>/ignoreeol</code>, <code>/ignorespaces</code> y <code>/ignoreallspaces</code> también se soportan.</p>
:cat	<p>Graba un archivo de una URL o una copia de trabajo dada en <code>/path</code> al lugar dado en <code>/savepath:ruta</code>. La revisión se da en <code>/revision:xxx</code>. Esto puede utilizarse para obtener un archivo cno una revisión concreta.</p>
:createpatch	<p>Creates a patch file for the path given in <code>/path</code>. To skip the file Save-As dialog you can pass <code>/savepath:ruta</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.</p>
:revisiongraph	<p>Muestra el gráfico de revisiones para la ruta dada en <code>/path</code>.</p> <p>To create an image file of the revision graph for a specific path, but without showing the graph window, pass <code>/output:path</code> with the path to the output file. The output file must have an extension that the revision graph can actually export to. These are: <code>.svg</code>, <code>.wmf</code>, <code>.png</code>, <code>.jpg</code>, <code>.bmp</code> and <code>.gif</code>.</p> <p>Since the revision graph has many options that affect how it is shown, you can also set the options to use when creating the output image file. Pass these options with <code>/options:XXXX</code>, where <code>XXXX</code> is a decimal value. The best way to find the required options is to start the revision graph the usual way, set all user-interface options and close the graph. Then the options you need to pass on the command line can be read from the registry <code>HKCU\Software\TortoiseSVN\RevisionGraphOptions</code>.</p>
:lock	<p>Bloquea un archivo o todos los archivos en un directorio dado en <code>/path</code>. Se muestra el diálogo Bloquear para que el usuario pueda introducir un comentario para el bloqueo.</p>
:unlock	<p>Desbloquea un archivo o todos los archivos en un directorio dado en <code>/path</code>.</p>
:rebuildiconcache	<p>Reconstruye la caché de íconos de Windows. Utilice esto sólo en el caso que los íconos de Windows se hayan corrompido. Un efecto secundario de esto (y no puede evitarse) es que los íconos del escritorio se recolocan. Para eliminar el mensaje de advertencia, pase <code>/noquestion</code>.</p>
:properties	<p>Shows the properties dialog for the path given in <code>/path</code>.</p> <p>For dealing with versioned properties this command requires a working copy.</p> <p>Revision properties can be viewed/changed if <code>/path</code> is an URL and <code>/rev:xxx</code> is specified.</p> <p>To open the properties dialog directly for a specific property, pass the property name as <code>/property:name</code>.</p>
:sync	<p>Exports/imports settings, either depending on whether the current settings or the exported settings are newer, or as specified.</p>



Comando	Descripción
	<p>If a path is passed with <code>/path</code>, then the path is used to store or read the settings from.</p> <p>The parameter <code>/askforpath</code> will show a file open/save dialog for the user to chose the export/import path.</p> <p>If neither <code>/load</code> nor <code>/save</code> is specified, then TortoiseSVN determines whether to export or import the settings by looking at which ones are more recent. If the export file is more recent than the current settings, then the settings are loaded from the file. If the current settings are more recent, then the settings are exported to the settings file.</p> <p>If <code>/load</code> is specified, the settings are imported from the settings file.</p> <p>If <code>/save</code> is specified, the current settings are exported to the settings file.</p> <p>The parameter <code>/local</code> forces a settings export to include local settings, i.e. settings that refer to local paths.</p>

**Tabla D.1. Lista de comandos y opciones disponibles**

Ejemplos (que deben introducirse en una única línea):

```
TortoiseProc.exe /command:commit
                 /path:"c:\svn_wc\file1.txt*c:\svn_wc\file2.txt"
                 /logmsg:"mensaje de registro de prueba" /closeonend
```

```
TortoiseProc.exe /command:update /path:"c:\svn_wc\" /closeonend
```

```
TortoiseProc.exe /command:log /path:"c:\svn_wc\file1.txt"
                 /startrev:50 /endrev:60 /closeonend
```

## D.2. Manejador de URLs Tsvncmd

Utilizando URLs especiales, también es posible llamar al TortoiseProc desde una página web.

TortoiseSVN registra un protocolo nuevo `tsvncmd`: que puede ser utilizado para crear enlaces que ejecutan comandos TortoiseSVN. Los comandos y parámetros son los mismos que cuando se automatiza TortoiseSVN desde la línea de comando.

El formato de la URL `tsvncmd`: es como sigue:

```
tsvncmd:command:cmd?parameter:paramvalue?parameter:paramvalue
```

siendo `cmd` uno de los comandos permitidos, `parametro` el nombre de un parámetro como `ruta` o `revision`, y `valor` el valor a utilizar para ese parámetro. La lista de parámetros permitidos depende del comando utilizado.

Los siguientes comandos son permitidos con `tsvncmd`: URLs:

- `:update`
- `:commit`
- `:diff`
- `:repobrowser`
- `:checkout`

- :export
- :blame
- :repostatus
- :revisiongraph
- :showcompare
- :log

Una simple URL puede tener este aspecto:

```
<a href="tsvncmd:command:update?path:c:\svn_wc?rev:1234">Update</a>
```

o en un caso más complejo:

```
<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 de TortoiseIDiff

La herramienta de diferencia de imágenes tiene algunas pocas opciones de comando que puede utilizar para controlar cómo se inicia la herramienta. El programa se llama `TortoiseIDiff.exe`.

La siguiente tabla contiene todas las opciones que pueden pasarse a la herramienta de diferencia de imágenes por la línea de comandos.

Opción	Descripción
:left	Ruta al archivo mostrado a la izquierda.
:lefttitle	Una cadena de título. Esta cadena se utiliza en el título de la vista de la imagen en lugar de la ruta completa al archivo de la imagen.
:right	Ruta al archivo mostrado a la derecha.
:righttitle	Una cadena de título. Esta cadena se utiliza en el título de la vista de la imagen en lugar de la ruta completa al archivo de la imagen.
:overlay	Si se especifica, la herramienta de diferencias de imágenes cambia al modo superpuesto (fundido alpha).
:fit	Si se especifica, la herramienta de diferencias de imágenes ajusta ambas imágenes juntas.
:showinfo	Muestra la caja de información de imagen.

#### Tabla D.2. Lista de las opciones disponibles

Ejemplo (que debería introducirse como una única línea):

```
TortoiseIDiff.exe /left:"c:\images\img1.jpg" /lefttitle:"imagen 1"
                  /right:"c:\images\img2.jpg" /righttitle:"imagen 2"
                  /fit /overlay
```

### D.4. TortoiseUDiff Commands

The unified diff viewer has only two command line options:

Opción	Descripción
/patchfile	Path to the unified diff file.
/p	Activates pipe mode. The unified diff is read from the console input.

### **Tabla D.3. Lista de las opciones disponibles**

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
```

---

# Apéndice E. Referencia cruzada del interface de línea de comandos

A veces este manual hace referencia a la documentación principal de Subversion, que describe Subversion en términos del Interfaz de Línea de Comandos (CLI). Para ayudarle a entender lo que TortoiseSVN hace tras el escenario, hemos compilado una lista mostrando los comandos CLI equivalentes a cada operación en la Interfaz de Usuario de TortoiseSVN.

## Nota

Incluso aunque hay equivalentes en CLI para lo que TortoiseSVN hace, recuerde que TortoiseSVN *no* llama al CLI sino que utiliza la librería de Subversion directamente.

Si cree que ha encontrado un error en TortoiseSVN, puede que le pidamos que intente reproducirlo utilizando el CLI, para que podamos distinguir los problemas de TortoiseSVN de los problemas de Subversion. Esta referencia le dice qué comando debe probar.

## E.1. Convenciones y reglas 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

Dado que TortoiseSVN es una Extensión del Shell de Windows, no es capaz de utilizar la noción de directorio de trabajo actual. Todas las rutas de las copias de trabajo se deben especificar con rutas absolutas, no relativas.

Algunos ítems son opcionales, y a menudo se controlan con casillas o botones de opciones en TortoiseSVN. Estas opciones se muestran [entre corchetes] en las definiciones de la línea de comandos.

## E.2. Comandos de TortoiseSVN

### E.2.1. Obtener

```
svn checkout [-depth ARG] [--ignore-externals] [-r rev] URL RUTA
```

El cuadro desplegable de profundidad está relacionado con el argumento `-depth`

Si se marca Omitir externos, utilice la opción `--ignore-externals`.

Si está obteniendo una revisión en concreto, especifíquela detrás de la URL utilizando la opción `-r`.

### E.2.2. Actualizar

```
svn info URL_of_WC
svn update [-r rev] RUTA
```

A día de hoy, la actualización no es una operación atómica en Subversion. Por tanto, TortoiseSVN primero encuentra la revisión HEAD del repositorio, y luego actualiza todos los ítems a ese número de revisión en concreto, evitando crear así una copia de trabajo mezclada.

Si sólo se selecciona un único ítem para actualizar, o si los ítems seleccionados no son todos del mismo repositorio, TortoiseSVN simplemente actualiza hasta HEAD.

Aquí no se utilizan opciones en la línea de comandos. Actualizar a la Revisión también implementa el comando actualizar, pero ofrece más opciones.

### E.2.3. Actualizar a la revisión

```
svn info URL_of_WC
svn update [-r rev] [-depth ARG] [--ignore-externals] RUTA
```

El cuadro desplegable de profundidad esta relacionado con el argumento `-depth`

Si se marca Omitir externos, utilice la opción `--ignore-externals`.

### E.2.4. Confirmar

En TortoiseSVN, el diálogo de confirmación utiliza varios comandos Subversion. La primera fase es una comprobación del estado que determina los ítems de su copia de trabajo que potencialmente pueden ser confirmados. Puede revisar la lista, comparar los archivos contra BASE y seleccionar los ítems que desea incluir en la confirmación.

```
svn status -v RUTA
```

Si se marca **Mostrar archivos no versionados**, TortoiseSVN le mostrará también todos los archivos y carpetas no versionados en su jerarquía de la copia de trabajo, teniendo en cuenta las reglas de ignorar. Esta característica en concreto no tiene equivalente directo en Subversion, ya que el comando `svn status` no entra en carpetas no versionadas.

Si selecciona cualquier archivo o carpeta no versionado, esos ítems primero se añadirán a su copia de trabajo.

```
svn add RUTA...
```

Cuando pulsa Aceptar, tiene lugar la confirmación en Subversion. Si ha dejado todas las casillas de selección de archivos sin tocar tal y como aparecieron, TortoiseSVN utilizará una única confirmación recursiva en la copia de trabajo. Si ha deseleccionado algún archivo, entonces se debe utilizar una confirmación no-recursiva (`-N`), y se debe especificar individualmente cada ruta en la línea de comandos de la confirmación.

```
svn commit -m "LogMessage" [-depth ARG] [--no-unlock] RUTA...
```

Mensaje de registro aquí representa los contenidos del cuadro de texto Mensaje de Registro. Puede estar vacío.

Si se marca **Mantener Bloqueos**, utilice la opción `--no-unlock`.

### E.2.5. Diff

```
svn diff RUTA
```

Si utiliza Diff desde el menú contextual principal, está comparando un archivo modificado contra su revisión BASE. La salida del comando CLI anterior también hace ésto y produce la salida en formato diff unificado.

Sin embargo, esto no es lo que TortoiseSVN utiliza. TortoiseSVN utiliza TortoiseMerge (o un programa de comparación de su elección) para mostrar las diferencias visualmente entre archivos de texto completos, por lo que no hay equivalente directo en el CLI.

También puede comparar 2 archivos cualquiera utilizando TortoiseSVN, estén o no bajo el control de versiones. TortoiseSVN simplemente alimenta con los dos archivos al programa de diferencias y le deja que él encuentre dónde están las diferencias.

## E.2.6. Mostrar registro

```
svn log -v -r 0:N --limit 100 [--stop-on-copy] RUTA
o
svn log -v -r M:N [--stop-on-copy] RUTA
```

Por defecto, TortoiseSVN intenta recoger 100 mensajes de registro utilizando el método `--limit`. Si la configuración establece que utilice las APIs antiguas, se usa la segunda forma para obtener los mensajes de registro de 100 revisiones del repositorio.

Si se marca Parar en copia/renombrado, utilice la opción `--stop-on-copy`.

## E.2.7. Comprobar modificaciones

```
svn status -v RUTA
o
svn status -u -v RUTA
```

La comprobación de estado inicialmente sólo muestra su copia de trabajo. Si pulsa en Comprobar repositorio, se comprueba también el repositorio para ver qué archivos cambiarían tras una actualización, lo que necesita la opción `-u`.

Si se marca Mostrar archivos no versionados, TortoiseSVN le mostrará también todos los archivos y carpetas no versionados en su jerarquía de la copia de trabajo, teniendo en cuenta las reglas de ignorar. Esta característica en concreto no tiene equivalente directo en Subversion, ya que el comando `svn status` no entra en carpetas no versionadas.

## E.2.8. Gráfico de revisión

El gráfico de revisiones es una característica única de TortoiseSVN. No hay equivalente en el cliente de línea de comandos.

Lo que hace TortoiseSVN es un

```
svn info URL_de_la_copia_de_trabajo
svn log -v URL
```

donde URL es el repositorio *raíz* y analiza la información devuelta.

## E.2.9. Navegador de repositorios

```
svn info URL_de_la_copia_de_trabajo
svn list [-r rev] -v URL
```

Puede utilizar `svn info` para determinar la raíz del repositorio, que es el nivel superior mostrado en el navegador de repositorios. No puede navegar más arriba de ese nivel. Además, este comando devuelve toda la información sobre bloqueos que se muestra en el navegador de repositorios.

La llamada `svn list` le mostrará los contenidos de un directorio, cuando se proporciona una URL y una revisión.

### E.2.10. Editar conflictos

Este comando no tiene equivalente CLI. Invoca TortoiseMerge o una herramienta externa de fusión/comparación en 3 paneles para mirar los archivos en conflictos y ver qué líneas hay que utilizar.

### E.2.11. Resuelto

```
svn resolved RUTA
```

### E.2.12. Renombrar

```
svn rename RUTA_ACTUAL NUEVA_RUTA
```

### E.2.13. Eliminar

```
svn delete RUTA
```

### E.2.14. Revertir

```
svn status -v RUTA
```

La primera fase es una comprobación del estado que determina los ítems en su copia de trabajo que pueden ser revertidos potencialmente. Puede revisar la lista, comparar los archivos contra BASE y seleccionar los ítems que desea incluir en la operación de revertir.

Cuando pulsa Aceptar, tiene lugar la operación de revertir en Subversion. Si ha dejado todas las casillas de selección de archivos sin tocar tal y como aparecieron, TortoiseSVN utilizará una única operación revertir recursivamente (-R) en la copia de trabajo. Si ha deseleccionado algún archivo, entonces se debe especificar individualmente cada ruta en la línea de comandos de la operación de revertir.

```
svn revert [-R] RUTA...
```

### E.2.15. Limpieza

```
svn cleanup RUTA
```

### E.2.16. Obtener bloqueo

```
svn status -v RUTA
```

La primera fase es una comprobación del estado que determina los archivos en su copia de trabajo que pueden ser potencialmente bloqueados. Puede seleccionar los ítems que desea bloquear.

```
svn lock -m "Mensaje de bloqueo" [--force] RUTA...
```

Mensaje de bloqueo aquí representa los contenidos del cuadro de texto Mensaje de bloqueo. Puede estar vacío.

Si se marca Robar los bloqueos, utilice la opción `--force`.

## E.2.17. Quitar bloqueo

```
svn unlock RUTA
```

## E.2.18. Rama/Etiqueta

```
svn copy -m "Mensaje de registro" URL URL
o
svn copy -m "Mensaje de registro" URL@rev URL@rev
o
svn copy -m "Mensaje de registro" RUTA URL
```

El diálogo Rama/Etiqueta hace una copia en el repositorio. Hay 3 botones de opciones:

- Revisión HEAD en el repositorio
- Revisión específica en el repositorio
- Copia de trabajo

que corresponden a las 3 variantes de línea de comandos anteriores.

Mensaje de registro aquí representa los contenidos del cuadro de texto Mensaje de Registro. Puede estar vacío.

## E.2.19. Cambiar

```
svn info URL_de_la_copia_de_trabajo
svn switch [-r rev] URL RUTA
```

## E.2.20. Fusionar

```
svn merge [--dry-run] --force From_URL@revN To_URL@revM RUTA
```

Probar fusión realiza la misma fusión con la opción `--dry-run`.

```
svn diff From_URL@revN To_URL@revM
```

Diff unificado muestra la operación de diferenciación que se utilizará para hacer la fusión.



### E.2.21. Exportar

```
svn export [-r rev] [--ignore-externals] URL RUTA_Exportación
```

Esta forma se utiliza cuando se accede desde una carpeta sin versionar, y la carpeta se utiliza como el destino.

La exportación de una copia de trabajo a otro lugar se hace sin utilizar las bibliotecas de Subversion, por lo que no hay línea de comandos equivalente.

Lo que TortoiseSVN hace es copiar todos los archivos al nuevo destino mientras le muestra el progreso de la operación. Opcionalmente, pueden exportarse también los archivos y/o carpetas sin versionar.

En ambos casos, si se marca Omitir externos, utilice la opción `--ignore-externals`.

### E.2.22. Relocalizar

```
svn switch --relocate URL_Origen URL_Destino
```

### E.2.23. Crear repositorio aquí

```
svnadmin create --fs-type fsfs RUTA
```

### E.2.24. Añadir

```
svn add RUTA...
```

Si había seleccionado una carpeta, en primer lugar TortoiseSVN la escanea recursivamente en busca de ítems que puedan ser añadidos.

### E.2.25. Importar

```
svn import -m "Mensaje de registro" RUTA URL
```

Mensaje de registro aquí representa los contenidos del cuadro de texto Mensaje de Registro. Puede estar vacío.

### E.2.26. Autoría

```
svn blame -r N:M -v RUTA  
svn log -r N:M RUTA
```

Si utiliza TortoiseBlame para ver la información de autoría, se necesita también el archivo de registro para mostrar los mensajes de registro en un texto de ayuda. Si visualiza la autoría como un archivo de texto, no se necesita esta información.

### E.2.27. Añadir a la lista de ignorados

```
svn propset svn:ignore RUTA > ficherotemporal  
{editar los nuevos ítems a ignorar en el archivo ficherotemporal}  
svn propset svn:ignore -F ficherotemporal RUTA
```

Dado que `svn:ignore` a menudo tiene un valor con varias líneas, aquí se muestra cómo se cambiaría utilizando un archivo de texto mejor que hacerlo directamente en la línea de comandos.

### E.2.28. Crear parche

```
svn diff RUTA > fichero-parche
```

TortoiseSVN crea un archivo de parche en formato diff unificado comparando la copia de trabajo con su versión BASE.

### E.2.29. Aplicar parche

Aplicar parches es un trabajo complicado a no ser que el parche y la copia de trabajo estén en la misma revisión. Afortunadamente, puede utilizar TortoiseMerge, que no tiene equivalente directo en Subversion.

---

# Apéndice F. Detalles de implementación

Este apéndice contiene una discusión más detallada sobre la implementación de algunas de las características de TortoiseSVN.

## F.1. Íconos superpuestos

Cada archivo y carpeta tiene un valor del estado de Subversion tal y como lo devuelve la librería de Subversion. En el cliente de línea de comandos, estos valores se representan por códigos de una única letra, pero en TortoiseSVN se muestran gráficamente utilizando los íconos superpuestos. Dado que el número de íconos superpuestos es muy limitado, cada ícono puede representar uno de varios valores de estado.



El ícono superpuesto *En conflicto* se utiliza para representar el estado en `conflicto`, donde una actualización o un cambio han resultado en conflictos entre los cambios locales y los cambios descargados desde el repositorio. También se utiliza para indicar el estado `obstruído`, que puede ocurrir cuando una operación no puede terminar.



El ícono superpuesto *Modificado* representa el estado `modificado`, donde ha hecho modificaciones locales; el estado `fusionado`, donde los cambios del repositorio se han fusionado con sus cambios locales; y el estado `reemplazado`, donde un archivo ha sido eliminado y reemplazado por otro diferente con el mismo nombre.



El ícono superpuesto *Eliminado* representa el estado `eliminado`, donde un ítem está marcado para su eliminación; o el estado `faltante`, donde un ítem no está presente. Naturalmente un ítem que está faltante no puede tener un ícono, pero la carpeta padre puede marcarse si uno de sus ítems hijo falta.



El ícono superpuesto *Añadido* se utiliza simplemente para representar el estado `añadido` cuando un ítem ha sido añadido al control de versiones.



El ícono superpuesto *En Subversion* se utiliza para representar un ítem que está en el estado `normal`, o un ítem versionado cuyo estado aún se desconoce. Dado que TortoiseSVN utiliza un proceso de caché en segundo plano para obtener los estados, puede llevar unos pocos segundos antes que se actualice el ícono superpuesto.



El *Necesidades de bloqueo emphasis> se utiliza sobreimpresi svn: needs-lock literal> conjunto de propiedades.*



El ícono superpuesto *Bloqueado* se utiliza cuando la copia de trabajo local contiene un bloqueo para ese archivo.



El ícono superpuesto *Ignorado* se utiliza para representar un ítem que está en el estado `ignorado`, bien por un patrón global de ignorado, o por la propiedad `svn:ignore` de la carpeta padre. Este ícono superpuesto es opcional.



El ícono superpuesto *No versionado* se utiliza para representar a un ítem que está en el estado no versionado. Este es un ítem dentro de una carpeta versionada, pero que él mismo no está bajo el control de versiones. Este ícono es 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.

Un ítem sólo puede tener un valor de estado de Subversion. Por ejemplo un archivo puede estar modificado localmente y puede estar marcado para su eliminación al mismo tiempo. Subversion devuelve un único valor de estado - en este caso *eliminado*. Estas prioridades están definidas en el propio Subversion.

Cuando TortoiseSVN muestra el estado recursivamente (la configuración por defecto), cada carpeta muestra un ícono superpuesto reflejando su propio estado y el estado de todos sus hijos. Para mostrar un único ícono *resumen*, utilizamos el orden de prioridad mostrado arriba para determinar cual utilizar, tomando *En conflicto* como la que tiene mayor prioridad.

De Hecho, es posible que no todos estos iconos se utilizan en el sistema. Esto es porque el número de sobreimpresiones permitidas por Windows está limitado a 15. Windows utiliza 4 de ellas, y las 11 restantes pueden ser utilizados por otras aplicaciones. Si no hay suficientes huecos de sobreimpresión disponibles, TortoiseSVN intenta ser un *Ciudadano Bueno (TM) emphasis> y limita su uso de sobreimpresiones para darles otras aplicaciones de una oportunidad.*

Puesto que hay clientes de Tortoise disponibles para otros sistemas de control de versiones, hemos creado un componente común que es responsable de mostrar los iconos. Los detalles técnicos no son importantes, todo lo que necesitas saber es que este componente compartido permite a todos los clientes de Tortoise emplear los mismos iconos y por lo tanto el límite de 11 slots disponibles no se usa al instalar más de un cliente de Tortoise. Por supuesto hay un pequeño inconveniente: todos los clientes de Tortoise usan los mismos iconos, por lo que no puedes averiguar por los iconos la versión del sistema de control que emplea una copia de trabajo.

- *Normal*, *Modificado* y *En conflicto* siempre se cargan y están visibles.
- *Borrado* se carga si es posible, pero se cambia por *Modificado* si no hay suficientes huecos.
- *Sólo-lectura* se carga si es posible, pero se cambia por *Normal* si no hay suficientes huecos.
- *Cerrado* se carga si es posible, pero vuelve a *Normal* si no hay suficientes huecos.
- *Añadido* se carga si es posible, pero vuelve a *Modificado* si no hay suficientes huecos.

---

# Apéndice G. Paquetes de Idiomas y Correctores Ortográficos

El instalador estándar solo soporta el idioma inglés, pero puede descargar paquetes de idiomas y diccionarios para la corrección ortográfica por separado después de la instalación.

## G.1. Paquetes de idiomas

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 paquete de idioma se encuentra como un instalador .msi. Simplemente ejecuta el programa de instalación y sigue las instrucciones. Cuando termine la instalación, la traducción estará disponible.

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. Corrector 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].

El instalador añade automáticamente los diccionarios de inglés de EE.UU. (US) y de Reino Unido (UK). Si desea otros idiomas, la opción más sencilla es instalar uno de los paquetes de idiomas de TortoiseSVN. Esto instalará los ficheros de diccionario adecuados, así como la interfaz de usuario de TortoiseSVN local. Una vez finalizada la instalación, el diccionario también estará listo para usar.

O también puede instalar los diccionarios usted mismo. Si tiene OpenOffice o Mozilla instalado, puede copiar estos diccionarios, que están ubicados en las carpetas de instalación de dichas aplicaciones. En caso contrario, necesitará descargar los archivos de diccionarios necesarios de <http://wiki.services.openoffice.org/wiki/Dictionaries> [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:

- es\_ES.aff
- es\_ES.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.

Si instala múltiples diccionarios, TortoiseSVN utilizará estas reglas para seleccionar cuál utilizar.

1. Comprobar la configuración `tsvn:projectlanguage`. Lea [Sección 4.18, “Configuración del proyecto”](#) para encontrar información sobre como se establecen las propiedades de proyecto.

2. Si no se ha establecido un idioma del proyecto, o ese idioma no está instalado, intenta con el idioma que corresponde al de Windows.
3. Si el idioma exacto de Windows no funciona, pruebe el “Base”, por ejemplo, es\_AR (Español Argentina) cae de nuevo a es (Español).
4. Si nada de lo anterior funciona, entonces el idioma por defecto es el Inglés, que se incluye con la instalación estándar.

---

# Glosario

Actualizar	Este comando de Subversion incorpora los últimos cambios del repositorio a su copia de trabajo, fusionando cualquier cambio hecho por otros con los cambios locales en la copia de trabajo.
Añadir	Un comando de Subversion que se utiliza para añadir un archivo o un directorio a su copia de trabajo. Los nuevos ítems se añaden al repositorio cuando confirme.
Autoría	Este comando es sólo para archivos de texto, y anota cada línea para mostrar la revisión del repositorio en la que se cambió por última vez, y el autor que hizo ese cambio. Nuestra implementación GUI se llama TortoiseBlame y también le muestra la fecha y hora de la confirmación y el mensaje de registro cuando mueve el ratón por encima del número de revisión.
Bloqueo	Cuando obtiene un bloqueo de un ítem versionado, lo marca en el repositorio como no confirmable, excepto para la copia de trabajo desde la que se obtuvo el bloqueo.
Cambiar	De la misma forma que “Actualizar-a-la-revisión” cambia la ventana temporal de una copia de trabajo para que quede como estuvo en un punto diferente de su historia, “Cambiar” cambia la ventana espacial de una copia de trabajo para que apunte a una parte diferente del repositorio. Es particularmente útil cuando esté trabajando en troncos y ramas en los que sólo hay unos pocos archivos diferentes. Puede cambiar su copia de trabajo entre los dos y sólo se transferirán los archivos cambiados.
Confirmar	Este comando de Subversion se utiliza para pasar los cambios desde su copia de trabajo local al repositorio, creando una nueva revisión en el repositorio.
Conflicto	Cuando los cambios desde el repositorio se mezclan con los cambios locales, a veces esos cambios ocurren en las mismas líneas. En este caso Subversion no puede decidir automáticamente qué versión utilizar, y se dice que el archivo está en conflicto. Tiene que editar el archivo manualmente y resolver el conflicto antes que pueda confirmar más cambios.
Copia de trabajo	Esta es su “caja de arena” local, el área donde trabaja con los archivos versionados, y normalmente se encuentra en su disco duro local. Puede crear una copia de trabajo al “Obtener” desde un repositorio, y devolver sus cambios al repositorio utilizando “Confirmar”.
Copiar	En un repositorio de Subversion puede crear una copia de un único archivo o un árbol completo. Estos se implementan como “copias baratas” que actúan un poco como un enlace al original en el sentido que casi no ocupan espacio. Haciendo una copia preserva la historia del ítem en la copia, por lo que puede rastrear los cambios que se hicieron antes que ocurriera la copia.
Diff	Abreviatura de “Mostrar diferencias”. Muy útil cuando desea ver exactamente qué cambios se han hecho.
Eliminar	Cuando elimina un ítem versionado (y confirma el cambio), el ítem ya no existirá nunca más en el repositorio después de la revisión confirmada. Pero por supuesto aún existe en las revisiones previas en el repositorio, por lo que aún puede acceder a él. Si es necesario, puede copiar un ítem eliminado y “resucitarlo” por completo con su historia.
Exportar	Este comando produce una copia de una carpeta versionada, igual que una copia de trabajo pero sin los directorios locales <code>.svn</code> .

FSFS	Un sistema de archivos propietario de Subversion que se utiliza como soporte de los repositorios. Se puede utilizar en unidades compartidas de red. Se utiliza por defecto para los repositorios creados a partir de la versión 1.2.
Fusionar	<p>El proceso por el cual los cambios del repositorio se añaden a su copia de trabajo sin interrumpir cualquier cambio que haya realizado ya localmente. A veces estos cambios no se pueden reconciliar automáticamente, y se dice que la copia de trabajo está en conflicto.</p> <p>Las fusiones ocurren automáticamente cuando actualiza su copia de trabajo. También puede fusionar cambios específicos desde otras ramas utilizando el comando Fusionar de TortoiseSVN.</p>
GPO	Objeto de políticas de grupo.
Historial	Muestra la historia de las revisiones de un archivo o carpeta. También se conoce como “Registro”.
Importar	Comando de Subversion para importar una jerarquía de carpetas completa en el repositorio en una única revisión.
Limpieza	Citando el libro de Subversion: “Limpia recursivamente la copia de trabajo, eliminando los bloqueos y continuando las operaciones sin terminar. Si alguna vez obtiene un error copia de trabajo bloqueada, ejecute este comando para eliminar los bloqueos y conseguir que su copia de trabajo vuelva a un estado usable de nuevo.” Tenga en cuenta que en este contexto “bloqueado” se refiere a un bloqueo del sistema de archivos local, no a bloqueos del repositorio.
Obtener	Un comando de Subversion que crea una copia local de trabajo en un directorio vacío al descargar los archivos versionados desde el repositorio.
Parche	Si una copia de trabajo tiene cambios únicamente en archivos de texto, es posible utilizar el comando Diff de Subversion para generar un único archivo con el resumen de esos cambios en el formato de Diff unificado. Un archivo de este tipo a menudo se denomina “Parche”, y puede ser enviado por email a otro (o a una lista de correo) y aplicado en otra copia de trabajo. Alguien sin acceso de confirmación puede hacer cambios y enviar un archivo de parche para que lo aplique un confirmador autorizado. O si no está seguro de un cambio, puede enviar el parche a otros para que lo revisen.
Propiedad	Además de versionar sus directorios y archivos, Subversion le permite añadir metainformación versionada - llamadas “propiedades” para cada uno de sus directorios y archivos versionados. Cada propiedad tiene un nombre y un valor, como un entrada del registro. Subversion tiene algunas propiedades especiales que utiliza internamente, como <code>svn:eol-style</code> . TortoiseSVN también <code>svn:logminsize</code> . Puede a
Propiedad de revisión (revprop)	Al igual que los archivos pueden tener propiedades, las revisiones del repositorio también. Algunas revprops especiales se añaden automáticamente cuando se crea la revisión; son: <code>svn:date</code> <code>svn:author</code> <code>svn:log</code> que representan la fecha/hora de la confirmación, el confirmador y el mensaje de registro, respectivamente. Estas propiedades pueden editarse, pero no están versionadas, por lo que cualquier cambio es permanente y no puede deshacerse.
Rama	Un término de uso frecuente en los sistemas de control de versiones para describir lo que ocurre cuando el desarrollo se parte en un punto en concreto y sigue dos caminos separados. Puede crear una rama desde la línea principal de desarrollo para desarrollar una nueva funcionalidad sin hacer que la línea



principal quede inestable. O puede hacer una rama de una versión estable en la que sólo hará correcciones de errores, mientras los nuevos desarrollos tienen lugar en el inestable tronco. En Subversion una rama se implementa como una “copia barata”.

Registro	Muestra la historia de las revisiones de un archivo o carpeta. También se conoce como “Historial”.
Relocalizar	Si su repositorio se mueve, quizás porque lo ha movido a un directorio diferente en su servidor, o el nombre de dominio del servidor ha cambiado, necesitará “relocalizar” su copia de trabajo para que sus URLs de repositorio apunten al nuevo lugar.  Nota: sólo debería utilizar este comando si su copia de trabajo apunta al mismo lugar en el mismo repositorio, pero es el propio repositorio el que se ha movido. En cualquier otra circunstancia probablemente necesita el comando “Cambiar” en su lugar.
repositorio	Un repositorio es un lugar central donde se almacenan y mantienen los datos. Un repositorio puede ser un lugar donde se encuentran múltiples bases de datos o archivos para distribuirlos en una red, o un repositorio puede ser un lugar directamente accesible por el usuario sin tener que viajar por una red.
Resolver	Cuando los archivos en una copia de trabajo se quedan en un estado de conflicto tras una fusión, estos conflictos deben arreglarse por un humano utilizando un editor (o quizás TortoiseMerge). Este proceso se denomina “Resolviendo conflictos”. Cuando finalice, marque los archivos con conflictos como resueltos, y eso le permitirá confirmarlos.
Revertir	Subversion mantiene una copia local “prístina” de cada archivo tal y como estaban cuando actualizó su copia de trabajo por última vez. Si ha hecho cambios y decide que quiere deshacerlos, puede utilizar el comando “revertir” para volver a la copia prístina.
Revisión	Cada vez que confirma un conjunto de cambios, crea una nueva “revisión” en el repositorio. Cada revisión representa el estado del árbol del repositorio en un cierto punto de su historia. Si desea volver en el tiempo para examinar el repositorio tal y como estaba en la revisión N.  En otro sentido, una revisión puede referirse al conjunto de cambios que se hicieron cuando esa revisión se creó.
revisión BASE	La revisión base actual de un archivo o una carpeta en su <i>copia de trabajo</i> . Esta es la revisión en la que estaba el archivo o la carpeta cuando se hizo la última operación de obtener, actualizar o confirmar. La revisión BASE normalmente no equivale a la revisión HEAD.
revisión HEAD	La última revisión de un archivo o una carpeta en el <i>repositorio</i> .
SVN	Una abreviatura muy usada para Subversion.  El nombre del protocolo personalizado de Subversion que utiliza el servidor de repositorios “svnserve”.

---

# Índice

## Símbolos

'copia de trabajo' no versionada, 129

## A

Acceso, 17  
acceso directo, 198  
acciones en el lado del servidor, 122  
actualizar, 38, 195  
alabanza, 119  
añadir, 74  
añadir archivos al repositorio, 26  
anotar, 119  
archivos especiales, 28  
archivos temporales, 26  
archivos/carpetas no versionados, 75  
arrastrar con el botón derecho, 24  
arrastrar-y-soltar, 24  
autenticación, 25  
auto-props, 85  
automatización, 204, 210, 211, 211  
autoría, 119

## B

bloqueando, 114

## C

caché de autenticación, 25  
caché de registro, 162  
cambiar, 105  
cambios, 197  
CLI, 213  
click con el botón derecho, 22  
cliente de línea de comandos, 213  
COM, 181, 189  
commit monitor, 178  
comparar, 68  
comparar archivos, 197  
comparar carpetas, 197  
comparar revisiones, 70  
comprobación de actualización, 200  
comprobar nueva versión, 200  
concordancia de patrones, 76  
configuración, 138  
confirmar, 31, 31  
conflicto, 10, 40  
conflicto de árbol, 40  
conflictos de fusión, 112  
Consola de Windows, xi  
control de errores, 132  
control de versiones, xi  
controlador de dominio, 200  
copia de seguridad, 19  
copia de trabajo, 11

copiar, 102, 122  
copiar archivos, 74  
corrector ortográfico, 222  
Crear, 16  
    TortoiseSVN, 16  
crear copia de trabajo, 28  
crear repositorio, 16

## D

desenganchar del repositorio, 198  
deshabilitar funciones, 201  
deshacer, 80, 196  
deshacer cambio, 196  
deshacer confirmación, 196  
despliegue, 200  
desprotección parcial, 28  
desversionar, 131, 198  
diccionario, 222  
diferenciando, 49  
diferenciar, 68, 117  
diferenciar imágenes, 72  
diff unificado, 117

## E

editar registro/autor, 63  
eliminar, 78, 78  
eliminar versionado, 198  
enlace, 20  
enlace de obtener, 20  
entradas de menú contextual, 201  
enviar cambios, 31  
estadísticas, 65  
estado, 44, 46  
estado de la copia de trabajo, 44  
etiqueta, 74, 102  
expandir palabras claves, 83  
expansión de comodines, 76  
explorador, xi  
exportar, 129  
exportar cambios, 70  
extensión, 189  
externos, 99, 197  
extracción de versión, 181

## F

FAQ, 194  
filtro, 63  
fusionar, 106  
    dos árboles, 109  
    rango de revisiones, 107

## G

ganchos, 19  
ganchos cliente, 165  
GPO, 200  
gráfico, 125  
gráfico de revisión, 125

## H

herramientas de diferenciación, 73  
herramientas de fusión, 73  
historial, 53  
historial de registro de fusión, 62

## I

IBugtraqProvider, 189  
íconos, 44  
ignorar, 75  
ignorar global, 139  
importar, 26  
importar en el sitio, 27  
instalación, 1  
interfaz COM de SubWCRev, 186

## L

Libro de Subversion, 7  
limpiar, 80  
limpieza, 82  
línea de comandos, 204, 211, 212  
lista de cambios, 49

## M

manejador de arrastre, 24  
Manejador de URL, 210  
marcar una entrega, 102  
maximizar, 26  
mensaje de registro, 195, 195  
mensaje vacío, 195  
mensajes de confirmación, 53  
mensajes de registro, 53  
Menú contextual, 22  
Microsoft Word, 73  
modificaciones, 46  
monitoring projects, 178  
mover, 79  
mover archivos, 74  
mover el servidor, 131  
moviendo, 195  
msi, 200

## N

navegador de repositorios, 122  
número de versión en los archivos, 181

## O

obtención dispersa, 28  
obtener, 28  
obtener cambios, 38

## P

palabras claves, 83  
paquetes de idioma, 222  
parche, 117  
patrón de ignorar, 139  
políticas de grupo, 200, 201

prioridad de superposición, 220  
project monitor, 178  
propiedades de la revisión, 63  
propiedades de proyecto, 86  
Propiedades de Subversion, 83  
Propiedades de TortoiseSVN, 86  
propiedades de windows, 45  
proyectos ASP, 201  
proyectos comunes, 197  
proyectos de vendedores, 197

## R

rama, 74, 102  
registro, 53, 172  
registro de fusión, 111  
relocalizar, 131  
remote commits, 178  
renombrar, 79, 122, 195  
renombrar archivos, 74  
reorganizar, 195  
repositorio, 7, 26  
repositorios externos, 99  
resolver, 40  
revertir, 80, 196  
revisión, 13, 125  
revprops, 63  
rutas UNC, 17

## S

scripts gancho, 19, 165  
scripts gancho en el lado del servidor, 19  
seguimiento de bugs, 132  
seguimiento de incidencias, 132, 189  
seguir de bugs, 132  
servidor movido, 131  
servidor proxy, 154  
shelve, 51  
sitio web, 20  
sólo lectura, 114  
sonidos, 138  
SubWCRev, 181  
Superposiciones, 44, 220  
SVN\_ASP\_DOT\_NET\_HACK, 201

## T

TortoiseIDiff, 72  
traducciones, 222

## U

Unidad de red, 17  
unidades SUBST, 152  
unshelve, 51  
URL cambiada, 131  
URL del repositorio cambiada, 131

## V

ver cambios, 44

versión, 200  
versionar archivos nuevos, 74  
ViewVC, 137  
vínculo TortoiseSVN, 20  
visor de repositorios, 137  
visor de servidores, 122  
vista web, 137  
VS2003, 201

## **W**

WebSVN, 137