

# **TortoiseSVN**

**Một máy khách Subversion cho Windows**

**Version 1.8**

**Küng, Stefan  
Onken, Lübbe  
Large, Simon**

---

# **TortoiseSVN: Một máy khách Subversion cho Windows: Version 1.8**

bởi Küng, Stefan, Onken, Lübbe, và Large, Simon  
người phiên dịch-bản dịch:

Publication date 2013/11/11 21:33:57 (r24944)

---

---

# Mục lục

Lời nói đầu .....	xi
1. TortoiseSVN là gì? .....	xi
2. Các tính năng của TortoiseSVN .....	xi
3. Giấy phép .....	xii
4. Phát triển .....	xii
4.1. Lịch sử của TortoiseSVN .....	xii
4.2. Lời cảm ơn .....	xiii
5. Hướng dẫn đọc .....	xiii
6. Thuật ngữ sử dụng trong tài liệu này .....	xiv
1. Bắt đầu .....	1
1.1. Cài đặt TortoiseSVN .....	1
1.1.1. Yêu cầu hệ thống .....	1
1.1.2. Cài đặt .....	1
1.2. Các khái niệm cơ bản .....	1
1.3. Đi lái thử .....	2
1.3.1. Tạo một kho lưu trữ .....	2
1.3.2. Nhập khẩu một dự án .....	2
1.3.3. Kiểm xu một bản sao làm việc .....	3
1.3.4. Thực hiện Thay đổi .....	3
1.3.5. Thêm nhiều tập tin .....	4
1.3.6. Xem Lịch sử dự án .....	4
1.3.7. Hoàn tác thay đổi .....	5
1.4. Đi tiếp .....	5
2. Khái niệm kiểm soát Phiên bản cơ bản .....	7
2.1. Kho lưu trữ .....	7
2.2. Mô hình Phiên bản .....	7
2.2.1. Vấn đề chia sẻ tập tin .....	8
2.2.2. Giải pháp Khóa-Sửa-Mở khóa .....	8
2.2.3. Giải pháp Sao chép-Sửa đổi-Hợp nhất .....	9
2.2.4. Subversion làm gì? .....	11
2.3. Subversion trong hành động .....	11
2.3.1. Bản sao Làm việc .....	11
2.3.2. URL kho chứa .....	12
2.3.3. Sửa đổi .....	13
2.3.4. Làm thế nào Các Bản Sao Làm Việc theo dõi các Kho .....	15
2.4. Tóm tắt .....	15
3. Kho lưu trữ .....	16
3.1. Tạo kho lưu trữ .....	16
3.1.1. Tạo một Repository với các dòng lệnh máy khách .....	16
3.1.2. Tạo Kho Lưu Trữ Với TortoiseSVN .....	16
3.1.3. Truy cập địa phương đến Kho .....	17
3.1.4. Truy cập vào một kho lưu trữ trên một mạng chia sẻ .....	17
3.1.5. Bố cục Kho chứa .....	18
3.2. Sao lưu kho lưu trữ .....	19
3.3. kịch bản móc phía máy chủ .....	20
3.4. Các Liên kết Kiểm xuất .....	20
3.5. Truy cập Kho .....	21
4. Hướng dẫn sử dụng hàng ngày .....	22
4.1. Các Tính Năng Chung .....	22
4.1.1. Biểu tượng Lớp phủ .....	22
4.1.2. Tr đơn ngữ cảnh .....	22
4.1.3. Kéo và thả .....	24
4.1.4. Phím tắt phổ biến .....	25
4.1.5. Xác thực .....	25
4.1.6. Tối đa hoá cửa sổ .....	26

4.2. Nhập khẩu dữ liệu vào một kho chứa .....	26
4.2.1. Nhập khẩu .....	26
4.2.2. Nhập khẩu Tại chỗ .....	28
4.2.3. Các tập tin Đặc biệt .....	28
4.3. Kiểm Xuất Một Bản Sao Làm Việc .....	29
4.3.1. Độ sâu Kiểm xu .....	29
4.4. Cam kết các Thay đổi của bạn vào Kho .....	31
4.4.1. Hộp Thoại Cam kết .....	31
4.4.2. Danh sách Thay đổi .....	34
4.4.3. Chỉ cam kết các phần của tập tin .....	34
4.4.4. Loại trừ mục khỏi Danh mục Cam kết .....	34
4.4.5. Thông Điệp Tường Trình Cam Kết .....	34
4.4.6. Tiên độ Cam kết .....	36
4.5. Cập nhật bản sao làm việc của bạn với thay đổi từ những người khác .....	37
4.6. Giải quyết mâu thuẫn .....	38
4.6.1. Các xung đột tập tin .....	39
4.6.2. Xung đột về thuộc tính .....	40
4.6.3. Mâu Thuẫn Cây .....	40
4.7. Lấy Thông tin Trạng thái .....	43
4.7.1. Biểu tượng Lớp phủ .....	43
4.7.2. Tình trạng Chi tiết .....	44
4.7.3. Cột TortoiseSVN Trong Windows Explorer .....	45
4.7.4. Trạng thái địa phương và từ xa .....	46
4.7.5. Xem các khác biệt .....	48
4.8. Danh sách Thay đổi .....	49
4.9. Hộp thoại Tường trình Chính sửa .....	51
4.9.1. Gọi Hộp thoại Tường Trình Sửa Đổi .....	52
4.9.2. Hoạt động Tường Trình Sửa Đổi .....	52
4.9.3. Lấy thông tin thêm .....	53
4.9.4. Lấy thêm thông điệp tường trình .....	59
4.9.5. Bản sửa đổi làm việc hiện tại .....	59
4.9.6. Hợp nhất tính năng theo dõi .....	60
4.9.7. Thay đổi Thông Điệp Tường Trình và Tác giả .....	60
4.9.8. Lọc Thông Điệp Tường Trình .....	61
4.9.9. Thông Tin Thông kê .....	63
4.9.10. Chế Độ Ngoại Tuyển .....	66
4.9.11. Làm mới màn hình xem .....	66
4.10. Xem Các Khác Biệt .....	66
4.10.1. Khác biệt của tập tin .....	67
4.10.2. Tùy chọn cho Hết-Dòng và Khoảng Trắng .....	68
4.10.3. So sánh các thư mục .....	68
4.10.4. Tìm Khác Biệt Hình Ảnh Sử Dụng TortoiseIDiff .....	70
4.10.5. Tìm khác biệt các tài liệu văn phòng .....	71
4.10.6. Công cụ Khác Biệt / Hợp Nhất Bên ngoài .....	71
4.11. Thêm Tập Tin Và Thư Mục Mới .....	72
4.12. Sao chép / di chuyển / Đổi tên tập tin và thư mục .....	72
4.13. Bỏ qua các Tập Tin và Thư mục .....	73
4.13.1. Việc Khớp các Mẫu trong Danh sách Bỏ qua .....	74
4.14. Xóa, Di chuyển và đổi tên .....	75
4.14.1. Xóa các tập tin và thư mục .....	76
4.14.2. Di chuyển tập tin và thư mục .....	77
4.14.3. Đối phó với các cuộc xung đột kiểu chữ tên tập tin .....	77
4.14.4. Sửa chữa đổi tên tập tin .....	78
4.14.5. Xóa các tập tin chưa phiên bản .....	78
4.15. Hoàn tác các thay đổi .....	78
4.16. Dọn sạch .....	80
4.17. Thiết lập Dự án .....	80
4.17.1. Thuộc tính Subversion .....	80

4.17.2. Thuộc tính dự án TortoiseSVN .....	84
4.17.3. Bộ Biên tập Thuộc tính .....	86
4.18. Mục bên ngoài .....	92
4.18.1. Thư mục bên ngoài .....	92
4.18.2. Các tập tin bên ngoài .....	94
4.19. Phn Nhánh / Tạo Thẻ .....	94
4.19.1. Tạo một Chi nhánh hoặc Thẻ .....	95
4.19.2. Các cách khác để tạo ra một chi nhánh hoặc thẻ .....	97
4.19.3. Kiểm xu hoặc Chuyển mạch .....	97
4.20. Hợp nhất .....	98
4.20.1. Hợp nhất một Phạm vi Các sửa đổi .....	99
4.20.2. Việc Sáp Nhập Hai Cây Khác Nhau .....	101
4.20.3. Tùy Chọn Hợp Nhất .....	102
4.20.4. Rà soát các Kết quả Hợp nhất .....	103
4.20.5. Theo dõi hợp nh .....	103
4.20.6. Xử Lý Xung Đột trong Hợp Nhất .....	104
4.20.7. Hợp nhất một Chi nhánh Đây đủ .....	105
4.20.8. Bảo trì các Chi nhánh Tính năng .....	105
4.21. Khóa .....	106
4.21.1. Khóa làm việc thế nào trong Subversion .....	106
4.21.2. Lấy một khóa .....	107
4.21.3. Phát hành một khóa .....	108
4.21.4. Kiểm tra trạng thái Khóa .....	108
4.21.5. Làm các Tập tin không bị Khóa Chi đọc .....	109
4.21.6. Các kịch bản Hook khóa .....	109
4.22. Tạo và áp dụng các bản vá .....	109
4.22.1. Tạo một tập tin v .....	109
4.22.2. Áp dụng một tập tin v .....	110
4.23. Ai Đã Thay Đổi Dòng Nào? .....	111
4.23.1. Đồ Lỗi vì Tập Tin .....	112
4.23.2. Đồ lỗi cho khác biệt .....	114
4.24. Trình duyệt Kho .....	114
4.25. Các đồ thị sửa đổi .....	117
4.25.1. Các Nút Đồ thị Sửa đổi .....	118
4.25.2. Thay đổi cách xem .....	119
4.25.3. Sử dụng Đồ thị .....	120
4.25.4. Làm mới màn hình xem .....	121
4.25.5. Cắt tia các cây .....	121
4.26. Xuất khẩu một bản sao làm việc Subversion .....	122
4.26.1. Loại bỏ một bản sao làm việc từ kiểm soát phiên bản .....	123
4.27. Di dời một bản sao làm việc .....	123
4.28. Tích hợp với Hệ thống Theo dõi Lỗi / Trình theo dõi Vấn đ .....	124
4.28.1. Thêm số Vấn đề vào Thông điệp Tường tr .....	125
4.28.2. Lấy Thông tin từ Bộ Theo Dõi Vấn Đề .....	128
4.29. Tích hợp với các Trình Xem Kho Lưu Trữ dựa trên Web .....	129
4.30. Các thiết l của TortoiseSVN .....	130
4.30.1. Các Thiết Lập Tổng Qu .....	130
4.30.2. Các Thiết Lập Cho Đồ thị Sửa Đổi .....	139
4.30.3. Các Thiết Lập Cho Lốp Phủ Biểu Tượng .....	142
4.30.4. Các Thiết Lập Mạng .....	145
4.30.5. Các Thiết Lập Cho Chương Trình Bên Ngoài .....	147
4.30.6. Thiết Lập Cho Dữ Liệu Đã Lưu .....	151
4.30.7. Lưu Đệm Tường Trình .....	152
4.30.8. Kịch Bản Hook Phía Máy Khách .....	155
4.30.9. Thiết Lập TortoiseBlame .....	160
4.30.10. Thiết Lập Nâng Cao .....	160
4.30.11. Các Thiết Lập Xuất Khẩu TSVN .....	165
4.31. Bước Cuối cùng .....	165

5. Chương trình SubWCRev .....	166
5.1. Dòng Lệnh SubWCRev .....	166
5.2. Thay Thế Từ Khoá .....	166
5.3. Ví dụ Từ khoá .....	167
5.4. giao diện COM .....	168
6. giao diện IBugtraqProvider .....	171
6.1. Quy ước đặt tên .....	171
6.2. Giao diện IBugtraqProvider .....	171
6.3. Giao diện IBugtraqProvider2 .....	173
A. Các Câu Hỏi Thường Gặp (FAQ) .....	176
B. Làm thế nào để tôi .....	177
B.1. Di chuyển / sao chép nhiều tập tin cùng một lúc .....	177
B.2. Buộc người dùng phải nhập một thông điệp tường trình .....	177
B.2.1. Kích bản Hook trên máy chủ .....	177
B.2.2. Các thuộc tính dự án .....	177
B.3. Cập nhật các tập tin được lựa chọn từ kho .....	177
B.4. Phục hồi (Hoàn tác) các sửa đổi trong kho .....	178
B.4.1. Sử dụng hộp thoại tường trình sửa đổi .....	178
B.4.2. Sử dụng hộp thoại hợp nhất .....	178
B.4.3. Sử dụng <code>svndumpfilter</code> .....	178
B.5. So sánh hai phiên bản của một tập tin hoặc thư mục .....	179
B.6. Bao gồm các tiêu dự án chung .....	179
B.6.1. Sử dụng <code>svn:externals</code> .....	179
B.6.2. Sử dụng một bản sao làm việc lồng nhau .....	179
B.6.3. Sử dụng một vị trí tương đối .....	179
B.6.4. Add the project to the repository .....	180
B.7. Tạo một phím tắt đến một kho lưu trữ .....	180
B.8. Bỏ qua các tập tin được đã được phiên bản .....	180
B.9. Bỏ phiên bản một bản sao hoạt động .....	181
B.10. Loại bỏ một bản sao hoạt động .....	181
C. Các Lời Khuyên Hữu Ích Cho Các Quản Trị .....	182
C.1. Triển khai TortoiseSVN thông qua các chính sách nhóm .....	182
C.2. Chuyển hướng kiểm tra nâng cấp .....	182
C.3. Thiết lập biến môi trường <code>SVN_ASP_DOT_NET_HACK</code> .....	183
C.4. Vô hiệu hoá các mục trình đơn ngữ cảnh .....	183
D. Tự động hoá TortoiseSVN .....	185
D.1. Các lệnh TortoiseSVN .....	185
D.2. xử lý <code>Tsvncmd</code> URL .....	186
D.3. Các lệnh <code>TortoiseIDiff</code> .....	187
E. Tham Chiếu Chéo Giao Diện Dòng Lệnh .....	188
E.1. Các quy ước và quy tắc cơ bản .....	188
E.2. Các lệnh TortoiseSVN .....	188
E.2.1. Kiểm xuất .....	188
E.2.2. Cập nhật .....	188
E.2.3. Cập nhật đến bản sửa đổi .....	189
E.2.4. Cam kết .....	189
E.2.5. Tìm Khác Biệt .....	189
E.2.6. Hiện Tường Trình .....	190
E.2.7. Kiểm tra xem có sửa đổi .....	190
E.2.8. Đồ Thị Sửa Đổi .....	190
E.2.9. Trình Duyệt Kho Lưu Trữ .....	190
E.2.10. Chỉnh sửa xung đột .....	190
E.2.11. Đã giải quyết .....	190
E.2.12. Đổi tên .....	191
E.2.13. Xóa .....	191
E.2.14. Hoàn nguyên .....	191
E.2.15. Dọn sạch .....	191
E.2.16. Nhận Khóa .....	191

---

E.2.17. Thả Khóa .....	191
E.2.18. Nhánh/Thẻ .....	191
E.2.19. Chuyên .....	192
E.2.20. Hợp nhất .....	192
E.2.21. Xuất khẩu .....	192
E.2.22. Di dời .....	192
E.2.23. Tạo Kho Chứa Ổ Đầy .....	192
E.2.24. Thêm .....	192
E.2.25. Nhập khẩu .....	193
E.2.26. Đồ lỗi .....	193
E.2.27. Thêm vào Danh sách Bỏ qua .....	193
E.2.28. Tạo Bản Vá .....	193
E.2.29. Áp dụng Bản vá .....	193
F. Thông tin chi tiết Thực hiện .....	194
F.1. Biểu tượng Lốp phủ .....	194
G. Các gói ngôn ngữ và bộ kiểm tra chính tả .....	196
G.1. Gói Ngôn Ngữ .....	196
G.2. Bộ kiểm tra chính tả .....	196
Thuật ngữ .....	198
Chỉ mục .....	201

---

## Danh sách Hình

1.1. Các trình đơn TortoiseSVN cho các thư mục không phiên bản .....	2
1.2. Hộp thoại nhập khẩu .....	3
1.3. Trình xem khác biệt của tập tin .....	4
1.4. Hộp thoại Tường Trình .....	5
2.1. Một hệ thống Máy Khách/Máy Chủ điển hình .....	7
2.2. Vấn đề cần tránh .....	8
2.3. Giải pháp Khóa-Sửa-Mở khóa .....	9
2.4. Giải pháp Sao chép-Sửa đổi-Hợp nhất .....	10
2.5. ... Sao chép-Sửa đổi-Hợp nhất Tiếp theo .....	10
2.6. Các hệ thống tập tin của kho .....	12
2.7. Kho lưu trữ .....	14
3.1. Các trình đơn TortoiseSVN cho các thư mục không phiên bản .....	16
4.1. Explorer hiển thị lớp phủ biểu tượng .....	22
4.2. Trình đơn ngữ cảnh cho một thư mục dưới sự kiểm soát phiên bản .....	23
4.3. Trình đơn explorer tập tin cho một phím tắt trong một thư mục được phiên bản .....	24
4.4. Kéo phải trình đơn cho một thư mục dưới sự kiểm soát phiên bản .....	25
4.5. Hộp thoại xác thực .....	26
4.6. Hộp thoại nhập khẩu .....	27
4.7. Hộp thoại Kiểm Xu .....	29
4.8. Hộp thoại Cam kết .....	32
4.9. Bộ kiểm tra chính tả của Hộp Thoại Cam Kết .....	35
4.10. Hộp thoại tiến độ thể hiện một cam kết đang tiến triển .....	36
4.11. Hộp thoại tiến độ hiển thị cập nhật hoàn thành .....	37
4.12. Explorer hiển thị lớp phủ biểu tượng .....	43
4.13. trang thuộc tính Explorer, tab Subversion .....	45
4.14. Kiểm tra xem có sửa đổi .....	46
4.15. Hộp thoại cam kết với danh sách thay đổi .....	50
4.16. Hộp thoại Tường Trình Sửa Đổi .....	52
4.17. Bảng trên cùng của hộp thoại tường trình sửa đổi với trình đơn ngữ cảnh .....	53
4.18. The Code Collaborator Settings Dialog .....	56
4.19. Bảng trên cùng của trình đơn ngữ cảnh trong 2 bản sửa đổi được lựa chọn .....	56
4.20. Khung hộp thoại tường trình dưới đây với trình đơn ngữ cảnh .....	57
4.21. The Log Dialog Bottom Pane with Context Menu When Multiple Files Selected. ....	58
4.22. Hộp Thoại Tường Trình Hiển Thị Theo Dõi Hợp Nhất Các Bản Sửa Đổi .....	60
4.23. Biểu đồ phân phối của các Cam kết-bởi-Tác giả .....	63
4.24. Biểu đồ hình bánh của các Cam kết-bởi-Tác giả .....	64
4.25. Đồ thị Cam kết-theo-ngày .....	65
4.26. Đến Hộp Thoại Ngoại Tuyển .....	66
4.27. Hộp Thoại So Sánh Bản Sửa Đổi .....	69
4.28. Trình xem khác biệt hình ảnh .....	70
4.29. Trình đơn ngữ cảnh Explorer cho các tập tin không phiên bản .....	72
4.30. Kéo phải trình đơn cho một thư mục dưới sự kiểm soát phiên bản .....	73
4.31. Trình đơn ngữ cảnh Explorer cho các tập tin không phiên bản .....	74
4.32. Trình đơn ngữ cảnh của Explorer cho các tập tin được phiên bản .....	76
4.33. Hộp thoại hoàn nguyên .....	79
4.34. Trang thuộc tính Subversion .....	81
4.35. Thêm thuộc tính .....	82
4.36. Trang thuộc tính svn: externals .....	86
4.37. Trang thuộc tính svn:keywords .....	87
4.38. Trang thuộc tính svn:eol-style .....	87
4.39. Trang thuộc tính tsvn:bugtraq .....	88
4.40. Kích thước của trang thuộc tính thông điệp tường trình .....	89
4.41. Trang thuộc tính ngôn ngữ .....	89
4.42. trang thuộc tính svn:mime-type .....	90
4.43. trang thuộc tính svn:needs-lock .....	90



4.44. trang thuộc tính svn:executable .....	90
4.45. Các mẫu thông điệp tường trình cho hộp thoại thuộc tính .....	91
4.46. Hộp tho Chi nhánh / Thẻ .....	95
4.47. Hộp thoại Chuyên ồi .....	98
4.48. Trình hướng dẫn Hợp nh - Chọn Phạm vi Sửa đổi .....	99
4.49. Trình hướng dẫn sáp nh - Sáp nhập cây .....	101
4.50. Hộp Thoại Gọi Lại Xung Đột Hợp Nhất .....	104
4.51. Hộp thoại tái tích hợp sáp nh .....	105
4.52. Hộp thoại khóa .....	107
4.53. Hộp thoại Kiểm tra Sửa đổi .....	108
4.54. Hộp thoại Tạo Bản Vá .....	110
4.55. Chú Thích / Hộp Thoại Đồ Lỗi .....	112
4.56. TortoiseBlame .....	113
4.57. Trình duyệt Kho .....	115
4.58. Một Đồ thị Sửa đổi .....	117
4.59. Hộp thoại xuất khẩu từ URL .....	122
4.60. Hộp Tho Di Dời .....	123
4.61. Hội Thoại Thuộc Tính Bugtraq .....	125
4.62. Hộp thoại ví dụ truy vấn theo dõi vấn đề .....	129
4.63. Hộp Thoại Thiết Lập, Trang Tổng Quát .....	131
4.64. Hộp Thoại Thiết Lập, Trang Trình Đơn Ngữ Cảnh .....	133
4.65. Trang Hộp Thoại Thiết Lập, Các Hộp Thoại 1 .....	134
4.66. Trang Hộp Thoại Thiết Lập, Hộp Thoại 2 .....	136
4.67. Hộp Thoại Thiết Lập, Trang Hộp Thoại 3 .....	137
4.68. Trang Hộp Thoại Thiết Lập, Màu .....	138
4.69. Hộp Thoại Thiết Lập, Trang Biểu Đồ Sửa Đổi .....	139
4.70. Hộp Thoại Thiết Lập, Trang Màu Sắc Đồ Thị Sửa Đổi .....	140
4.71. Hộp Thoại Thiết Lập, Trang Lớp Phủ Biểu Tượng .....	142
4.72. Hộp Thoại Thiết Lập, Trang Tập Hợp Biểu Tượng .....	144
4.73. Hộp Thoại Thiết Lập, Trang Xử Lý Biểu Tượng .....	145
4.74. Hộp Thoại Thiết Lập, Trang Mạng .....	146
4.75. Hộp Thoại Thiết Lập, Trang Xem Khác Biệt .....	147
4.76. Hộp thoại Thiết Lập, Hộp Thoại Khác Biệt/Hợp Nhất Nâng Cao .....	150
4.77. Hộp Thoại Thiết Lập, Trang Đã Lưu Dữ Liệu .....	151
4.78. Hộp Thoại Thiết Lập, Trang bộ nhớ đệm tường trình .....	152
4.79. Hộp Thoại Thiết Lập, Thống kê Bộ Nhớ Đệm Lưu Trữ Tường Trình .....	154
4.80. Hộp Thoại Thiết Lập, Trang Kích Bản Hook .....	155
4.81. Hộp Thoại Thiết Lập, Cấu Hình Kích Bản Hook .....	156
4.82. Hộp Thoại Thiết Lập, Trang Tích Hợp Bộ Theo Dõi Vấn Đề .....	159
4.83. Hộp Thoại Thiết Lập, Trang TortoiseBlame .....	160
4.84. Thanh tác vụ với cách nhóm mặc định .....	162
4.85. Thanh tác vụ với cách nhóm kho .....	162
4.86. Thanh tác vụ với cách nhóm kho .....	163
4.87. Nhóm thanh tác vụ với lớp phủ màu cho kho lưu trữ .....	163
C.1. Hộp thoại cam kết, hiển thị thông báo nâng cấp .....	182

---

## Danh sách Bảng

2.1. URL truy cập kho lưu trữ .....	13
5.1. Danh sách các thiết bị chuyên mạch dòng lệnh có sẵn .....	166
5.2. List of SubWCRev error codes .....	166
5.3. Danh sách các từ khóa có sẵn .....	166
5.4. các phương pháp COM / tự động hóa được hỗ trợ .....	168
C.1. Các mục trong trình đơn và các giá trị của chúng .....	183
D.1. Danh sách các lệnh có thể dùng và các tùy chọn .....	185
D.2. Danh sách các tùy chọn có sẵn .....	187

---

# Lời nói đầu



TortoiseSVN

Kiểm soát phiên bản là nghệ thuật quản lý thay đổi thông tin. Nó từ lâu đã là một công cụ quan trọng cho các lập trình viên, người thường dành thời gian của họ thực hiện những thay đổi nhỏ trên phần mềm và sau đó hoàn tác hoặc kiểm tra một số những thay đổi vào ngày hôm sau. Hãy tưởng tượng một đội ngũ các nhà phát triển làm việc đồng thời - và thậm chí có thể đồng thời trên cùng các tập tin! - Và bạn có thể thấy lý do tại sao một hệ thống tốt là cần thiết để *quản lý sự hỗn loạn tiềm tàng* .

## 1. TortoiseSVN là gì?

TortoiseSVN là một trình máy khách Windows mã nguồn mở miễn phí cho hệ thống kiểm soát phiên bản *Apache™ Subversion®* . TortoiseSVN quản lý tập tin và thư mục theo thời gian. Các tập tin được lưu trữ trong một kho trung tâm. Kho lưu trữ phần nhiều giống như một máy chủ tập tin bình thường, ngoại trừ việc nó nhớ tất cả các thay đổi đã từng được thực hiện trên các tập tin và thư mục của bạn. Điều này cho phép bạn khôi phục lại phiên bản cũ của tập tin của bạn và kiểm tra lịch sử của việc dữ liệu của bạn đã được thay đổi như thế nào và khi nào, và ai đã thay đổi nó. Đây là lý do tại sao nhiều người nghĩ rằng Subversion và các hệ thống kiểm soát phiên bản nói chung là một loại của “cỗ máy thời gian” .

Một số hệ thống kiểm soát phiên bản cũng là phần mềm quản lý cấu hình (SCM). Các hệ thống này được thiết kế đặc biệt để quản lý cây mã nguồn, và có nhiều tính năng cụ thể cho phát triển phần mềm - chẳng hạn như sự hiểu biết bản xứ về ngôn ngữ lập trình, hoặc cung cấp các công cụ để xây dựng phần mềm. Subversion, tuy nhiên, không phải là một trong những hệ thống này, nó là một hệ thống tổng quát có thể được sử dụng để quản lý *bất kỳ* bộ sưu tập các tập tin, bao gồm cả mã nguồn.

## 2. Các tính năng của TortoiseSVN

Điều gì làm cho TortoiseSVN là một trình khách Subversion tốt? Dưới đây là một danh sách ngắn các tính năng.

### Tích hợp với Shell

TortoiseSVN tích hợp hoàn toàn vào Windows shell (tức là explorer). Điều này có nghĩa là bạn có thể tiếp tục làm việc với các công cụ bạn đã quen thuộc. Và bạn không phải đổi sang một ứng dụng khác nhau mỗi khi bạn cần các chức năng của kiểm soát phiên bản.

Và bạn không bị giới hạn vào việc sử dụng Windows Explorer, trình đơn ngữ cảnh của TortoiseSVN làm việc trong các trình quản lý tập tin khác, và cũng trong hộp thoại File/Open mà là phổ biến trong các ứng dụng Windows tiêu chuẩn nhất. Tuy nhiên, bạn nên nhớ rằng TortoiseSVN cố tình phát triển như là một phần mở rộng cho Windows Explorer. Vì vậy, có thể có trong các ứng dụng khác, sự tích hợp là không đầy đủ, và ví dụ như các lớp phủ biểu tượng có thể không được hiển thị.

### Lớp phủ biểu tượng

Tình trạng của tất cả các tập tin và thư mục có phiên bản được chỉ định bởi các biểu tượng che phủ nhỏ. Bằng cách đó bạn có thể nhìn thấy ngay lập tức tình trạng của bản sao làm việc của bạn là gì.

### Giao diện Người dùng Đồ họa

Khi bạn liệt kê các thay đổi vào một tập tin hoặc thư mục, bạn có thể nhấp chuột vào một sửa đổi để xem các bình luận về commit đó. Bạn cũng có thể thấy một danh sách các tập tin thay đổi - chỉ cần kích đúp vào một tập tin để xem chính xác những gì thay đổi.

Hộp thoại cam kết liệt kê tất cả các mục mà sẽ được bao gồm trong một cam kết, và mỗi mục có một hộp kiểm để bạn có thể chọn những mục bạn muốn bao gồm. Các tập tin không phiên bản cũng có thể được liệt kê, trong trường hợp bạn quên thêm tập tin mới.

Dễ dàng truy cập vào các câu lệnh Subversion

Tất cả các lệnh Subversion có sẵn từ menu trình đơn ngữ cảnh. TortoiseSVN thêm trình đơn con riêng của mình ở đó.

Kể từ khi TortoiseSVN là một trình khách Subversion, chúng tôi cũng muốn để cho bạn thấy một số các tính năng của bản thân Subversion:

Lập phiên bản thư mục

CVS chỉ theo dõi lịch sử các tập tin cá nhân, nhưng Subversion thực hiện một hệ thống tập tin có phiên bản “ảo” để theo dõi những thay đổi trên toàn bộ cây thư mục theo thời gian. Tập tin và thư mục được phiên bản. Kết quả là, thực sự có các lệnh **di chuyển** và **sao chép** từ trình khách hoạt động trên các tập tin và thư mục.

Commit nguyên tử

Một cam kết hoặc đi vào kho lưu trữ hoàn toàn, hoặc không gì cả. Điều này cho phép các nhà phát triển xây dựng và cam kết thay đổi như các khối lô-gic.

Siêu dữ liệu có phiên bản

Mỗi tập tin và thư mục có một tập hợp vô hình của “thuộc tính” đính kèm. Bạn có thể tạo và lưu trữ bất kỳ cặp khóa/giá trị tùy ý bạn muốn. Các thuộc tính được phiên bản theo thời gian, giống như nội dung tập tin.

Lựa chọn của các lớp mạng

Subversion có một khái niệm trừu tượng cho truy cập vào kho lưu trữ, làm cho nó dễ dàng cho mọi người để thực hiện cơ chế mạng lưới mới. Mạng máy chủ “tiên tiến” của Subversion là một mô-đun cho máy chủ web Apache, mà có thể dùng một biến thể của HTTP được gọi là WebDAV/DeltaV. Điều này cho Subversion một lợi thế lớn trong sự ổn định và khả năng tương tác, và cung cấp nhiều tính năng quan trọng miễn phí: xác thực, ủy quyền, nén đường truyền, và duyệt kho lưu trữ, .... một tiến trình máy chủ Subversion cũng có sẵn. Máy chủ này dùng một giao thức tùy chỉnh mà có thể dễ dàng tạo đường hầm qua SSH.

Xử lý dữ liệu nhất quán

Subversion thể hiện sự khác biệt tập tin bằng cách sử dụng một thuật toán tìm khác biệt nhị phân, mà hoạt động giống nhau trên cả hai văn bản (con người có thể đọc được) và tập tin nhị phân (con người không thể đọc được). Cả hai loại tập tin được lưu trữ nén như nhau trong kho, và sự khác biệt được truyền đi theo cả hai hướng trên mạng.

Phân nhánh và gắn thẻ hiệu quả

Chi phí phân nhánh và gắn thẻ không cần phải tỷ lệ thuận với quy mô dự án. Subversion tạo ra các chi nhánh và các thẻ bằng cách đơn giản sao chép các dự án, bằng cách sử dụng một cơ chế tương tự như một liên kết cứng. Vì vậy, các hoạt động này sẽ chỉ mất một lượng thời gian rất nhỏ, và rất ít không gian trong kho.

## 3. Giấy phép

TortoiseSVN là một dự án Mã nguồn Mở được phát triển theo Giấy phép Công cộng GNU (GPL). Nó là miễn phí để tải về và sử dụng miễn phí, hoặc là cá nhân hoặc thương mại, trên bất kỳ số lượng máy tính cá nhân.

Mặc dù hầu hết mọi người chỉ cần tải về bộ cài đặt, bạn cũng có đầy đủ quyền truy cập đọc vào mã nguồn của chương trình này. Bạn có thể duyệt nó trên liên kết này <http://code.google.com/p/tortoisesvn/source/browse/>. Dòng phát triển hiện nay là nằm dưới / trunk / , và các phiên bản phát hành đều nằm trong / tags / .

## 4. Phát triển

Cả hai TortoiseSVN và Subversion được phát triển bởi một cộng đồng của những người đang làm việc trên các dự án. Họ đến từ các quốc gia khác nhau trên toàn thế giới, làm việc với nhau để tạo ra phần mềm tuyệt vời.

### 4.1. Lịch sử của TortoiseSVN

Trong năm 2002, Tim Kemp thấy rằng Subversion là một hệ thống kiểm soát phiên bản rất tốt, nhưng nó thiếu một trình khách giao diện đồ họa tốt. Ý tưởng cho một chương trình khách Subversion là một tích hợp với Windows shell được lấy cảm hứng bởi các trình khách tương tự như CVS tên là TortoiseCVS. Tim nghiên cứu mã nguồn của TortoiseCVS và sử dụng nó như là một cơ sở cho TortoiseSVN. Sau đó, ông bắt đầu dự án, đăng ký tên miền [tortoisesvn.org](http://tortoisesvn.org) và đặt mã nguồn trực tuyến.

Trong khoảng thời gian đó, Stefan Kung đang tìm kiếm một hệ thống kiểm soát phiên bản tốt và miễn phí và tìm thấy Subversion và mã nguồn cho TortoiseSVN. Bởi vì TortoiseSVN vẫn chưa sẵn sàng để sử dụng, ông tham gia dự án và bắt đầu lập trình. Ông đã sớm viết lại hầu hết các mã hiện có và bắt đầu bổ sung thêm lệnh và các tính năng, đến một điểm mà không còn lại gì của mã ban đầu.

Lúc Subversion trở nên ổn định hơn, nó thu hút nhiều người dùng hơn và nhiều người cũng bắt đầu sử dụng TortoiseSVN làm trình khách Subversion của họ. Các cơ sở người dùng phát triển nhanh chóng (và vẫn đang phát triển mỗi ngày). Đó là khi Lübbe Onken đề nghị giúp đỡ với một số biểu tượng đẹp và lô gô cho TortoiseSVN. Hiện tại ông chăm lo trang web và quản lý nhiều bản dịch.

## 4.2. Lời cảm ơn

Tim Kemp  
cho bắt đầu dự án TortoiseSVN

Stefan Kung  
công việc khó khăn để có được TortoiseSVN những gì nó là bây giờ, và sự lãnh đạo của dự án

Lübbe Onken  
các biểu tượng đẹp, biểu tượng, sẵn lỗi, dịch thuật và quản lý các bản dịch

Simon Large  
cho việc duy trì các tài liệu

Stefan Fuhrmann  
cho bộ nhớ cache đăng nhập và chỉnh sửa đồ thị

Sách Subversion  
cho việc giới thiệu tuyệt vời về Subversion và chương của 2 mà chúng ta sao chép ở đây

Dự án phong cách Tigris  
đối với một số phong cách được tái sử dụng trong tài liệu này

Những người đóng góp của chúng tôi  
cho các bản vá lỗi, các báo cáo lỗi và những ý tưởng mới, và để giúp đỡ người khác bằng cách trả lời câu hỏi trong danh sách gửi thư của chúng tôi

Các nhà tài trợ của chúng tôi  
nhiều giờ niềm vui với âm nhạc mà họ gửi cho chúng tôi

## 5. Hướng dẫn đọc

Cuốn sách này được viết cho những người biết máy tính, những người muốn sử dụng Subversion để quản lý dữ liệu của họ, nhưng muốn sử dụng một chương trình máy khách với giao diện đồ họa chứ không phải dòng lệnh. TortoiseSVN là một mở rộng cho Windows và nó giả định rằng người sử dụng quen thuộc với các nhà thám hiểm cuar Windows và biết làm thế nào để sử dụng nó.

Điều này **Lời nói đầu** giải thích những gì TortoiseSVN là một chút về dự án TortoiseSVN và cộng đồng của những người làm việc trên đó, và các điều kiện cấp phép cho việc sử dụng và phân phối.

Các **Chương 1, Bắt đầu** giải thích làm thế nào để cài đặt TortoiseSVN trên máy tính của bạn, và làm thế nào để bắt đầu sử dụng nó ngay lập tức.

Trong **Chương 2, Khái niệm kiểm soát Phiên bản cơ bản** chúng tôi cung cấp cho một giới thiệu ngắn. *Subversion* sửa đổi hệ thống điều khiển làm nền tảng cho TortoiseSVN. Đây là vay từ tài liệu cho dự án Subversion và giải thích các phương pháp tiếp cận khác nhau để kiểm soát phiên bản, và làm thế nào Subversion làm việc được.

Chương về **Chương 3, Kho lưu trữ** giải thích làm thế nào để thiết lập một kho lưu trữ địa phương, đó là hữu ích để thử nghiệm Subversion và TortoiseSVN bằng cách sử dụng một máy tính duy nhất. Nó cũng giải thích một chút

về quản lý kho lưu trữ cũng có liên quan đến kho nằm trên một máy chủ. Ngoài ra còn có một phần ở đây trên làm thế nào để thiết lập một máy chủ nếu bạn cần một cái.

Các **Chương 4, *Hướng dẫn sử dụng hàng ngày*** là phần quan trọng nhất vì nó giải thích tất cả các tính năng chính của TortoiseSVN và làm thế nào để sử dụng chúng. Nó có dạng của một hướng dẫn, bắt đầu với kiểm tra ra một bản sao làm việc, sửa chữa, cam kết thay đổi của bạn, vv Sau đó nó tiến triển đến các chủ đề nâng cao hơn.

**Chương 5, *Chương trình SubWCRev*** là một chương trình riêng biệt với TortoiseSVN có thể trích xuất các thông tin từ bản sao làm việc của bạn và viết nó vào một tập tin. Điều này là hữu ích cho việc bao gồm xây dựng thông tin trong dự án của bạn.

Các **Phụ lục B, *Làm thế nào để tôi...*** ... phần trả lời một số câu hỏi phổ biến về thực hiện nhiệm vụ không rõ ràng bao phủ ở những nơi khác.

Phần **Phụ lục D, *Tự động hoá TortoiseSVN*** cho thấy làm thế nào các TortoiseSVN hộp thoại GUI có thể được gọi từ dòng lệnh. Điều này rất hữu ích cho các kịch bản mà bạn vẫn cần tương tác người dùng.

Các **Phụ lục E, *Tham Chiếu Chéo Giao Diện Dòng Lệnh*** cung cấp cho một mối tương quan giữa các TortoiseSVN lệnh và tương đương trong các chương trình máy khách Subversion dòng lệnh `svn.exe`.

## 6. Thuật ngữ sử dụng trong tài liệu này

Để thực hiện đọc các tài liệu dễ dàng hơn, tên của tất cả các màn hình và thực đơn từ TortoiseSVN được đánh dấu trong một phong chữ khác nhau. Ví dụ: Các **Hộp Thoại Tường Trình**.

Một sự lựa chọn của thực đơn được chỉ định với một mũi tên. TortoiseSVN → **Hiện Tường Trình** có nghĩa là: chọn **Hiện Tường Trình** từ **TortoiseSVN** thực đơn ngữ cảnh.

Tường hợp một thực đơn ngữ cảnh địa phương xuất hiện trong vòng một của hộp thoại TortoiseSVN, nó được hiển thị như thế này: **Context Menu** → Lưu như là...

Nút giao diện người dùng được chỉ định như thế này: Nhấn **OK** để tiếp tục.

Hoạt động người sử dụng được chỉ ra bằng cách sử dụng một font chữ đậm. **Alt + A** : Bấm đồng thời **Alt** - Key trên bàn phím của bạn và trong khi giữ nó xuống Bấm **Một** -Key là tốt. Phải kéo : Nhấn nút chuột phải và trong khi giữ nó xuống *kéo* các mục đến vị trí mới.

Hệ thống đầu ra và đầu vào bàn phím được chỉ định với một phong chữ khác nhau.



### Quan trọng

Ghi chú quan trọng được đánh dấu bằng một biểu tượng.



### Mẹo

Mẹo làm cho cuộc sống của bạn dễ dàng hơn.



### Cẩn thận

Những nơi mà bạn phải cẩn thận những gì bạn đang làm.



### Cảnh báo

Trong trường hợp phải được thực hiện cẩn thận cực kỳ. Dữ liệu bị hỏng hoặc điều khó chịu khác có thể xảy ra nếu những cảnh báo này được bỏ qua.



---

# Chương 1. Bắt đầu

Phần này là nhằm vào những người muốn tìm hiểu TortoiseSVN là tất cả những gì và chạy thử nghiệm nó. Nó giải thích làm thế nào để cài đặt TortoiseSVN và thiết lập một kho lưu trữ địa phương, và nó hướng dẫn cho bạn thông qua các hoạt động được sử dụng phổ biến nhất.

## 1.1. Cài đặt TortoiseSVN

### 1.1.1. Yêu cầu hệ thống

TortoiseSVN chạy trên Windows XP với gói dịch vụ 3 hoặc cao hơn và có sẵn trong cả hai 32-bit và 64-bit. Cài đặt cho Windows 64-bit cũng bao gồm các phần mở rộng 32-bit. Điều đó có nghĩa là bạn không cần phải cài đặt phiên bản 32-bit riêng biệt để có được menu ngữ cảnh TortoiseSVN và lớp phủ trong các ứng dụng 32-bit.



#### Quan trọng

Nếu bạn đang sử dụng Windows XP, bạn *phải* có ít nhất là cài đặt gói dịch vụ 3. Nó sẽ không làm việc nếu bạn chưa cài đặt SP!

Hỗ trợ cho Windows 98, Windows ME và Windows NT4 đã bị bỏ trong phiên bản 1.2.0, và Windows 2000 và XP SP2 hỗ trợ đã bị bỏ trong 1.7.0. Bạn vẫn có thể tải về và cài đặt phiên bản cũ nếu bạn cần chúng.

### 1.1.2. Cài đặt

TortoiseSVN đi kèm với một bộ cài đặt dễ sử dụng. Kích đúp vào tập tin cài đặt và làm theo hướng dẫn. Bộ cài đặt sẽ lo liệu phần còn lại. Đừng quên khởi động lại sau khi cài đặt.



#### Quan trọng

Bạn cần quyền quản trị để cài đặt TortoiseSVN.

Gói ngôn ngữ được dịch giao diện người dùng TortoiseSVN ra nhiều ngôn ngữ khác nhau. Hãy kiểm tra **Phụ lục G, Các gói ngôn ngữ và bộ kiểm tra chính tả** để biết thêm thông tin về việc làm thế nào để cài đặt.

Nếu bạn gặp bất kỳ vấn đề gì trong quá trình hoặc sau khi cài đặt TortoiseSVN xin tham khảo Hỏi Đáp trực tuyến của chúng tôi tại <http://tortoisesvn.net/faq.html>.

## 1.2. Các khái niệm cơ bản

Trước khi chúng ta bắt đầu làm việc với các tập tin thật, điều quan trọng là có cái nhìn tổng quát về cách Subversion làm việc và các thuật ngữ được dùng.

### Kho lưu trữ

Subversion sử dụng một cơ sở dữ liệu trung tâm, trong đó tất cả các phiên bản của bạn kiểm soát các tập tin với toàn bộ lịch sử của chúng. Cơ sở dữ liệu này được gọi là *Kho lưu trữ*. Kho lưu trữ thường đặt trên một máy chủ tập tin chạy chương trình máy chủ Subversion, là nguồn cung cấp nội dung cho các trình khách Subversion (như TortoiseSVN) theo yêu cầu. Nếu bạn chỉ sao lưu một thứ, sao lưu kho lưu trữ của bạn vì nó là bản sao tổng thể dứt khoát của tất cả các dữ liệu của bạn.

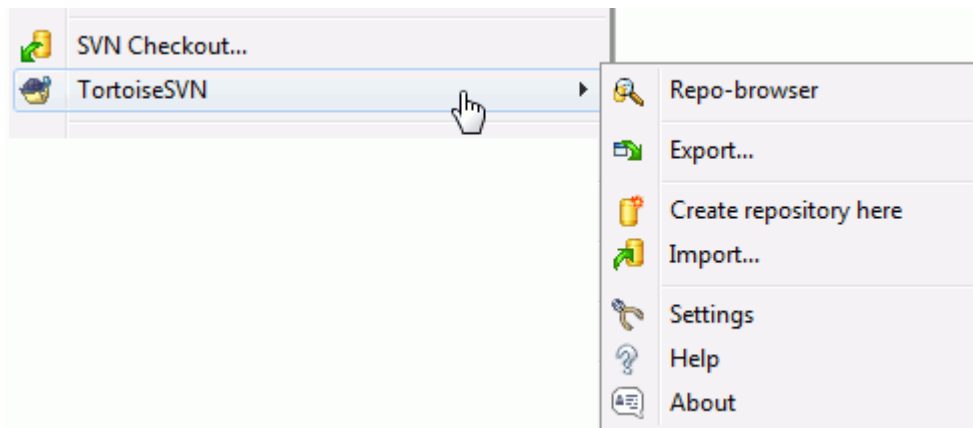
### Bản sao làm việc

Đây là nơi bạn làm công việc thực sự. Mọi nhà phát triển đều có bản sao làm việc của chính mình, đôi khi được biết đến như một sandbox, trên máy tính địa phương của mình. Bạn có thể kéo xuống phiên bản mới nhất từ kho, làm việc trên nó tại địa phương mà không ảnh hưởng đến bất cứ ai khác, sau đó khi bạn đang hài lòng với những thay đổi bạn đã thực hiện cam kết chúng lại vào kho lưu trữ.



Bản sao Subversion làm việc không chứa lịch sử của dự án, nhưng nó giữ một bản sao của các tập tin như chúng tồn tại trong kho lưu trữ trước khi bạn bắt đầu thay đổi. Điều này có nghĩa rằng nó rất dễ dàng để kiểm tra chính xác những gì thay đổi bạn đã thực hiện.

Bạn cũng cần phải biết nơi để tìm TortoiseSVN vì không có nhiều để xem từ Start Menu. Điều này là do TortoiseSVN là một phần mở rộng Shell, vì vậy đầu tiên của tất cả, khởi động Windows Explorer. Nhấp chuột phải vào một thư mục trong Explorer và bạn sẽ thấy một số mục mới trong menu ngữ cảnh như thế này:



Hình 1.1. Các trình đơn TortoiseSVN cho các thư mục không phiên bản

## 1.3. Đi lái thử

Phần này cho bạn thấy làm thế nào để dùng thử một số các tính năng phổ biến nhất được sử dụng trên một kho lưu trữ thử nghiệm nhỏ. Đương nhiên nó không giải thích tất cả mọi thứ - điều này chỉ là hướng dẫn bắt đầu nhanh. Một khi bạn có mọi thứ sắp đặt và đang chạy, bạn nên dành thời gian để đọc phần còn lại của hướng dẫn sử dụng này, sẽ đưa bạn thông qua những điều chi tiết hơn. Nó cũng giải thích thêm về việc thiết lập một máy chủ Subversion phù hợp.

### 1.3.1. Tạo một kho lưu trữ

Đối với một dự án thực tế, bạn sẽ có một kho lưu trữ một nơi nào đó an toàn và một máy chủ Subversion để kiểm soát nó. Theo mục đích của hướng dẫn này chúng tôi sẽ sử dụng tính năng kho lưu trữ Subversion địa phương cho phép truy cập trực tiếp vào một kho lưu trữ được tạo ra trên ổ cứng của bạn mà không cần một máy chủ nào.

Đầu tiên tạo ra một thư mục trống trên máy tính của bạn. Nó có thể đi bất cứ nơi nào, nhưng trong hướng dẫn này chúng tôi sẽ gọi nó là `C:\svn_repos`. Bây giờ kích chuột phải vào thư mục mới và từ trình đơn ngữ cảnh chọn **TortoiseSVN** → **Tạo Kho Lưu Trữ ở đây ...**. Kho lưu trữ này sau đó được tạo ra bên trong thư mục, sẵn sàng cho bạn sử dụng. Chúng tôi cũng sẽ tạo ra các trúc thư mục mặc định nội bộ bằng cách nhấn vào nút **Tạo cấu trúc thư mục**.

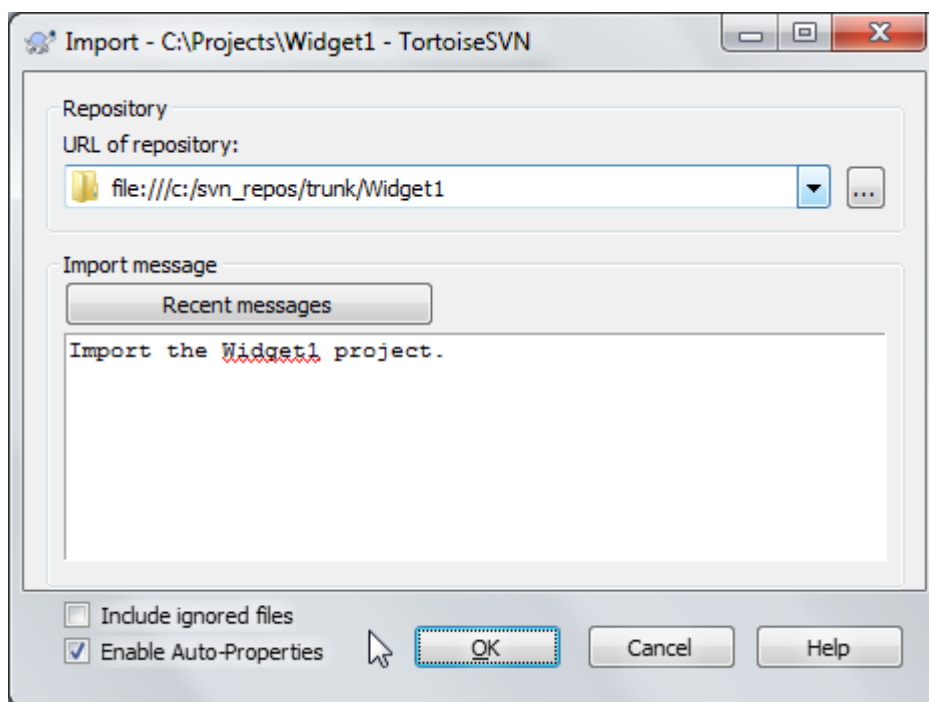


#### Quan trọng

Các tính năng kho lưu trữ địa phương là rất hữu ích cho kiểm tra và đánh giá, nhưng trừ khi bạn đang làm việc như một nhà phát triển duy nhất trên một máy tính, bạn nên luôn luôn sử dụng một máy chủ Subversion thích hợp. Nó là hấp dẫn trong một công ty nhỏ để tránh các công việc thiết lập một máy chủ và chỉ cần truy cập vào kho của bạn trên một mạng chia sẻ. Đừng bao giờ làm điều đó. Bạn sẽ bị mất dữ liệu. Đọc [Phần 3.1.4, “Truy cập vào một kho lưu trữ trên một mạng chia sẻ”](#) để tìm hiểu lý do tại sao đây là một ý tưởng tồi, và làm thế nào để thiết lập một máy chủ.

### 1.3.2. Nhập khẩu một dự án

Bây giờ chúng ta có một kho lưu trữ, nhưng nó là hoàn toàn trống rỗng vào lúc này. Hãy giả sử tôi có một tập hợp các tập tin trong `C:\Projects\Widget1` mà tôi muốn thêm vào. Định hướng đến thư mục `Widget1` trong Explorer và phải nhấp chuột vào nó. Bây giờ chọn **TortoiseSVN** → **Nhập khẩu ...** mà đem lên một hộp thoại



**Hình 1.2. Hộp thoại nhập khẩu**

Một kho lưu trữ Subversion được gọi bằng URL, cho phép chúng ta xác định một kho lưu trữ bất cứ nơi nào trên Internet. Trong trường hợp này chúng ta cần chỉ đến kho lưu trữ địa phương của chúng ta có URL `file:///C:/svn_repos/trunk/`, và chúng ta thêm tên dự án của chúng ta `Widget1`. Lưu ý rằng có 3 dấu gạch chéo sau `file:` và đó là dấu gạch chéo phía trước được sử dụng xuyên suốt.

Các tính năng quan trọng khác của hộp thoại này là **Nhập khẩu tin nhắn** hộp cho phép bạn nhập một thông báo mô tả những gì bạn đang làm. Khi bạn đến để xem xét thông qua lịch sử dự án của bạn, các thông báo cam kết là một tài liệu hướng dẫn có giá trị về những thay đổi đã được thực hiện và lý do tại sao. Trong trường hợp này, chúng ta có thể nói một cái gì đó đơn giản như “Nhập khẩu các dự án Widget1”. Click vào **OK** và thư mục được bổ sung vào kho lưu trữ của bạn.

### 1.3.3. Kiểm xu một bản sao làm việc

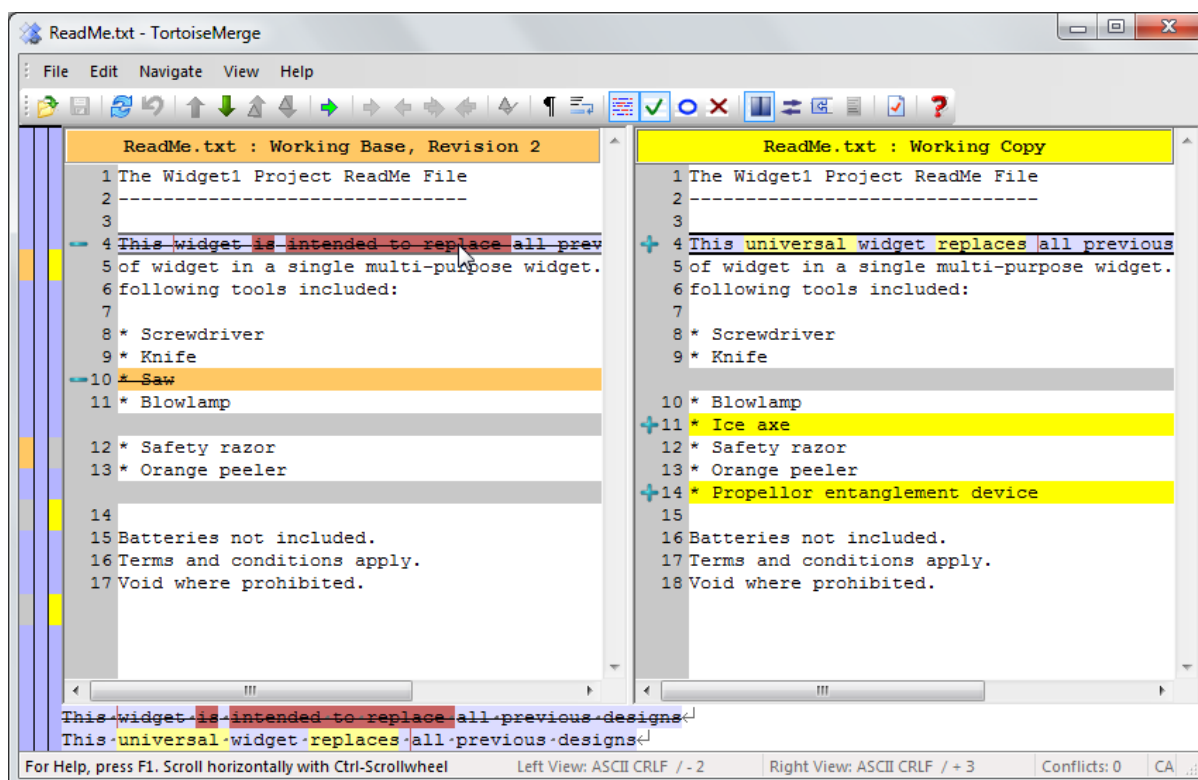
Bây giờ chúng ta có một dự án trong kho của chúng ta, chúng ta cần tạo ra một bản sao hoạt động để sử dụng cho công việc ngày-sang-ngày. Lưu ý rằng các hành vi nhập khẩu một thư mục không tự động chuyển thư mục đó vào một bản sao làm việc. Thuật ngữ Subversion để tạo ra một bản sao làm việc tươi mới là **Kiểm xuất**. Chúng tôi sẽ kiểm xuất thư mục `Widget1` của kho lưu trữ của chúng ta vào một thư mục phát triển trên máy tính được gọi là `C:\Projects\Widget1-Dev`. Tạo thư mục đó, sau đó nhấp chuột phải vào nó và chọn **TortoiseSVN** → **Kiểm xuất ...**. Nhập URL để kiểm xuất, trong trường hợp này `file:///c:/svn_repos/trunk/Widget1` và nhấp vào **OK**. Thư mục phát triển của chúng ta sau đó được lấp đầy với các tập tin từ kho.

Bạn sẽ thấy rằng sự xuất hiện của thư mục này là khác nhau từ thư mục ban đầu của chúng tôi. Tập tin đều có một dấu kiểm màu xanh lá cây ở góc dưới cùng bên trái. Đây là những biểu tượng tình trạng TortoiseSVN chỉ xuất hiện trong một bản sao làm việc. Nhà nước màu xanh lá cây chỉ ra rằng tập tin là không thay đổi so với phiên bản trong kho.

### 1.3.4. Thực hiện Thay đổi

Thời gian để làm việc. Trong `Widget1-dev` chúng tôi bắt đầu chỉnh sửa các tập tin - chúng ta hãy nói rằng chúng ta thay đổi `Widget1.c` và `Readme.txt`. Lưu ý rằng các lớp phủ biểu tượng trên những tập tin này đã thay đổi thành màu đỏ, cho thấy rằng những thay đổi đã được thực hiện tại địa phương.

Tuy nhiên, các thay đổi là những gì? Nhấp chuột phải vào một trong các tập tin thay đổi và chọn **TortoiseSVN** → **Khác**. Công cụ so sánh tập tin của TortoiseSVN bắt đầu, hiển thị cho bạn chính xác những dòng đã thay đổi.



**Hình 1.3. Trình xem khác biệt của tập tin**

OK, vì chúng ta rất vui với những thay đổi, chúng ta hãy cập nhật các kho lưu trữ. Hành động này được gọi là Cam kết các thay đổi. Nhấp chuột phải vào Widget1-dev thư mục và chọn TortoiseSVN → Cam kết. Hộp thoại cam kết liệt kê các tập tin thay đổi, mỗi một hộp kiểm. Bạn có thể muốn chọn chỉ là một tập hợp con của các tập tin, nhưng trong trường hợp này chúng tôi sẽ cam kết những thay đổi đến cả hai tập tin. Nhập một thông điệp để mô tả tất cả những gì thay đổi và bấm vào OK. Các hộp thoại tiến triển sẽ hiển thị các tập tin được tải lên kho lưu trữ và bạn đã làm xong.

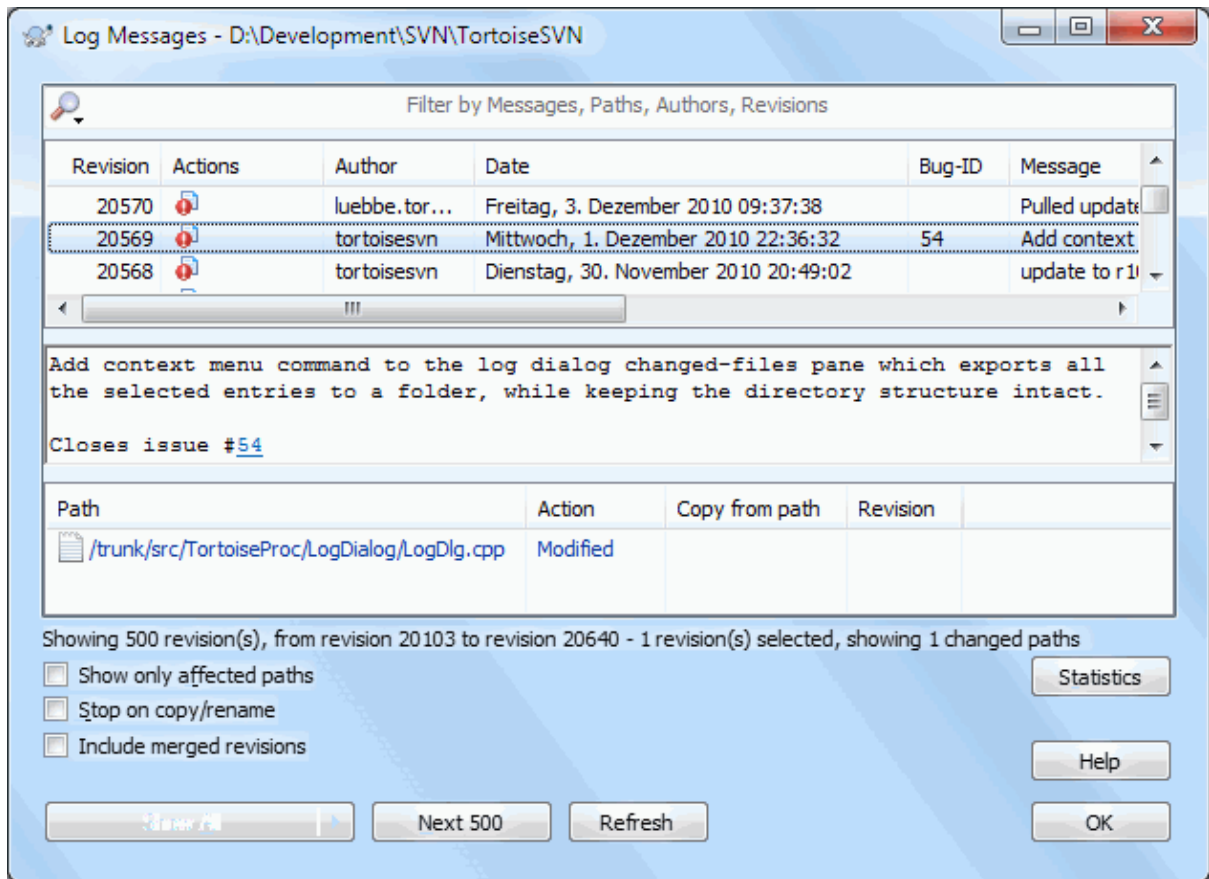
### 1.3.5. Thêm nhiều tập tin

Khi dự án phát triển, bạn sẽ cần thêm tập tin mới - chúng ta hãy nói rằng bạn thêm một số tính năng mới trong Extras.c và thêm một tài liệu tham khảo hiện có Makefile. Nhấp chuột phải vào thư mục và TortoiseSVN → Thêm. Hộp thoại Thêm bây giờ cho bạn thấy tất cả các tập tin không phiên bản và bạn có thể chọn những người cái mà bạn muốn thêm vào. Một cách khác để thêm các tập tin sẽ là kích chuột phải vào tập tin và chọn TortoiseSVN → Thêm.

Bây giờ khi bạn thực hiện cam kết các thư mục, tập tin mới xuất hiện là Thêm và các tập tin hiện có như Thay đổi. Lưu ý rằng bạn có thể nhấn đôi vào tập tin sửa đổi để kiểm tra chính xác những thay đổi đã được thực hiện.

### 1.3.6. Xem Lịch sử dự án

Một trong những tính năng hữu ích nhất của TortoiseSVN là hộp thoại Tường Trình. Điều này cho thấy bạn một danh sách của tất cả các cam kết bạn đã thực hiện một tập tin hoặc thư mục, và hiển thị những thông báo cam kết chi tiết mà bạn đã nhập ;-)



**Hình 1.4. Hộp thoại Tường Trình**

OK, do đó, tôi lừa một chút ở đây và sử dụng một ảnh chụp màn hình từ kho lưu trữ TortoiseSVN.

Các cửa sổ đầu cho thấy một danh sách các sửa đổi cam kết cùng với sự bắt đầu của thông báo cam kết. Nếu bạn chọn một trong những sửa đổi, cửa sổ giữa sẽ hiển thị các thông báo tường trình đầy đủ cho sửa đổi đó và khung bên dưới sẽ hiển thị một danh sách các tập tin và thư mục được thay đổi.

Mỗi khung có một menu ngữ cảnh cung cấp cho bạn cách nhiều hơn nữa bằng cách sử dụng các thông tin. Trong khung dưới cùng, bạn có thể nhấp đúp chuột vào một tập tin để xem chính xác những thay đổi đã được thực hiện trong bản chỉnh sửa đó. Đọc [Phần 4.9, “Hộp thoại Tường trình Chính sửa”](#) để có được toàn bộ câu chuyện.

### 1.3.7. Hoàn tác thay đổi

Một trong những tính năng của tất cả các hệ thống kiểm soát phiên bản là họ cho phép bạn lùi lại thay đổi mà bạn đã thực hiện trước đó. Như bạn mong đợi, TortoiseSVN làm cho việc này được dễ dàng để truy cập.

Nếu bạn muốn được thoát khỏi những thay đổi mà bạn chưa cam kết và thiết lập lại tập tin của bạn như trước khi bạn bắt đầu chỉnh sửa, TortoiseSVN → Hoàn nguyên là bạn của bạn. Điều này loại bỏ những thay đổi (vào thùng rác, chỉ để phòng hờ) và trở lại với phiên bản cam kết bạn bắt đầu. Nếu bạn muốn thoát khỏi chỉ một số trong những thay đổi, bạn có thể sử dụng TortoiseMerge để xem sự khác biệt và chọn lọc trở lại các dòng thay đổi.

Nếu bạn muốn phục hồi lại những ảnh hưởng của một phiên bản nhất định, bắt đầu với hộp thoại Tường Trình và tìm những sửa đổi vi phạm. Chọn Thực Đơn Ngữ Cảnh → Hoàn nguyên các thay đổi từ phiên bản này và những thay đổi sẽ được hoàn tác.

## 1.4. Đi tiếp ...

Hướng dẫn này đã đưa cho bạn đi một vòng rất nhanh chóng về một số tính năng quan trọng và hữu ích nhất của TortoiseSVN, nhưng tất nhiên có nhiều điều hơn mà chúng tôi đã không trình bày. Chúng tôi mạnh mẽ khuyên

bạn nên dành thời gian để đọc phần còn lại của sổ tay này, đặc biệt là **Chương 4, *Hướng dẫn sử dụng hàng ngày*** cung cấp cho bạn rất nhiều chi tiết hơn về hoạt động ngày-sang-ngày.

Chúng tôi đã có rất nhiều khó khăn để đảm bảo rằng nó vừa nhiều thông tin và dễ dàng để đọc, nhưng chúng tôi nhận ra rằng vẫn còn có rất nhiều khó khăn! Hãy dành thời gian của bạn và không ngại thử những điều trên một kho lưu trữ thử nghiệm như bạn đi cùng. Cách tốt nhất để học là bằng cách sử dụng nó.

---

# Chương 2. Khái niệm kiểm soát Phiên bản cơ bản

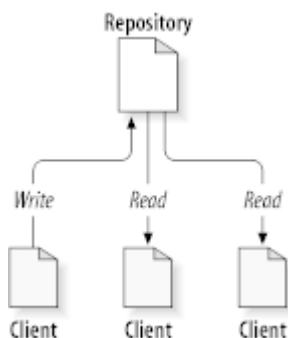
Chương này là một phiên bản sửa đổi của cùng một chương trong cuốn sách Subversion. Một phiên bản trực tuyến của cuốn sách Subversion có sẵn ở đây: <http://svnbook.red-bean.com/> [<http://svnbook.red-bean.com/>].

Chương này là một giới thiệu ngắn, bình thường tới Subversion. Nếu bạn là người mới đến với kiểm soát phiên bản, chương này chắc chắn là dành cho bạn. Chúng ta bắt đầu với một cuộc thảo luận về khái niệm chung về kiểm soát phiên bản, làm việc theo cách của chúng tôi vào những ý tưởng cụ thể sau Subversion, và hiển thị một số ví dụ đơn giản về Subversion đang được sử dụng.

Mặc dù các ví dụ trong chương này cho thấy mọi người chia sẻ bộ sưu tập của mã nguồn chương trình, hãy nhớ rằng Subversion có thể quản lý bất kỳ loại sưu tập tập tin nào - nó không bị giới hạn trong việc giúp các lập trình viên máy tính.

## 2.1. Kho lưu trữ

Subversion là một hệ thống tập trung chia sẻ thông tin. Tại cốt lõi của nó là một *kho*, đó là một cửa hàng trung tâm dữ liệu. Kho lưu trữ thông tin trong các hình thức của một *cây hệ thống tập tin* - Một hệ thống cấp bậc của các tập tin và thư mục. Số lượng bất kỳ *các chương trình khách* kết nối với kho lưu trữ, và sau đó đọc hoặc viết vào những tập tin này. Bằng cách ghi dữ liệu, chương trình khách hàng làm cho thông tin có sẵn cho những người khác; bằng cách đọc dữ liệu, chương trình khách hàng nhận được thông tin từ những người khác.



**Hình 2.1. Một hệ thống Máy Khách/Máy Chủ điển hình**

Vì vậy, tại sao điều này thú vị? Cho đến nay, điều này nghe giống như định nghĩa của một máy chủ tập tin thông thường. Và quả thực, các kho lưu trữ là một loại máy chủ tập tin, nhưng nó không giống loại bình thường của bạn. Những gì làm cho kho lưu trữ Subversion đặc biệt là *nó nhớ mọi sự thay đổi* từng được viết vào nó: mỗi lần thay đổi cho mỗi tập tin, và thậm chí thay đổi cây thư mục, chẳng hạn như việc bổ sung, xóa, và sắp xếp lại các tập tin và thư mục.

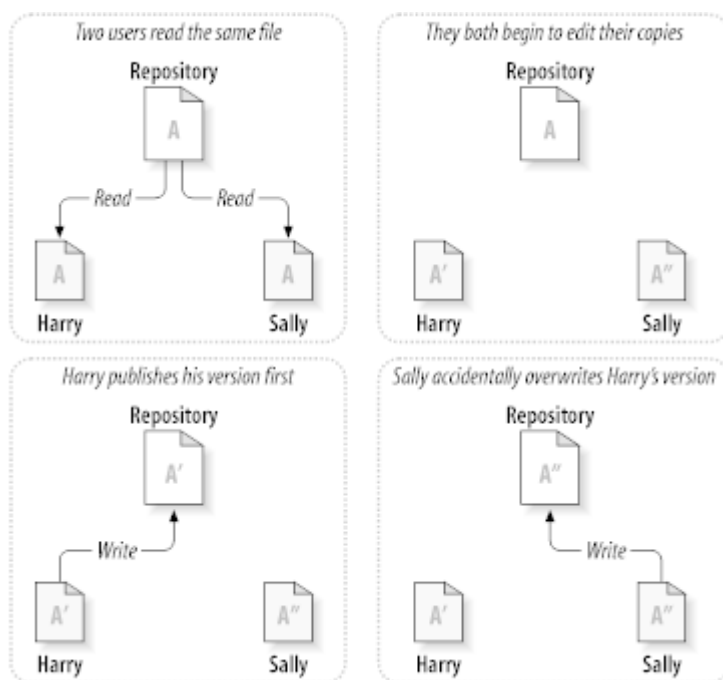
Khi một khách hàng đọc dữ liệu từ kho lưu trữ, nó thường chỉ nhìn thấy có phiên bản mới nhất của cây hệ thống tập tin. Nhưng khách hàng cũng có khả năng xem *trước* trạng thái của hệ thống tập tin. Ví dụ, một khách hàng có thể đặt câu hỏi lịch sử như, “thư mục này đã làm những gì chứa thứ Tư tuần trước?”, Hoặc “người cuối cùng để thay đổi tập tin này, và thay đổi những gì họ làm cho?” Đây là loại câu hỏi được ở trung tâm của bất kỳ *hệ thống kiểm soát phiên bản*: Hệ thống được thiết kế để ghi lại và theo dõi những thay đổi dữ liệu theo thời gian.

## 2.2. Mô hình Phiên bản

Tất cả các hệ thống kiểm soát phiên bản phải giải quyết cùng một vấn đề cơ bản: làm thế nào hệ thống sẽ cho phép người dùng chia sẻ thông tin, nhưng ngăn cản họ vô tình dẫm trên chân của nhau? Đó là tất cả quá dễ dàng cho người dùng vô tình ghi đè lên các thay đổi khác trong kho lưu trữ.

### 2.2.1. Vấn đề chia sẻ tập tin

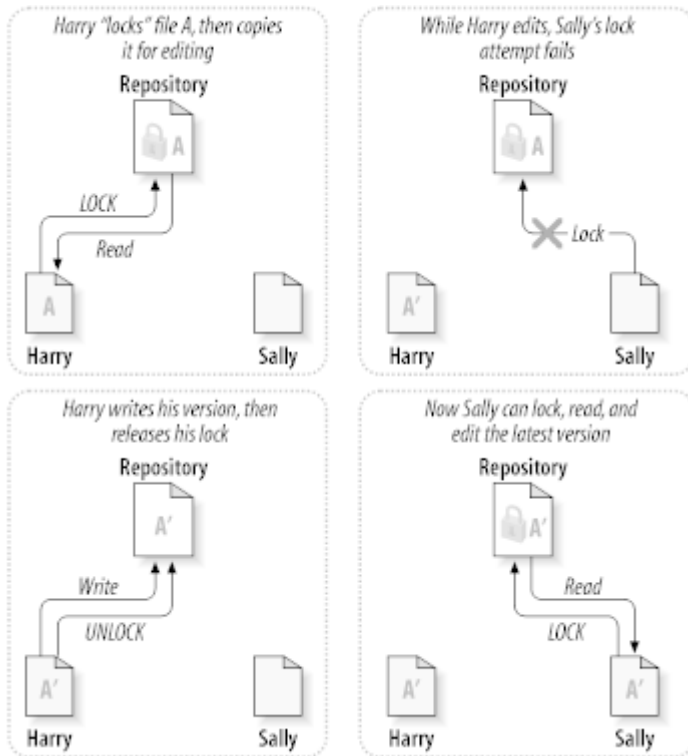
Hãy xem xét kịch bản này: giả sử chúng ta có hai đồng nghiệp, Harry và Sally. Mỗi quyết định chỉnh sửa các tập tin kho lưu trữ tại cùng một thời gian. Nếu Harry lưu thay đổi của mình vào kho lưu trữ đầu tiên, có thể một vài khoảng khắc sau đó Sally vô tình có thể ghi đè lên chúng với phiên bản mới của tập tin. Trong khi Harry phiên bản của tập tin sẽ không bị mất mãi mãi (do hệ thống nhớ mỗi lần thay đổi), bất kỳ thay đổi nào Harry đã thực hiện *sẽ không* có mặt trong phiên bản mới hơn của Sally của tập tin, bởi vì cô không bao giờ nhìn thấy những thay đổi để bắt đầu với Harry. Tác phẩm của Harry vẫn còn có hiệu quả bị mất hoặc ít nhất là mất tích từ các phiên bản mới nhất của tập tin - và có lẽ do tai nạn. Điều này chắc chắn là một tình huống mà chúng ta muốn tránh!



Hình 2.2. Vấn đề cần tránh

### 2.2.2. Giải pháp Khóa-Sửa-Mở khóa

Nhiều phiên bản kiểm soát hệ thống sử dụng một mô hình *khóa-thay đổi-mở khóa* để giải quyết vấn đề này, mà là một giải pháp rất đơn giản. Trong hệ thống như vậy, kho lưu trữ chỉ cho phép một người thay đổi một tập tin tại một thời điểm. Đầu tiên Harry phải *khóa* các tập tin trước khi có thể bắt đầu thay đổi nó. Khóa một tập tin là rất giống như mượn một cuốn sách từ thư viện, nếu Harry đã khóa một tập tin, thì sau đó Sally không thể thực hiện bất kỳ thay đổi với nó. Nếu cô ấy cố gắng để khóa các tập tin, các kho lưu trữ sẽ từ chối yêu cầu. Tất cả những gì có thể làm là đọc các tập tin, và chờ đợi cho Harry để kết thúc những thay đổi của mình và phát hành khóa của mình. Sau khi Harry mở các tập tin, hết lượt của mình, và bây giờ Sally có thể lần lượt của mình bằng cách khóa và chỉnh sửa.



**Hình 2.3. Giải pháp Khóa-Sửa-Mở khóa**

Vấn đề với mô hình khóa, sửa đổi, mở khóa là nó có một chút hạn chế, và thường trở thành một rào cản cho người sử dụng:

- *Khóa có thể gây ra các vấn đề hành chính.* Đôi khi Harry sẽ khóa một tập tin và sau đó quên nó. Trong khi đó, bởi vì Sally là vẫn đang chờ đợi để chỉnh sửa các tập tin, bàn tay của cô bị bó buộc. Và sau đó Harry đi vào kỳ nghỉ. Bây giờ Sally phải nhờ một quản trị viên để phát hành khóa của Harry. Tình hình kết thúc lên gây ra rất nhiều sự chậm trễ không cần thiết và lãng phí thời gian.
- *Khóa có thể gây ra tuần tự không cần thiết.* Nếu Harry là chỉnh sửa bắt đầu của một tập tin văn bản, và Sally chỉ đơn giản là muốn chỉnh sửa cuối cùng một tập tin? Những thay đổi này không chồng chéo lên nhau chút nào. Họ có thể dễ dàng chỉnh sửa các tập tin cùng một lúc, và không có hại lớn, giả sử những thay đổi sáp nhập với nhau đúng cách. Không cần cho họ phải thay phiên nhau trong tình huống này.
- *Khóa có thể tạo ra một cảm giác an toàn sai lầm.* Giả vờ rằng Harry khóa và chỉnh sửa tập tin A, trong khi đồng thời Sally khóa và chỉnh sửa các tập tin B. Nhưng giả sử rằng A và B phụ thuộc vào nhau, và các thay đổi được thực hiện cho từng ngữ nghĩa không tương thích. Đột nhiên A và B không làm việc cùng nhau nữa. Hệ thống khóa bắt lực để ngăn chặn vấn đề nhưng nó bằng cách nào đó cung cấp một cảm giác an toàn sai. Thật dễ dàng cho Harry và Sally để tưởng tượng rằng các tập tin khóa, mỗi bắt đầu một công việc an toàn, cách biệt, và do đó ức chế sự thảo luận về những thay đổi không tương thích của họ sớm hơn.

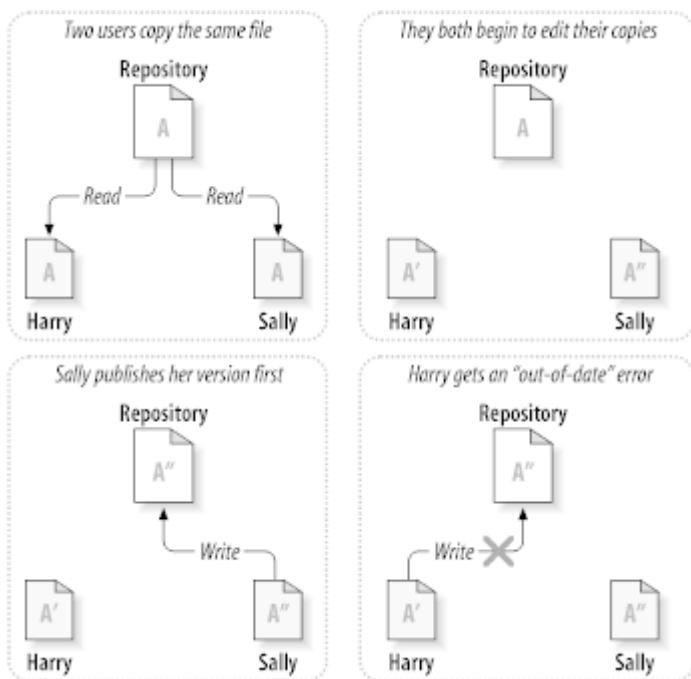
### 2.2.3. Giải pháp Sao chép-Sửa đổi-Hợp nhất

Subversion, CVS, và các hệ thống kiểm soát phiên bản khác sử dụng một mô hình *sao chép, sửa đổi, hợp nhất* như là một thay thế cho khóa. Trong mô hình này, khách hàng của mỗi người dùng đọc các kho lưu trữ và tạo ra một cá nhân *bản sao làm việc* của tập tin hoặc dự án. Người dùng sau đó làm việc song song, sửa đổi các bản tin riêng của họ. Cuối cùng, các bản tin được kết hợp lại với nhau thành một phiên bản mới, cuối cùng. Hệ thống kiểm soát phiên bản thường hỗ trợ với sự kết hợp, nhưng cuối cùng một con người có trách nhiệm để làm cho nó xảy ra một cách chính xác.

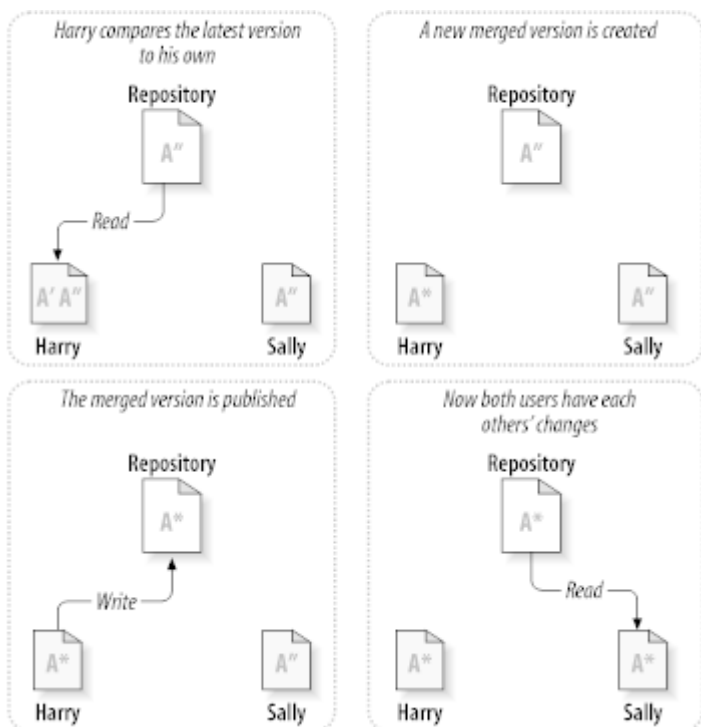
Dưới đây là một ví dụ. Hãy nói rằng Harry và Sally đều tạo ra các bản sao làm việc cùng một dự án, sao chép từ kho. Họ làm việc đồng thời, và thay đổi cùng một tập tin A trong bản sao của họ. Sally lưu thay đổi của mình vào kho lưu trữ đầu tiên. Khi Harry cố gắng để lưu các thay đổi của ông sau đó, kho lưu trữ thông báo cho ông rằng



tập tin của mình *quá cũ* . Nói cách khác, tập tin A đó trong kho lưu trữ bằng cách nào đó thay đổi kể từ khi ông mới sao chép nó. Vì vậy, Harry yêu cầu khách hàng của mình để *hợp nhất* bất kỳ thay đổi mới nhất từ kho vào bản sao hoạt động của tập tin A. Có thể là những thay đổi của Sally không chồng chéo lên nhau với bản của riêng ông ta, do đó, một khi ông có cả bộ những thay đổi tích hợp, ông lưu bản sao làm việc của mình trở lại vào kho.



**Hình 2.4. Giải pháp Sao chép-Sửa đổi-Hợp nhất**



**Hình 2.5. ... Sao chép-Sửa đổi-Hợp nhất Tiếp theo**

Nhưng điều gì sẽ xảy ra nếu thay đổi của Sally *làm* trùng với thay đổi của Harry? Những gì sau đó? Tình trạng này được gọi là một *xung đột* , Và nó thường là không phải là một vấn đề gì nhiều. Khi Harry yêu cầu khách

hàng của mình để hợp nhất các thay đổi kho lưu trữ mới nhất vào bản sao làm việc của mình, bản sao của tập tin A là bằng cách nào đó gắn cờ là trong trạng thái xung đột: anh ta sẽ có thể nhìn thấy cả hai bộ những thay đổi xung đột, và tự lựa chọn giữa chúng. Lưu ý rằng phần mềm có thể không tự động giải quyết xung đột, con người chỉ có khả năng hiểu biết và thực hiện những lựa chọn thông minh cần thiết. Khi Harry đã giải quyết thủ công những thay đổi chồng chéo (có lẽ bằng cách thảo luận về cuộc xung đột với Sally), ông một cách an toàn có thể lưu các tập tin bị sáp nhập trở lại vào kho.

Mô hình sao chép, sửa đổi, sáp nhập có thể nghe một chút hỗn loạn, nhưng trong thực tế, nó chạy rất trơn tru. Người dùng có thể làm việc song song, không bao giờ chờ đợi nhau. Khi họ làm việc trên các tập tin tương tự, nó hóa ra rằng hầu hết các thay đổi đồng thời của họ không chồng chéo lên nhau ở tất cả các xung đột hiếm khi xảy ra. Và số lượng thời gian cần thiết để giải quyết xung đột là ít hơn so với thời gian đã mất bằng một hệ thống khóa.

Cuối cùng, nó đi xuống đến một yếu tố quan trọng: người sử dụng thông tin liên lạc. Khi người sử dụng giao tiếp kém, cả hai cú pháp và ngữ nghĩa xung đột gia tăng. Không có hệ thống có thể buộc người dùng để giao tiếp hoàn hảo, và không có hệ thống có thể phát hiện xung đột ngữ nghĩa. Vì vậy, không có điểm bị đẩy vào một lời hứa sai lầm rằng một hệ thống khóa bằng cách nào đó sẽ ngăn chặn xung đột, trong thực tế, khóa có vẻ ức chế năng suất hơn bất cứ điều gì khác.

Có một tình hình phổ biến nơi mô hình khóa-sửa đổi-mở khóa cho kết quả tốt hơn, và đó là nơi mà bạn có các tập tin không hợp nhất được. Ví dụ, nếu kho lưu trữ của bạn có chứa một số hình ảnh đồ họa, và hai người thay đổi hình ảnh tại cùng một thời gian, không có cách nào cho những thay đổi được sáp nhập với nhau. Hoặc là Harry hoặc Sally sẽ mất các thay đổi.

#### 2.2.4. Subversion làm gì?

Subversion sử dụng các bản sao chỉnh sửa, kết hợp giải pháp theo mặc định, và trong nhiều trường hợp này là tất cả những gì mà bạn cần. Tuy nhiên, phiên bản 1.2, Subversion cũng hỗ trợ các khóa tập tin, vì vậy nếu bạn có các tập tin không hợp nhất được, hoặc nếu bạn chỉ đơn giản là bị ép buộc vào một chính sách khóa bởi ban quản lý, Subversion vẫn sẽ cung cấp các tính năng bạn cần.

### 2.3. Subversion trong hành động

#### 2.3.1. Bản sao Làm việc

Bạn đã đọc về bản sao làm việc, bây giờ chúng tôi sẽ chứng minh làm thế nào các khách hàng Subversion tạo ra và sử dụng chúng.

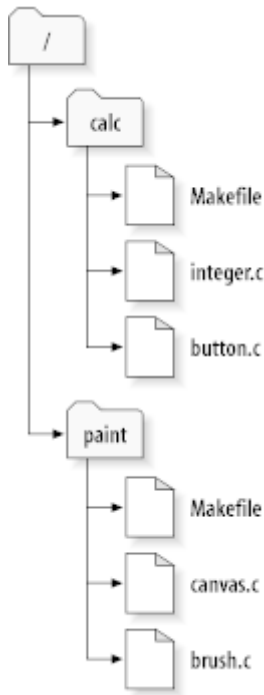
Một bản sao làm việc Subversion là một cây thư mục bình thường trên hệ thống địa phương của bạn, có chứa một bộ sưu tập các tập tin. Bạn có thể chỉnh sửa những tập tin này tùy cách bạn muốn, và nếu chúng là tập tin mã nguồn, bạn có thể biên dịch chương trình của bạn từ chúng theo cách thông thường. Bản sao làm việc của bạn là khu vực làm việc riêng tư của riêng bạn: Subversion sẽ không bao giờ kết hợp với thay đổi của người khác, cũng không thay đổi của bạn có sẵn cho những người khác, cho đến khi bạn cho nó để làm như vậy một cách rõ ràng.

Sau khi bạn đã thực hiện một số thay đổi các tập tin trong bản sao làm việc của bạn và xác nhận rằng chúng hoạt động đúng, Subversion cung cấp cho bạn các lệnh để *xuất bản* thay đổi của bạn cho người khác làm việc với bạn về dự án của bạn (bằng cách viết vào kho lưu trữ). Nếu người khác công bố những thay đổi riêng của họ, Subversion cung cấp cho bạn với lệnh hợp nhất những thay đổi đó vào thư mục làm việc của bạn (bằng cách đọc từ kho).

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

Một kho lưu trữ Subversion điển hình thường xuyên tổ chức các tập tin (hoặc mã nguồn) cho nhiều dự án, thông thường, mỗi dự án là một thư mục con trong cây hệ thống tập tin của kho lưu trữ. Trong sự sắp xếp này, bản sao làm việc của người dùng thường sẽ tương ứng với một cây con đặc biệt của kho.

Ví dụ, giả sử bạn có một kho lưu trữ có chứa hai dự án phần mềm.



**Hình 2.6. Các hệ thống tập tin của kho**

Nói cách khác, thư mục gốc của kho lưu trữ có hai thư mục con: `paint` và `calc`.

Để có được một bản sao làm việc, bạn phải *kiểm tra ra* một số cây con của kho. (Thuật ngữ *kiểm tra ra* có thể nghe như nó có cái gì để làm với khóa hoặc dự trữ tài nguyên, nhưng nó không, nó chỉ đơn giản là tạo ra một bản sao riêng của các dự án cho bạn).

Giả sử bạn thay đổi `button.c`. Bởi vì `svn` thư mục nhớ lại ngày sửa đổi của tập tin và nội dung ban đầu, Subversion có thể nói rằng bạn đã thay đổi các tập tin. Tuy nhiên, Subversion không thực hiện thay đổi công khai cho đến khi bạn cho phép nó một cách rõ ràng. Các hành vi xuất bản thay đổi của bạn thường được gọi là *cam kết* (Hoặc *kiểm tra vào*) các thay đổi vào kho.

Để xuất bản các thay đổi của bạn cho người khác, bạn có thể sử dụng lệnh Subversion **cam kết**.

Bây giờ thay đổi của bạn `button.c` đã được cam kết vào kho lưu trữ, nếu một người dùng khác kiểm tra một bản sao làm việc của `/calc`, Họ sẽ thấy các thay đổi trong phiên bản mới nhất của tập tin.

Giả sử bạn có một cộng tác viên, Sally, người kiểm tra ra một bản sao làm việc của `/calc` tại cùng một thời gian bạn đã làm. Khi bạn thực hiện thay đổi của bạn trên `button.c`, bản sao làm việc của Sally thì không thay đổi; Subversion chỉ thay đổi bản sao làm việc theo yêu cầu của người dùng.

Để mang lại cho dự án của mình được cập nhật, Sally có thể yêu cầu Subversion để *cập nhật* bản sao làm việc, bằng cách sử dụng lệnh Subversion **cập nhật**. Điều này sẽ kết hợp các thay đổi của bạn vào bản sao làm việc của mình, cũng như bất kỳ những người khác đã được cam kết kể từ khi cô kiểm tra nó ra.

Lưu ý rằng Sally đã không cần phải xác định các tập tin để cập nhật; Subversion sử dụng các thông tin trong thư mục `svn`, và biết thêm thông tin trong kho, để quyết định các tập tin cần được đưa đến nay.

### 2.3.2. URL kho chứa

Kho Subversion có thể được truy cập thông qua nhiều phương pháp khác nhau - trên đĩa địa phương, hoặc thông qua giao thức mạng khác nhau. Một vị trí kho, tuy nhiên, luôn luôn là một URL. Giảm đồ URL cho thấy phương pháp truy cập:

Giản đồ	Phương pháp truy cập
file://	Truy cập vào kho lưu trữ trực tiếp trên ổ đĩa cục bộ hoặc mạng.

### Bảng 2.1. URL truy cập kho lưu trữ

Đối với hầu hết các phần, URL của Subversion sử dụng các cú pháp tiêu chuẩn, cho phép đổi với tên máy chủ và số cổng được xác định như là một phần của URL. Các `file://` phương pháp tiếp cận thường được sử dụng cho truy cập địa phương, mặc dù nó có thể được sử dụng với các đường dẫn UNC đến một máy chủ mạng. Do đó, URL có dạng `file://hostname/path/to/repos`. Đối với các máy tính địa phương, phần `hostname` của URL được yêu cầu để vắng mặt hoặc `localhost`. Vì lý do này, các đường dẫn địa phương thường xuất hiện với ba dấu gạch chéo, `file:///path/to/repos`.

Ngoài ra, người sử dụng lược đồ `file://` trên nền tảng Windows sẽ cần phải sử dụng một cách không chính thức “tiêu chuẩn” cú pháp để truy cập vào kho lưu trữ trên cùng một máy, nhưng trên một ổ đĩa khác với ổ đĩa làm việc hiện tại của khách hàng. Một trong hai cú pháp đường dẫn URL sau đây sẽ làm việc X là ổ đĩa mà kho cư trú:

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

Lưu ý rằng một URL sử dụng dấu gạch chéo bình thường mặc dù hình thức bản địa (không phải URL) của một đường dẫn trên Windows sử dụng những dấu xò ngược.

Bạn có thể truy cập vào một kho lưu trữ FSFS thông qua một mạng chia sẻ, nhưng bạn *không thể* truy cập vào một kho lưu trữ BDB theo cách này.



#### Cảnh báo

Bạn không tạo ra hoặc truy cập vào một kho lưu trữ Berkeley DB trên một mạng chia sẻ. Nó *không thể* tồn tại trên một hệ thống tập tin từ xa. Không, ngay cả khi bạn có ổ đĩa mạng được ánh xạ một ký tự ổ đĩa. Nếu bạn cố gắng để sử dụng Berkeley DB trên một mạng chia sẻ, kết quả là không thể đoán trước - bạn có thể nhìn thấy các lỗi bí ẩn ngay lập tức, hoặc nó có thể là vài tháng trước khi bạn khám phá ra rằng cơ sở dữ liệu kho của bạn một bị hỏng cách tinh vi.

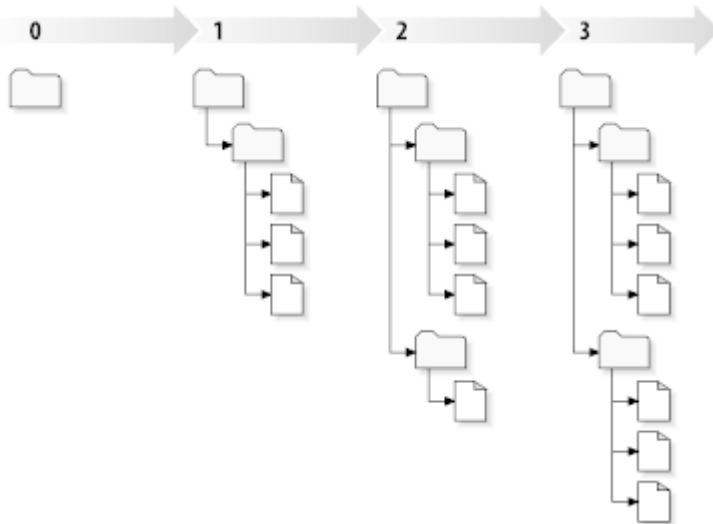
### 2.3.3. Sửa đổi

Một tác vụ **cam kết svn** có thể công bố thay đổi đối với bất kỳ số lượng các tập tin và thư mục như là một giao dịch nguyên tử duy nhất. Trong bản sao làm việc của bạn, bạn có thể thay đổi nội dung tập tin, tạo, xóa, đổi tên và sao chép các tập tin và thư mục, và sau đó cam kết các bộ hoàn chỉnh của các thay đổi như một đơn vị.

Trong kho lưu trữ, mỗi cam kết được đối xử như là một giao dịch nguyên tử: hoặc là tất cả các cam kết thay đổi diễn ra, hoặc không gì trong số đó diễn ra. Subversion giữ lại tính nguyên tử này trong khi đối mặt với tai nạn chương trình, sự cố hệ thống, các vấn đề mạng, và hành động khác của người dùng.

Mỗi khi kho chấp nhận một cam kết, điều này tạo ra một trạng thái mới của cây hệ thống tập tin, được gọi là *sửa đổi*. Mỗi sửa đổi được chỉ định một số tự nhiên duy nhất, lớn hơn số của sửa đổi trước. Việc sửa đổi ban đầu của một kho lưu trữ mới được tạo ra là số không, và bao gồm không có gì ngoài một thư mục gốc rỗng.

Một cách tốt đẹp để hình dung ra các kho lưu trữ như một loạt các cây. Hãy tưởng tượng một mảng các số sửa đổi, bắt đầu từ 0, kéo dài từ trái sang phải. Mỗi số sửa đổi có một cây treo hệ thống tập tin bên dưới nó, và mỗi cây là một “ảnh chụp” cách kho nhìn sau mỗi cam kết.



**Hình 2.7. Kho lưu trữ**

### Số Sửa Đổi Toàn Cục

Khác với các hệ thống quản lý phiên bản khác, con số chỉnh sửa của Subversion áp dụng đối với *toàn bộ cây*, không phải từng tập tin. Mỗi số phiên bản lựa chọn toàn bộ cây, một trạng thái đặc biệt của kho sau khi một số thay đổi được cam kết. Một cách khác để suy nghĩ về nó là sửa đổi N đại diện cho tình trạng của hệ thống tập tin kho lưu trữ sau khi các cam kết thứ N. Khi một người sử dụng Subversion nói về "sửa đổi 5 của `foo.c`", họ thật sự muốn nói là "`foo.c` xuất hiện trong bản chỉnh sửa 5." Nói chung, các phiên bản N và M của một tập tin *không* nhất thiết phải có sự khác biệt!

Điều quan trọng cần lưu ý rằng các bản sao làm việc không phải luôn luôn tương ứng với bất kỳ phiên bản duy nhất trong kho, chúng có thể chứa tập tin từ nhiều chỉnh sửa. Ví dụ, giả sử bạn kiểm tra ra một bản sao làm việc từ một kho lưu trữ có sửa đổi gần đây nhất là 4:

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

Tại thời điểm này, thư mục làm việc này tương ứng chính xác với sửa đổi 4 trong kho. Tuy nhiên, giả sử bạn thực hiện một sự thay đổi đến `button.c`, và cam kết thay đổi đó. Giả sử không có các cam kết đã diễn ra, cam kết của bạn sẽ tạo sửa đổi 5 của kho, và bản sao của bạn làm việc bây giờ sẽ như thế này:

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

Giả sử rằng, vào thời điểm này, Sally cam kết thay đổi trong `integer.c`, tạo ra sửa đổi 6. Nếu bạn sử dụng **svn cập nhật** để mang lại bản sao làm việc của bạn được cập nhật nhất, sau đó nó sẽ như thế này:

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

Thay đổi của Sally trong `integer.c` sẽ xuất hiện trong bản sao làm việc của bạn, và thay đổi của bạn sẽ vẫn có mặt trong `button.c`. Trong ví dụ này, các văn bản của `Makefile` là giống hệt nhau trong phiên bản 4, 5, và 6, nhưng Subversion sẽ đánh dấu bản sao của bạn làm việc của `Makefile` với phiên bản 6 để chỉ ra rằng

nó vẫn còn hiện tại. Vì vậy, sau khi bạn làm một cập nhật sạch sẽ ở trên cùng của bản sao làm việc của bạn, nó thường sẽ tương ứng chính xác với một bản sửa đổi trong kho.

### 2.3.4. Làm thế nào Các Bản Sao Làm Việc theo dõi các Kho

Đối với mỗi tập tin trong một thư mục làm việc, Subversion ghi lại hai phần thông tin thiết yếu trong khu vực hành chính `svn/` .:

- tập tin làm việc của bạn được dựa trên sửa đổi nào (điều này được gọi là *bản sửa đổi làm việc* của tập tin), và
- một ghi dấu thời gian khi các bản sao địa phương được cập nhật lần cuối bởi kho lưu trữ.

Với thông tin này, bằng cách nói chuyện với kho, Subversion có thể cho biết trong bốn trạng thái sau đây mà một tập tin làm việc có:

Không thay đổi, và hiện tại

Các tập tin là không thay đổi trong thư mục làm việc, và không có thay đổi đến tập tin đó đã được cam kết vào kho kể từ khi sửa đổi làm việc của nó. Một **cam kết** tập tin sẽ không làm gì, và một **cập nhật** của các tập tin sẽ không làm gì cả.

Thay đổi tại địa phương, và hiện tại

Các tập tin đã được thay đổi thư mục làm việc, và không có thay đổi đến tập tin đó được cam kết vào kho kể từ khi sửa đổi bản cơ sở của nó. Có thay đổi cục bộ đã không được cam kết vào kho lưu trữ, do đó một **cam kết** của tập tin sẽ thành công trong việc xuất bản các thay đổi của bạn, và một **cập nhật** của các tập tin sẽ không làm gì cả.

Không thay đổi, và quá ngày

Các tập tin đã không được thay đổi trong thư mục làm việc, nhưng nó đã được thay đổi trong kho. Tập tin cuối cùng sẽ được cập nhật, để làm cho nó hiện tại với bản chính sửa công cộng. Một **cam kết** tập tin sẽ không làm gì, và một **cập nhật** của tập tin sẽ đưa thay đổi mới nhất vào bản sao làm việc của bạn.

Thay đổi tại địa phương, và quá ngày

Tập tin đã được thay đổi trong thư mục làm việc, và trong kho. Một **cam kết** của tập tin sẽ thất bại với một lỗi *quá ngày*. Tập tin cần được cập nhật đầu tiên; lệnh **cập nhật** sẽ cố gắng hợp nhất những thay đổi công cộng này với những thay đổi của địa phương. Nếu Subversion không thể hoàn thành việc hợp nhất một cách chính đáng tự động, nó để lại cho người sử dụng để giải quyết cuộc xung đột.

## 2.4. Tóm tắt

Chúng tôi đã bao phủ một số khái niệm Subversion cơ bản trong chương này:

- Chúng tôi đã giới thiệu những khái niệm về các kho lưu trữ trung tâm, các bản sao làm việc trên máy khách, và một mạng các cây sửa đổi của kho lưu trữ.
- Chúng tôi đã nhìn thấy một số ví dụ đơn giản của hai cộng tác viên có thể sử dụng Subversion để xuất bản và nhận các thay đổi từ một người khác, sử dụng mô hình 'sao chép sửa đổi, hợp nhất'.
- Chúng tôi đã nói một chút về các cách Subversion theo dõi và quản lý thông tin trong một bản sao làm việc.

---

# Chương 3. Kho lưu trữ

Bất kể giao thức bạn sử dụng để truy cập vào kho lưu trữ của bạn, bạn luôn cần phải tạo ra ít nhất một kho lưu trữ. Điều này có thể được thực hiện với dòng lệnh Subversion hoặc với TortoiseSVN.

Nếu bạn đã không tạo ra một kho lưu trữ Subversion, đây là lúc để làm điều đó ngay bây giờ.

## 3.1. Tạo kho lưu trữ

You can create a repository with the FSFS backend or with the older Berkeley Database (BDB) format. The FSFS format is generally faster and easier to administer, and it works on network shares and Windows 98 without problems. The BDB format was once considered more stable simply because it has been in use for longer, but since FSFS has now been in use in the field for several years, that argument is now rather weak. Read [Choosing a Data Store](http://svnbook.red-bean.com/en/1.8/svn.reposadmin.planning.html#svn.reposadmin.basics.backends) [http://svnbook.red-bean.com/en/1.8/svn.reposadmin.planning.html#svn.reposadmin.basics.backends] in the Subversion book for more information.

### 3.1.1. Tạo một Repository với các dòng lệnh máy khách

1. Tạo một thư mục trống với tên SVN (ví dụ như D:\SVN\ ), Được sử dụng như là gốc cho tất cả các kho lưu trữ của bạn.
2. Tạo một thư mục khác MyNewRepository bên trong D:\SVN\ .
3. Mở dấu nhắc lệnh (hay DOS-Box), thay đổi vào D:\SVN\ và đánh vào

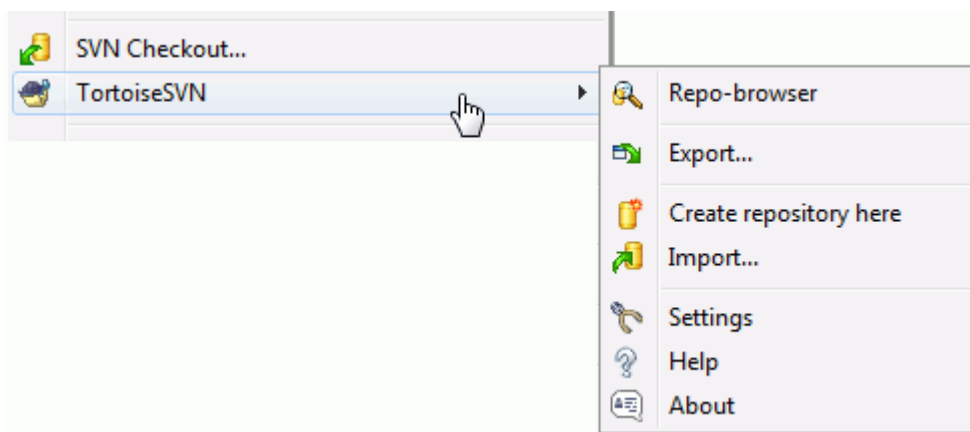
```
svnadmin create --fs-type bdb MyNewRepository
```

hoặc

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

Bây giờ bạn đã có một kho lưu trữ mới đặt tại D:\SVN\MyNewRepository .

### 3.1.2. Tạo Kho Lưu Trữ Với TortoiseSVN



**Hình 3.1. Các trình đơn TortoiseSVN cho các thư mục không phiên bản**

1. Mở cửa sổ explorer
2. Tạo một thư mục mới và đặt tên nó, ví dụ SVNRepository
3. Nhấp chuột phải vào thư mục mới được tạo ra và chọn TortoiseSVN → Tạo Kho Lưu Trữ ở đây ... .

Kho lưu trữ sau đó được tạo ra bên trong thư mục mới. *Bạn không thể tự chỉnh sửa các tập tin đó!* . Nếu bạn nhận được bất kỳ lỗi nào, hãy chắc chắn rằng thư mục rỗng và không được bảo vệ chống ghi.

Bạn cũng sẽ được hỏi xem bạn có muốn tạo ra một cấu trúc thư mục trong kho. Tìm hiểu về các lựa chọn bố trí ở [Phần 3.1.5, “Bố cục Kho chứa”](#) .

TortoiseSVN sẽ thiết lập một biểu tượng thư mục tùy chỉnh khi nó tạo ra một kho lưu trữ để bạn có thể xác định các kho địa phương dễ dàng hơn. Nếu bạn tạo một kho lưu trữ bằng cách sử dụng dòng lệnh máy khách chính thức biểu tượng thư mục này không được phân định.



### Mẹo

TortoiseSVN không còn cung cấp các tùy chọn để tạo ra các kho BDB, mặc dù bạn vẫn có thể sử dụng các dòng lệnh máy khách để tạo ra chúng. Kho FSFS thường dễ dàng hơn cho bạn để duy trì, và cũng có thể làm cho nó dễ dàng hơn cho chúng ta duy trì TortoiseSVN do vấn đề tương thích giữa các phiên bản BDB khác nhau.

TortoiseSVN hiện không hỗ trợ `file://` truy cập vào kho BDB do những vấn đề tương thích, mặc dù nó sẽ luôn luôn hỗ trợ định dạng kho lưu trữ này khi truy cập thông qua một máy chủ thông qua các giao thức `svn://`, `http://` hoặc `https://`.

Tất nhiên chúng tôi cũng khuyên bạn không sử dụng truy cập `file://` chút nào, ngoài mục đích thử nghiệm địa phương. Sử dụng một máy chủ an toàn hơn và đáng tin cậy hơn cho tất cả các sử dụng trừ dùng cho một nhà phát triển duy nhất .

### 3.1.3. Truy cập địa phương đến Kho

Để truy cập vào kho lưu trữ địa phương của bạn, bạn cần đường dẫn đến thư mục đó. Chỉ cần nhớ rằng Subversion hy vọng tất cả các đường dẫn kho lưu trữ trong mẫu `file:///C:/SVNRepository/`. Lưu ý việc sử dụng xuyên suốt các dấu gạch chéo về phía trước.

Để truy cập vào một kho lưu trữ nằm trên một mạng chia sẻ, bạn có thể sử dụng ánh xạ ổ đĩa, hoặc bạn có thể sử dụng đường dẫn UNC. Đối với các đường dẫn UNC, hình thức là `file://ServerName/path/to/Repos/` . Lưu ý rằng có chỉ có 2 dấu gạch chéo hàng đầu ở đây.

Trước SVN 1.2, đường dẫn UNC đã được đưa ra trong các hình thức mơ hồ hơn `file:///\\ServerName/path/to/Repos` . Hình thức này vẫn còn được hỗ trợ, nhưng không khuyến khích.



### Cảnh báo

Bạn không tạo ra hoặc truy cập vào một Berkeley DB kho lưu trữ trên một mạng chia sẻ. Nó *không thể* tồn tại trên một hệ thống tập tin từ xa. Không, ngay cả khi bạn có ổ đĩa mạng được ánh xạ một ký tự ổ đĩa. Nếu bạn cố gắng để sử dụng Berkeley DB trên một mạng chia sẻ, kết quả là không thể đoán trước - bạn có thể nhìn thấy các lỗi bí ẩn ngay lập tức, hoặc nó có thể là vài tháng trước khi bạn khám phá ra rằng cơ sở dữ liệu kho của bạn bị hỏng một cách tinh vi.

### 3.1.4. Truy cập vào một kho lưu trữ trên một mạng chia sẻ

Mặc dù theo lý thuyết có thể đặt một kho lưu trữ FSFS trên một mạng chia sẻ và có nhiều người dùng truy cập nó bằng cách sử dụng giao thức `file://`, điều này là chắc chắn *không* được khuyến khích. Trong thực tế, chúng tôi sẽ ngăn cản nó *mạnh mẽ*, và không hỗ trợ việc sử dụng đó.

Thứ nhất bạn đang đưa ra cho mỗi người sử dụng truy cập viết trực tiếp vào kho lưu trữ, do đó, bất kỳ người sử dụng có thể vô tình xóa toàn bộ kho lưu trữ hoặc làm cho nó không sử dụng được trong một số cách khác.

Thứ hai, không phải tất cả các giao thức chia sẻ tập tin mạng hỗ trợ các khóa mà Subversion yêu cầu, vì vậy bạn có thể tìm thấy kho lưu trữ của bạn bị hỏng. Nó có thể không xảy ra ngay lập tức, mà là vào một ngày hai người dùng sẽ cố gắng truy cập vào kho lưu trữ tại cùng một thời gian.



Thứ ba, các quyền tập tin đã được thiết lập chỉ cần như vậy. Bạn có thể chỉ là đi với nó trên một chia sẻ của Windows, nhưng đặc biệt khó khăn với SAMBA .

truy cập `file://` chỉ là để dành cho truy cập người sử dụng duy nhất của địa phương, đặc biệt là kiểm tra và gỡ lỗi. Khi bạn muốn chia sẻ các kho lưu trữ, bạn *thực sự* cần lập một máy chủ thích hợp, và nó gần như không khó khăn như bạn nghĩ. Đọc [Phần 3.5, “Truy cập Kho”](#) để được hướng dẫn về lựa chọn và thiết lập một máy chủ.

### 3.1.5. Bố cục Kho chứa

Trước khi bạn nhập dữ liệu vào kho lưu trữ trước tiên bạn nên suy nghĩ về cách bạn muốn tổ chức dữ liệu của bạn. Nếu bạn sử dụng một trong các bố trí được đề nghị, sau đó bạn sẽ có nó dễ dàng hơn nhiều.

Có một số cách, tiêu chuẩn được đề nghị cho việc tổ chức một kho lưu trữ. Hầu hết mọi người tạo ra một thư mục thân cây để giữ “dòng chính” của việc phát triển, một thư mục chi nhánh để chứa các bản sao chi nhánh, và một thư mục thẻ để chứa bản sao thẻ. Nếu kho lưu trữ chứa chỉ có một dự án, sau đó thường người ta tạo ra các thư mục cấp cao nhất này:

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

Bởi vì cách bố trí này thường được sử dụng, khi bạn tạo ra một kho lưu trữ mới sử dụng TortoiseSVN, nó cũng sẽ đề nghị tạo ra các cấu trúc thư mục đó cho bạn.

Nếu một kho lưu trữ có nhiều dự án, mọi người thường đánh chỉ số bố trí theo nhánh:

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

... hoặc bởi dự án:

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

Lập chỉ mục bởi dự án có ý nghĩa nếu dự án không liên quan chặt chẽ và mỗi dự án được kiểm tra ra một cách cá nhân. Đối với các dự án liên quan nơi bạn có thể muốn kiểm tra tất cả các dự án trong một bước, hoặc các dự án liên kết cùng nhau trong một gói phân phối duy nhất, nó thường tốt hơn để chỉ số theo chi nhánh. Bằng cách này bạn chỉ có một thân cây để kiểm tra ra, và các mối quan hệ giữa các tiểu dự án dễ dàng nhìn thấy được.

Nếu bạn áp dụng tiếp cận theo mức cao nhất `/Trunk/tags/branches`, không có gì để nói rằng bạn phải sao chép toàn bộ thân cây cho mỗi chi nhánh và thẻ, trong một số cách thì cấu trúc này cung cấp sự linh hoạt nhất.

Đối với các dự án không liên quan, bạn có thể thích sử dụng kho riêng biệt. Khi bạn cam kết thay đổi, đó là số phiên bản của toàn bộ kho với thay đổi, không phải là số phiên bản của dự án. Có 2 dự án không liên quan chia sẻ một kho lưu trữ có thể có nghĩa là những cách biệt lớn giữa các số sửa đổi. Các dự án Subversion và TortoiseSVN xuất hiện tại cùng một địa chỉ host, nhưng là kho hoàn toàn riêng biệt cho phép phát triển độc lập, và không có sự nhầm lẫn về số bản xây dựng.

Tất nhiên, bạn có thể tùy ý bỏ qua các bố cục chung này. Bạn có thể tạo ra bất kỳ loại biến thể, bất cứ điều gì làm việc tốt nhất cho bạn hoặc nhóm của bạn. Hãy nhớ rằng bất cứ điều gì bạn chọn, nó không phải là một cam kết lâu

dài. Bạn có thể tổ chức lại kho lưu trữ của bạn bất cứ lúc nào. Bởi vì các ngành và các thẻ là những thư mục bình thường, TortoiseSVN có thể di chuyển hoặc đổi tên chúng bất cứ cách nào bạn muốn.

Chuyển đổi từ một trong những bố trí khác chỉ là một vấn đề phát hành một loạt các di chuyển phía máy chủ. Nếu bạn không thích cách mà mọi thứ được tổ chức trong kho, chỉ cần sắp xếp các thư mục vòng vòng.

Vi vậy, nếu bạn chưa tạo ra một cấu trúc thư mục cơ bản bên trong kho của bạn, bạn nên làm điều đó bây giờ. Có hai cách để đạt được điều này. Nếu bạn chỉ đơn giản là muốn tạo ra một cấu trúc /Trunk/tags/branches, bạn có thể sử dụng trình duyệt kho để tạo ra ba thư mục (trong ba cam kết riêng biệt). Nếu bạn muốn tạo ra một hệ thống phân cấp sâu sắc hơn sau đó nó đơn giản tạo ra một cấu trúc thư mục trên đĩa đầu tiên và nhập khẩu nó trong một cam kết duy nhất, như thế này:

1. tạo ra một thư mục trống mới trên ổ đĩa cứng của bạn
2. tạo ra cấu trúc thư mục cấp cao nhất theo mong muốn của bạn bên trong thư mục đó - không đặt bất kỳ tập tin nào trong đó!
3. nhập khẩu cấu trúc này vào kho lưu trữ thông qua một kích chuột phải vào thư mục có chứa cấu trúc thư mục và chọn TortoiseSVN → **Nhập khẩu ...** Trong hộp thoại nhập khẩu, nhập URL đến kho lưu trữ của bạn và nhấn OK. Việc này sẽ nhập khẩu thư mục tạm thời của bạn vào thư mục gốc của kho để tạo ra bố trí kho lưu trữ cơ bản.

Lưu ý rằng tên của thư mục mà bạn đang nhập khẩu không xuất hiện trong kho lưu trữ, chỉ có nội dung của nó. Ví dụ, tạo cấu trúc thư mục sau đây:

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

Import C:\Temp\New vào thư mục gốc kho, sau đó sẽ như thế này:

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

## 3.2. Sao lưu kho lưu trữ

Bất kể kiểu kho nào bạn sử dụng, nó là cực kỳ quan trọng là bạn duy trì thường xuyên sao lưu, và rằng bạn xác minh các sao lưu. Nếu máy chủ thất bại, bạn có thể có thể truy cập vào một phiên bản gần đây của các tập tin của bạn, nhưng mà không có kho lưu trữ tất cả lịch sử của bạn bị mất mãi mãi.

cách đơn giản nhất (nhưng không được khuyến cáo) là chỉ để sao chép các thư mục kho lưu trữ trên các phương tiện sao lưu. Tuy nhiên, bạn có thể hoàn toàn chắc chắn rằng không có quá trình truy cập dữ liệu. Trong bối cảnh này, truy cập có nghĩa là *bất kỳ* truy cập nào. Một kho BDB được viết ngay cả khi hoạt động chỉ yêu cầu đọc, chẳng hạn như nhận tình trạng. Nếu kho của bạn được truy cập ở tất cả các bản sao, (trình duyệt web để mở, WebSVN, vv) sao lưu sẽ vô giá trị.

Phương pháp được đề nghị là chạy

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

để tạo ra một bản sao của kho lưu trữ của bạn một cách an toàn. Sau đó, sao lưu các bản sao. Các lựa chọn --clean-logs là không cần thiết, nhưng loại bỏ các tập tin tường trình dư thừa khi bạn sao lưu một kho lưu trữ BDB, có thể tiết kiệm một số không gian.

Công cụ svnadmin được cài đặt tự động khi bạn cài đặt máy khách Subversion dòng lệnh. Cách dễ nhất để có được điều này là kiểm chọn các tùy chọn để bao gồm các công cụ dòng lệnh khi cài đặt TortoiseSVN, nhưng nếu bạn thích, bạn có thể tải phiên bản mới nhất của công cụ dòng lệnh trực tiếp từ trang web [Subversion](http://subversion.apache.org/packages.html#windows) [http://subversion.apache.org/packages.html#windows].

### 3.3. kịch bản móc phía máy chủ

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

Các kịch bản này móc được thực hiện bởi các máy chủ nơi đặt các kho lưu trữ. TortoiseSVN cũng cho phép bạn cấu hình các lệnh móc phía máy khách được thực hiện tại địa phương sau các sự kiện nhất định. Xem **Phần 4.30.8, “Kịch Bản Hook Phía Máy Khách”** để biết thêm thông tin.

Các kịch bản móc mẫu có thể được tìm thấy trong thư mục `hooks` của kho. Những kịch bản mẫu phù hợp cho các máy chủ Unix / Linux, nhưng cần phải được sửa đổi nếu máy chủ của bạn dựa trên Windows. Móc có thể là một tập tin batch hoặc thực thi. Các mẫu dưới đây cho thấy một tập tin thực thi có thể được sử dụng để thực hiện một cái móc `pre-revprop-change`.

```
rem Chỉ cho phép các tin tường trình để được thay đổi.
if "%4" == "svn:log" exit 0
echo Thuộc tính '%4' không thể thay đổi được >&2
exit 1
```

Lưu ý rằng bất cứ điều gì gửi đến thiết bị xuất chuẩn được bỏ đi. Nếu bạn muốn có một thông báo xuất hiện trong hộp thoại Từ chối Cam kết, bạn phải gửi nó đến thiết bị lỗi chuẩn. Trong một tập tin batch việc này là đạt được bằng cách sử dụng `>&2`.



#### Thay Hooks

Nếu một kịch bản móc từ chối cam kết của bạn thì quyết định của nó là cuối cùng. Nhưng bạn có thể xây dựng một cơ chế ghi đè vào kịch bản bằng cách sử dụng kỹ thuật *Từ Kỳ Diệu*. Nếu kịch bản muốn từ chối các hoạt động đầu tiên quét các thông điệp tường trình cho một cụm từ thông qua đặc biệt, hoặc một cụm từ cố định hoặc có lẽ tên tập tin với một tiền tố. Nếu nó tìm thấy từ kỳ diệu đó nó cho phép các cam kết được tiến hành. Nếu cụm từ không được tìm thấy sau đó nó có thể chặn các cam kết với một thông điệp như “Bạn không nói từ kỳ diệu”. :-)

### 3.4. Các Liên kết Kiểm xuất

Nếu bạn muốn làm cho kho Subversion của bạn có sẵn cho những người khác, bạn có thể muốn bao gồm một liên kết đến từ trang web của bạn. Một cách để thực hiện điều này dễ tiếp cận hơn là bao gồm một *liên kết kiểm tra* cho những người sử dụng TortoiseSVN.

Khi bạn cài đặt TortoiseSVN, nó đăng ký giao thức mới `tsvn:`. Khi một người dùng TortoiseSVN nhấp vào liên kết như vậy, hộp thoại kiểm tra sẽ mở tự động với các URL kho lưu trữ đã được điền.

Để bao gồm một liên kết trong trang html của bạn, bạn cần phải thêm mã trông một cái gì đó như thế này:

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

Tất nhiên sẽ tốt hơn nếu bạn gửi kèm một hình ảnh phù hợp. Bạn có thể sử dụng *Lô gô của TortoiseSVN* [<http://tortoisesvn.net/images/TortoiseCheckout.png>] hoặc bạn có thể cung cấp hình ảnh của riêng bạn.

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

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

Bạn cũng có thể làm cho các liên kết đến một phiên bản cụ thể, ví dụ

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

### 3.5. Truy cập Kho

Để sử dụng TortoiseSVN (hoặc bất kỳ trình Subversion khác), bạn cần một nơi đặt kho của bạn. Bạn có thể lưu trữ kho lưu trữ của bạn tại địa phương và truy cập chúng bằng cách sử dụng giao thức `file://` hoặc bạn có thể đặt chúng trên một máy chủ và truy cập chúng với `http://` hoặc giao thức `svn://`. Hai giao thức máy chủ cũng có thể được mã hóa. Bạn sử dụng `https://` hoặc `svn+ssh://`, hoặc bạn có thể sử dụng `svn://` với SASL.

Nếu bạn đang sử dụng một dịch vụ lưu trữ công cộng như *Google Code* [<http://code.google.com/hosting/>] hoặc máy chủ của bạn đã được thiết lập bởi người khác sau đó có không có gì khác bạn cần phải làm. Di chuyển đọc theo đề *Chương 4, Hướng dẫn sử dụng hàng ngày*.

Nếu bạn không có một máy chủ và bạn làm việc một mình, hoặc nếu bạn chỉ là đánh giá Subversion và TortoiseSVN trong sự cô lập, các kho địa phương có lẽ là sự lựa chọn tốt nhất của bạn. Chỉ cần tạo ra một kho lưu trữ trên máy tính riêng của bạn như được mô tả trước đó trong *Chương 3, Kho lưu trữ*. Bạn có thể bỏ qua phần còn lại của chương này và đi trực tiếp *Chương 4, Hướng dẫn sử dụng hàng ngày* để tìm hiểu làm thế nào để bắt đầu sử dụng nó.

Nếu bạn đang suy nghĩ về việc thiết lập một kho lưu trữ nhiều người sử dụng trên một mạng chia sẻ, hãy nghĩ lại. Đọc *Phần 3.1.4, "Truy cập vào một kho lưu trữ trên một mạng chia sẻ"* để tìm hiểu lý do tại sao chúng tôi nghĩ rằng đây là một ý tưởng tồi. Thiết lập một máy chủ không phải là khó như nó có vẻ, và sẽ cung cấp cho bạn độ tin cậy và có thể tăng tốc độ tốt hơn.

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

Trong những ngày đầu của Subversion, thiết lập một máy chủ đòi hỏi một hiểu biết tốt về cấu hình máy chủ và trong các phiên bản trước của hướng dẫn sử dụng này chúng tôi bao gồm mô tả chi tiết làm thế nào để thiết lập một máy chủ. Kể từ đó mọi thứ đã trở nên dễ dàng hơn như hiện nay có một số trình cài đặt máy chủ đóng gói trước có sẵn trong đó hướng dẫn bạn thông qua quá trình cài đặt và cấu hình. Các liên kết này là dành cho một số những trình cài đặt chúng ta biết:

- *VisualSVN* [<http://www.visualsvn.com/server/>]
- *CollabNet* [<http://www.open.collab.net/products/subversion/whatsnew.html>]
- *UberSVN* [<http://www.ubersvn.com/>]

Bạn luôn luôn có thể tìm thấy các liên kết mới nhất ở website *Subversion* [<http://subversion.apache.org/packages.html>].

Bạn có thể tìm thêm các hướng dẫn Làm Thế Nào trên website *TortoiseSVN* [<http://tortoisesvn.net/usefultips.html>].

---

# Chương 4. Hướng dẫn sử dụng hàng ngày

Tài liệu này mô tả sử dụng hàng ngày của trình khách TortoiseSVN. Đó là *không phải* là giới thiệu về hệ thống kiểm soát phiên bản, và *không phải* là giới thiệu về Subversion (SVN). Nó giống như một nơi bạn có thể chuyển sang khi bạn gần biết những gì bạn muốn làm, nhưng không nhớ làm thế nào để làm điều đó.

Nếu bạn cần phần giới thiệu đến kiểm soát phiên bản với Subversion, thì đó chúng tôi khuyên bạn nên đọc cuốn sách tuyệt vời: [Kiểm soát phiên bản với Subversion](http://svnbook.red-bean.com/) [http://svnbook.red-bean.com/].

Tài liệu này cũng là một công việc đang tiến triển, chỉ như TortoiseSVN và Subversion. Nếu bạn tìm thấy bất kỳ sai lầm nào, xin vui lòng báo cáo đến danh sách gửi thư để chúng ta có thể cập nhật các tài liệu. Một số ảnh chụp màn hình trong Hướng dẫn Sử dụng Hàng ngày (DUG) có thể không phản ánh tình trạng hiện tại của phần mềm. Xin vui lòng tha thứ cho chúng tôi. Chúng tôi đang làm việc trên TortoiseSVN trong thời gian miễn phí của chúng tôi.

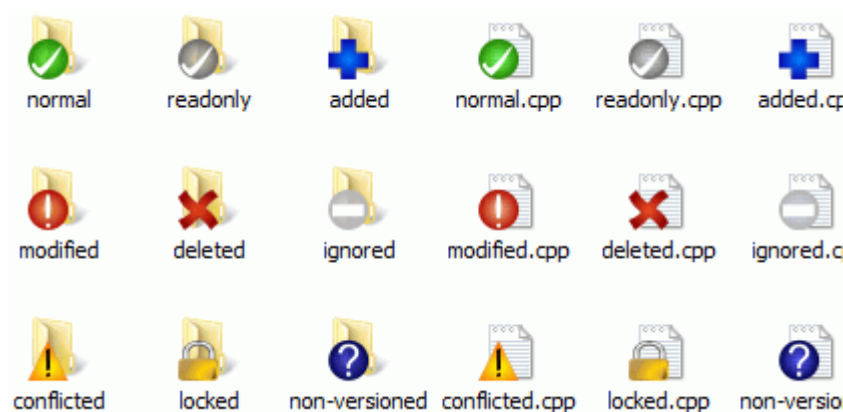
Để nhận được nhiều nhất của các hướng dẫn sử dụng hàng ngày:

- Bạn cần phải cài đặt TortoiseSVN đã.
- Bạn nên làm quen với các hệ thống kiểm soát phiên bản.
- Bạn nên biết những điều cơ bản của Subversion.
- Bạn nên thiết lập một máy chủ và / hoặc có truy cập đến một kho lưu trữ Subversion.

## 4.1. Các Tính Năng Chung

Phần này mô tả một số các tính năng của TortoiseSVN được áp dụng vào tất cả mọi thứ trong hướng dẫn sử dụng. Lưu ý rằng rất nhiều các tính năng này sẽ chỉ hiển thị trong một bản sao làm việc Subversion.

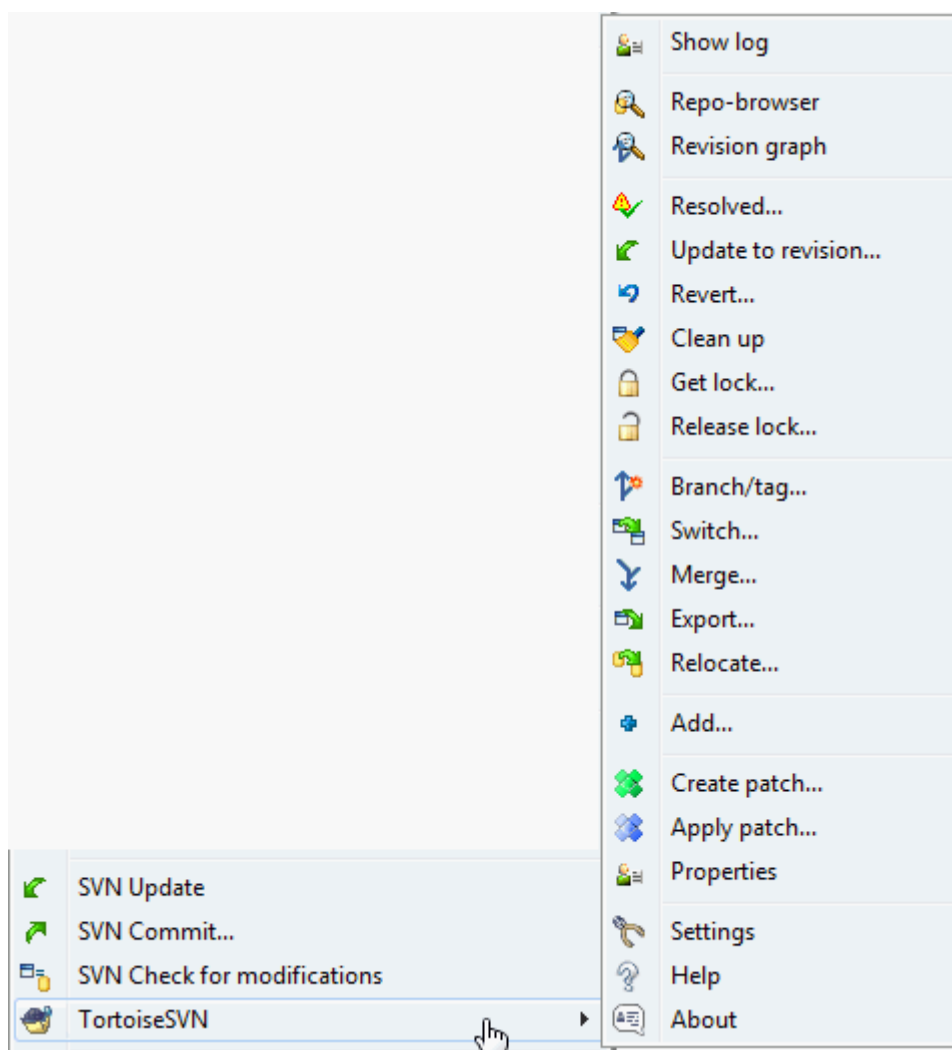
### 4.1.1. Biểu tượng Lớp phủ



**Hình 4.1. Explorer hiển thị lớp phủ biểu tượng**

Một trong các tính năng dễ thấy nhất của TortoiseSVN là các lớp phủ biểu tượng xuất hiện trên các tập tin trong bản sao làm việc của bạn. Những cái này chỉ cho bạn trong nháy mắt các tập tin của bạn đã được sửa đổi. Tham khảo [Phần 4.7.1, “Biểu tượng Lớp phủ”](#) để tìm hiểu các lớp phủ khác nhau đại diện những gì.

### 4.1.2. Tr đơn ngữ cảnh



**Hình 4.2. Trình đơn ngữ cảnh cho một thư mục dưới sự kiểm soát phiên bản**

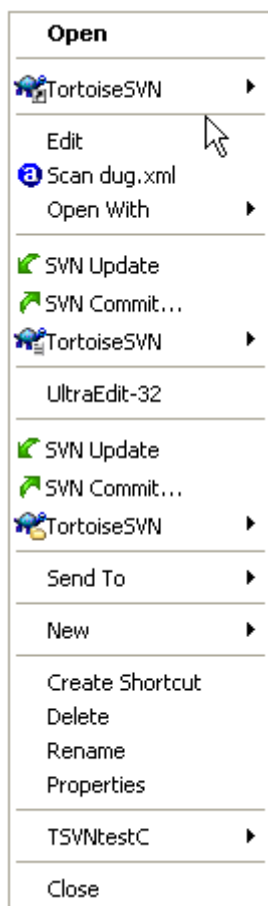
Tất cả các lệnh TortoiseSVN được gọi từ trình đơn ngữ cảnh của cửa sổ thám hiểm. Phần lớn trực tiếp nhìn thấy được, khi bạn bấm chuột phải vào một tập tin hoặc thư mục. Các lệnh mà có sẵn phụ thuộc vào việc các tập tin hoặc thư mục hoặc thư mục cha của nó là dưới sự kiểm soát phiên bản hoặc không. Bạn cũng có thể thấy trình đơn TortoiseSVN như là một phần của trình đơn khám phá tập tin.



### Mẹo

Một số lệnh rất hiếm khi được sử dụng chỉ có sẵn trong trình đơn ngữ cảnh mở rộng. Để đưa lên trình đơn ngữ cảnh mở rộng, giữ phím **Shift** khi bạn bấm chuột phải.

Trong một số trường hợp, bạn có thể thấy nhiều mục TortoiseSVN. Đây không phải là lỗi!

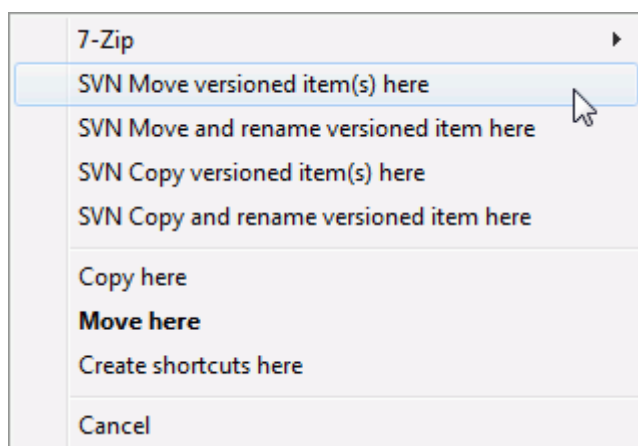


**Hình 4.3. Trình đơn explorer tập tin cho một phím tắt trong một thư mục được phiên bản**

Ví dụ này là dành cho một shortcut chưa phiên bản trong một thư mục được phiên bản, và trong trình đơn khám phá tập tin có *ba* mục cho TortoiseSVN. Một là cho các thư mục, một cho chính các shortcut, và thứ ba cho các đối tượng shortcut trỏ đến. Để giúp bạn phân biệt giữa chúng, các biểu tượng có một chi báo ở góc dưới bên phải để hiển thị cho biết các mục trình đơn là cho một tập tin, thư mục, shortcut hoặc cho nhiều mục được lựa chọn.

Nếu bạn đang sử dụng Windows 2000, bạn sẽ thấy rằng trình đơn ngữ cảnh được hiển thị dưới dạng văn bản, mà không có các biểu tượng trình đơn hiển thị ở trên. Chúng tôi nhận thức được rằng nó chạy được trong các phiên bản trước, nhưng Microsoft đã thay đổi cách xử lý biểu tượng cho Vista, đòi hỏi chúng ta sử dụng một phương pháp hiển thị khác nhau mà tiếc là không hoạt động trên Windows 2000.

### 4.1.3. Kéo và thả



**Hình 4.4. Kéo phải trình đơn cho một thư mục dưới sự kiểm soát phiên bản**

Các lệnh khác như xử lý kéo, khi bạn kéo phải các tập tin hoặc thư mục vào một vị trí mới bên trong bản sao làm việc hoặc khi bạn kéo phải một tập tin hoặc thư mục không được phiên bản vào một thư mục được kiểm soát phiên bản.

#### 4.1.4. Phím tắt phổ biến

Một số tác vụ phổ biến có các phím tắt của Windows, nhưng không xuất hiện trên các nút hoặc trong các trình đơn. Nếu bạn không thể nghĩ ra làm thế nào để làm một cái gì đó rõ ràng, giống như làm mới một cảnh xem, hãy kiểm tra ở đây.

F1

Giúp đỡ, tất nhiên.

F5

Làm mới cảnh xem hiện tại. Điều này có lẽ là lệnh một phím hữu ích nhất. Ví dụ ... Trong Explorer, điều này sẽ làm mới lớp phủ biểu tượng trên bản sao làm việc của bạn. Trong hộp thoại cam kết nó sẽ quét các bản sao làm việc để xem những gì có thể cần phải được cam kết. Trong hộp thoại tương trình sửa đổi, nó sẽ liên hệ với các kho lưu trữ một lần nữa để kiểm tra các thay đổi gần đây.

Ctrl-A

Chọn tất cả. Điều này có thể được sử dụng nếu bạn nhận được một thông báo lỗi và muốn sao chép và dán vào một email. Sử dụng Ctrl-A để chọn các thông báo lỗi và sau đó ...

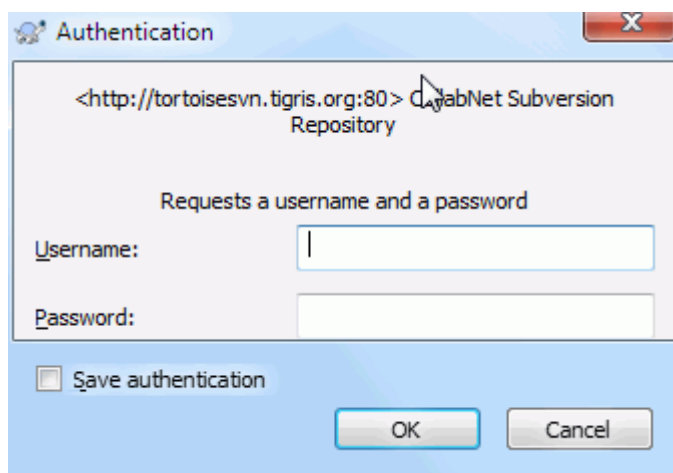
Ctrl-C

... Sao chép văn bản đã chọn.

#### 4.1.5. Xác thực

Nếu các kho lưu trữ mà bạn đang cố gắng truy cập có mật khẩu bảo vệ, một hộp thoại xác thực sẽ xuất hiện.





**Hình 4.5. Hộp thoại xác thực**

Nhập tên người dùng và mật khẩu của bạn. Hộp kiểm sẽ làm cho TortoiseSVN lưu trữ các thông tin trong thư mục mặc định của Subversion: `%APPDATA%\Subversion\auth` trong ba thư mục con:

- `svn.simple` có chứa thông tin quan trọng cho xác thực cơ bản (tên đăng nhập / mật khẩu). Lưu ý rằng mật khẩu được lưu trữ bằng cách sử dụng API WinCrypt, không ở dạng văn bản đơn giản.
- `svn.ssl.server` có giấy chứng nhận máy chủ SSL.
- `svn.username` chứa các thông tin xác thực chỉ cần tên người dùng (không có mật khẩu cần thiết).

Nếu bạn muốn xóa bộ nhớ cache xác thực cho tất cả các máy chủ, bạn có thể làm như vậy ở trang **Lưu dữ liệu** của hộp thoại thiết lập TortoiseSVN. Nút bấm đó sẽ dọn dẹp tất cả các dữ liệu xác thực được lưu trữ từ thư mục `auth` của Subversion, cũng như bất kỳ dữ liệu xác thực được lưu trữ trong registry các phiên bản trước của TortoiseSVN. Tham khảo [Phần 4.30.6, “Thiết Lập Cho Dữ Liệu Đã Lưu”](#).

Nếu bạn muốn dọn dẹp xác thực cho một lĩnh vực chỉ thì bạn sẽ phải thâm nhập vào những thư mục, tìm tập tin trong đó có các thông tin mà bạn muốn xóa và xóa các tập tin.

Một số người muốn có các dữ liệu xác thực đã bị xóa khi đăng xuất khỏi Windows, hoặc tắt máy tính. Cách để làm điều đó là sử dụng một kịch bản tắt máy để xóa các thư mục `%APPDATA%\Subversion\auth`, ví dụ như:

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

Bạn có thể tìm thấy một mô tả làm thế nào để cài đặt các kịch bản như vậy tại <http://www.windows-help-central.com/windows-shutdown-script.html>.

Để biết thêm thông tin về việc làm thế nào để thiết lập máy chủ của bạn để xác thực và kiểm soát truy cập, tham khảo [Phần 3.5, “Truy cập Kho”](#).

#### 4.1.6. Tối đa hoá cửa sổ

Nhiều người trong số các hộp thoại TortoiseSVN có rất nhiều thông tin để hiển thị, nhưng nó thường là hữu ích để tối đa hóa chỉ có chiều cao, chỉ có chiều rộng, chứ không phải là tối đa hóa để điền vào màn hình. Là một tiện nghi, có các phím tắt cho điều này trên **Tối đa hóa** nút. Sử dụng nút chuột giữa để tối đa hóa theo chiều dọc, và chuột phải để tối đa hóa theo chiều ngang.

## 4.2. Nhập khẩu dữ liệu vào một kho chứa

### 4.2.1. Nhập khẩu

Nếu bạn đang nhập vào một kho lưu trữ hiện tại đã có một số dự án, thì cấu trúc kho lưu trữ đã được quyết định. Nếu bạn đang nhập dữ liệu vào một kho lưu trữ mới, thì nó đáng để dành thời gian suy nghĩ về nó sẽ được tổ chức như thế nào. Đọc [Phần 3.1.5, “Bổ cục Kho chứa”](#) để được tư vấn thêm.

Phần này mô tả các lệnh nhập khẩu Subversion được thiết kế để nhập khẩu một hệ thống phân cấp thư mục vào kho lưu trữ trong một lần. Mặc dù nó làm được việc, nó có một số thiếu sót:

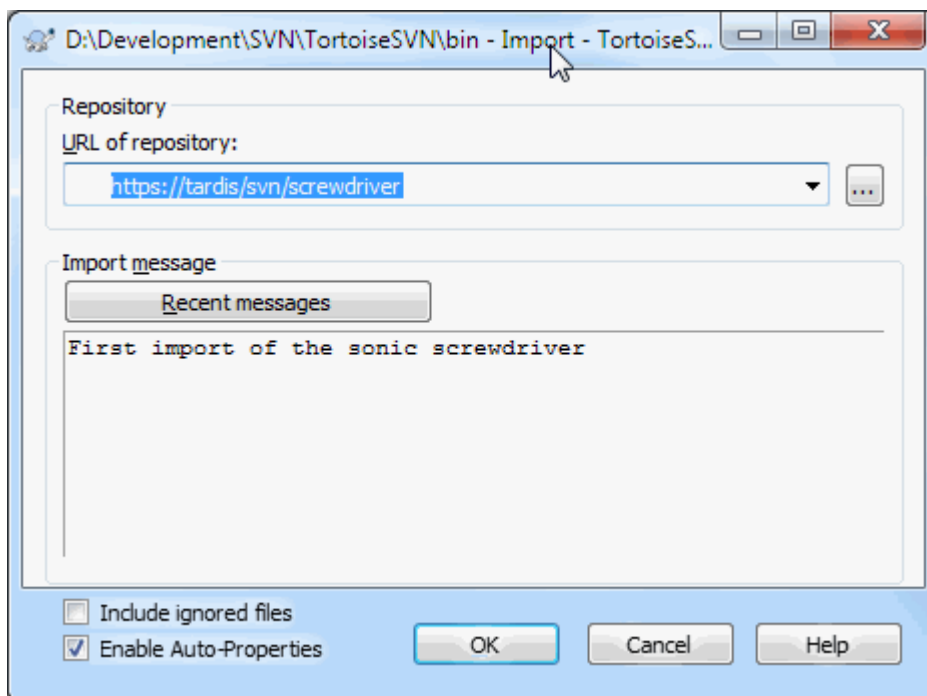
- Không có cách nào để lựa chọn các tập tin và thư mục để bao gồm, ngoài việc sử dụng các thiết lập bỏ qua toàn cục.
- Các thư mục được nhập khẩu không trở thành một bản sao làm việc. Bạn phải làm một kiểm tra ra để sao chép các tập tin từ máy chủ.
- Thật dễ dàng để nhập khẩu với các cấp độ thư mục sai trong kho.

Vì những lý do này, chúng tôi khuyên bạn không sử dụng lệnh nhập khẩu chút nào mà nên làm theo phương pháp hai bước được mô tả trong [Phần 4.2.2, “Nhập khẩu Tại chỗ”](#), Trừ khi bạn đang thực hiện bước đơn giản để tạo ra một cấu trúc `/trunk/tags/branches` ban đầu trong kho lưu trữ của bạn. Bởi vì bạn đang ở đây, đây là cách việc nhập khẩu cơ bản hoạt động...

Trước khi bạn nhập khẩu dự án của bạn vào một kho lưu trữ, bạn nên:

1. Hủy bỏ tất cả các tập tin không cần thiết để xây dựng dự án (các tập tin tạm thời, các tập tin được tạo ra bởi trình biên dịch, ví dụ như: \*.obj, những chương trình nhị phân được biên dịch, ...)
2. Tổ chức các tập tin trong thư mục và thư mục con. Mặc dù nó có thể đổi tên / di chuyển các tập tin sau đó, chúng tôi đề cao việc có được cấu trúc của dự án của bạn ngay ngắn trước khi nhập khẩu!

Bây giờ chọn thư mục cấp cao nhất của cấu trúc thư mục dự án của bạn trong cửa sổ thám hiểm và click chuột phải để mở trình đơn ngữ cảnh. Chọn lệnh `TortoiseSVN` → `Nhập khẩu ...` để gọi lên một hộp thoại:



**Hình 4.6. Hộp thoại nhập khẩu**

Trong hộp thoại này, bạn phải nhập vào URL của vị trí kho lưu trữ mà bạn muốn nhập khẩu dự án của bạn. Nó là rất quan trọng để nhận ra rằng các thư mục địa phương mà bạn đang nhập khẩu không tự nó xuất hiện trong kho, mà chỉ có nội dung của nó. Ví dụ nếu bạn có một cấu trúc:

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

and you import C:\Projects\Widget into `http://mydomain.com/svn/trunk` thì bạn có thể ngạc nhiên khi thấy rằng các thư mục con của bạn đi thẳng vào thân cây hơn là trong một thư mục con. Bạn cần phải chỉ định thư mục con như là một phần của URL, `http://mydomain.com/svn/trunk/Widget-X`. Lưu ý rằng các lệnh nhập khẩu sẽ tự động tạo ra các thư mục con trong kho nếu chúng không tồn tại.

Các thông báo nhập khẩu được sử dụng như một thông điệp tường trình.

Theo mặc định, các tập tin và thư mục mà phù hợp với các mô hình bỏ qua toàn cục *không* được nhập khẩu. Để ghi đề lên hành vi này, bạn có thể sử dụng hộp kiểm **Bao gồm các file bị bỏ qua**. Tham khảo [Phần 4.30.1, “Các Thiết Lập Tổng Qu”](#) để biết thêm thông tin về việc thiết lập một mô hình bỏ qua toàn cục.

Ngay sau khi bạn nhấn **OK** TortoiseSVN nhập khẩu các cây thư mục hoàn chỉnh bao gồm tất cả các tập tin vào kho lưu trữ. Dự án hiện đang được lưu trữ trong kho dưới sự kiểm soát phiên bản. Xin lưu ý rằng thư mục mà bạn đã nhập khẩu *KHÔNG* theo phiên bản kiểm soát! Để có được một *bản sao làm việc* có kiểm soát phiên bản bạn cần phải làm một Kiểm Xuất của các phiên bản bạn vừa nhập khẩu. Hoặc đọc để tìm hiểu làm thế nào để nhập khẩu một thư mục tại chỗ.

## 4.2.2. Nhập khẩu Tại chỗ

Giả sử bạn đã có một kho lưu trữ, và bạn muốn thêm vào một cấu trúc thư mục mới cho nó, chỉ cần làm theo các bước sau:

1. Sử dụng trình duyệt kho để tạo ra một thư mục dự án mới trực tiếp trong kho. Nếu bạn đang sử dụng một bộ cục tiêu chuẩn, có thể bạn sẽ muốn tạo cái này như là một thư mục con của thân cây chứ không phải trong thư mục gốc kho. Trình duyệt kho lưu trữ cho thấy cấu trúc kho lưu trữ giống như Windows Explorer, vì vậy bạn có thể nhìn thấy mọi thứ được tổ chức như thế nào.
2. Kiểm tra các thư mục mới trên đầu trang của thư mục mà bạn muốn nhập khẩu. Bạn sẽ nhận được một cảnh báo rằng các thư mục địa phương không rỗng. Bỏ qua cảnh báo đó. Bây giờ bạn có một thư mục cấp cao được phiên bản với nội dung không được phiên bản.
3. Sử dụng TortoiseSVN → **Thêm ...** trên thư mục được phiên bản này là thêm một số hoặc tất cả nội dung. Bạn có thể thêm và loại bỏ các tập tin, đặt thuộc tính `svn:ignore` trên các thư mục và thực hiện bất kỳ thay đổi khác mà bạn cần.
4. Cam kết các thư mục cấp cao nhất, và bạn có một cây mới được phiên bản, và một bản sao làm việc địa phương, được tạo ra từ thư mục hiện tại của bạn.

## 4.2.3. Các tập tin Đặc biệt

Đôi khi bạn cần phải có một tập tin dưới sự kiểm soát phiên bản, trong đó có chứa các dữ liệu người dùng cụ thể. Điều đó có nghĩa là bạn có một tập tin mà mọi nhà phát triển / người dùng cần phải sửa đổi để phù hợp với thiết lập địa phương của anh / cô ta. Tuy nhiên, phiên bản một tập tin như vậy là khó khăn bởi vì mỗi người sử dụng sẽ cam kết thay đổi của anh / cô ta mỗi khi vào kho lưu trữ.

Trong trường hợp này, chúng tôi đề nghị sử dụng các tập tin *mẫu*. Bạn tạo một tập tin mà chứa tất cả các dữ liệu mà nhà phát triển của bạn sẽ cần, thêm tập tin đó vào để kiểm soát phiên bản và để cho các nhà phát triển kiểm tra tập tin này ra. Sau đó, mỗi nhà phát triển *tạo một bản sao* của tập tin đó và đổi tên bản sao đó. Sau đó, sửa đổi các bản sao không phải là một vấn đề nữa.

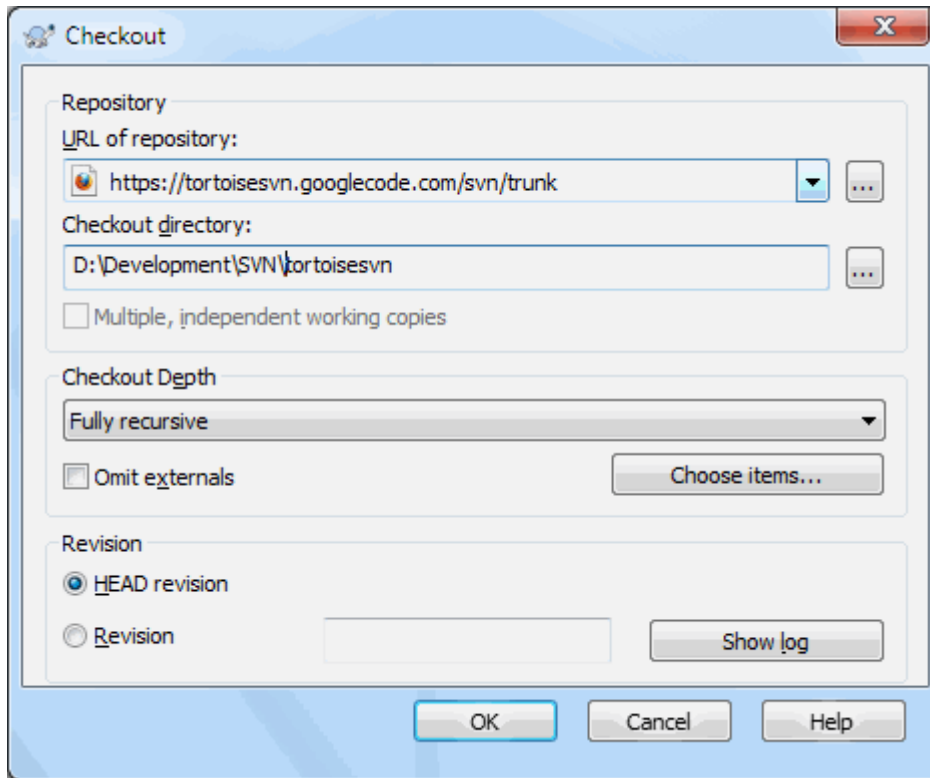
Ví dụ, bạn có thể có một nhìn vào kịch bản xây dựng của TortoiseSVN. Nó gọi một tập tin có tên `TortoiseVars.bat` không tồn tại trong kho. Chỉ có tập tin `TortoiseVars.tmpl`. `TortoiseVars.tmpl` là tập tin mẫu mà mỗi nhà phát triển đã tạo ra một bản sao từ và đổi tên file đó để `TortoiseVars.bat`. Bên trong tập tin đó, chúng tôi đã thêm ý kiến để người sử dụng sẽ thấy những dòng họ có chỉnh sửa và thay đổi theo để thiết lập địa phương của họ để có được nó làm việc.

Vì vậy, không làm phiền người sử dụng, chúng tôi cũng thêm vào các tập tin `TortoiseVars.bat` vào danh sách bỏ qua của thư mục mẹ của nó, tức là chúng ta đã thiết lập thuộc tính Subversion `svn:ignore` để bao gồm tên tập tin đó. Bằng cách đó nó sẽ không hiển thị như là không được phiên bản trên tất cả các cam kết.

### 4.3. Kiểm Xuất Một Bản Sao Làm Việc

Để có được một bản sao làm việc, bạn cần phải làm một *Kiểm Xu* từ một kho lưu trữ.

Chọn một thư mục trong cửa sổ thám hiểm mà bạn muốn đặt bản sao làm việc của bạn. Nhấp chuột phải để bật lên trình đơn ngữ cảnh và chọn lệnh `TortoiseSVN` → `Kiểm Xuất...`, sẽ trả về hộp thoại sau đây:



**Hình 4.7. Hộp thoại Kiểm Xu**

Nếu bạn nhập vào một tên thư mục không tồn tại, thì một thư mục với tên đó được tạo ra.

#### 4.3.1. Độ sâu Kiểm xu

Bạn có thể chọn *chiều sâu* bạn muốn kiểm xuất, cho phép bạn chỉ định độ sâu đệ quy vào các thư mục con. Nếu bạn muốn chỉ là một vài phần của một cây lớn, Bạn có thể kiểm xu các thư mục cấp cao nhất, sau đó cập nhật các thư mục được lựa chọn theo cách đệ quy.

Đệ quy hoàn toàn

Kiểm xuất toàn bộ cây, bao gồm tất cả các thư mục con và thư mục cấp dưới.

Các mục con gần nhất, bao gồm các thư mục

Kiểm xu thư mục chỉ định, bao gồm tất cả các tập tin và thư mục con, nhưng không sinh sôi thư mục con.

Chỉ có tập tin con

Kiểm xuất các thư mục chỉ định, bao gồm tất cả các tập tin nhưng không Kiểm xuất bất kỳ thư mục con nào.

Chỉ có mục này

Chỉ kiểm xu các thư mục. Không sinh sôi trong nó với các tập tin hoặc thư mục con.

Bản sao làm việc

Giữ lại độ sâu quy định tại bản sao làm việc. Tùy chọn này không được sử dụng trong hộp thoại kiểm tra, nhưng nó là mặc định trong tất cả các hộp thoại khác trong đó có một thiết lập độ sâu.

#### Loại trừ

Được sử dụng để giảm độ sâu bản sao làm việc sau khi một thư mục đã được sinh sôi. Tùy chọn này chỉ có sẵn trong hộp thoại **Cập nhật cho sửa đổi**.

Để dễ dàng lựa chọn các chỉ mục mà bạn muốn cho kiểm xuất và ép buộc bản sao làm việc chỉ giữ những mục đó, nhấp vào nút bấm **Chọn mục ...**. Điều này sẽ mở ra một hộp thoại mới, nơi bạn có thể kiểm tra tất cả các mục mà bạn muốn sao chép của bạn và bỏ chọn tất cả các mục bạn không muốn. Bản sao làm việc sau đó được biết đến như một kiểm xuất thua thốt. Bản cập nhật của bản sao làm việc đó sẽ không nạp các tập tin và thư mục bị thiếu nhưng chỉ cập nhật những gì bạn đã có trong bản sao làm việc của bạn.

Nếu bạn kiểm xuất một bản làm việc rời rạc (có nghĩa là, bằng cách chọn mục gì đó không phải đệ quy hoàn toàn cho độ sâu kiểm xuất), bạn có thể dễ dàng thêm hoặc bớt các thư mục con sau này bằng cách dùng một trong các cách sau.

#### 4.3.1.1. Cập Nhật Rời Rạc sử dụng Cập Nhật đến Bản Chính Sửa

Nhấp phải trên thư mục được kiểm xuất, sau đó dùng TortoiseSVN → **Cập Nhật đến Bản Sửa Đổi** và chọn **Chọn các mục....** Việc này sẽ mở ra cùng hộp thoại đã có trong kiểm xuất gốc và cho phép bạn chọn hoặc bỏ chọn các mục để bao gồm trong kiểm xuất. Phương thức này rất linh hoạt nhưng có thể chậm bởi các mục trong thư mục được cập nhật.

#### 4.3.1.2. Cập Nhật Rời Rạc sử dụng Trình Duyệt Kho

Nhấp phải trên thư mục được kiểm xuất, sau đó dùng TortoiseSVN → **Trình Duyệt Kho** để hiện trình duyệt kho. Tìm thư mục con mà bạn muốn thêm vào bản làm việc, sau đó dùng **Trình Đơn Ngữ Cảnh** → **Cập nhật mục đến bản chính sửa....**

#### 4.3.1.3. Cập Nhật Rời Rạc sử dụng Kiểm Tra Sửa Đổi

Trong hộp thoại kiểm tra sửa đổi, trước hết nhấn **shift** và bấm vào nút **Kiểm tra kho lưu trữ**. Hộp thoại sẽ hiển thị tất cả các tập tin và thư mục trong kho lưu trữ mà bạn đã không kiểm xuất dưới dạng thêm vào từ xa. Nhấp chuột phải v Tr Cmenuchoice >.

Tính năng này rất hữu ích khi bạn chỉ muốn để kiểm tra các bộ phận của một cây lớn, nhưng bạn muốn sự tiện lợi của việc cập nhật một bản sao làm việc duy nhất. Giả sử bạn có một cây lớn trong đó có thư mục con **Project01** đến **Project99**, và bạn chỉ muốn kiểm tra **Project03**, **Project25** và **Project76/SubProj**. Sử dụng các bước sau:

1. Kiểm xu thư mục mẹ với độ sâu “Chỉ có mục này” Bây giờ bạn có một thư mục trống cấp cao nhất.
2. Chọn thư mục mới và sử dụng **TortoiseSVN** → **Trình duyệt khoa** để hiển thị các nội dung của kho lưu trữ.
3. Nhấp chuột phải vào **Project03** và **Trình đơn ngữ cảnh** → **Cập nhật mục để sửa đổi ...**. Giữ các thiết lập mặc định và nhấp vào **OK**. Bây giờ bạn có thư mục được sinh sôi đầy đủ.

Lặp lại quá trình này cho **Project25**.

4. Hướng đến **Project76/SubProj** và làm như vậy. Lưu ý lúc này thư mục **Project76** không có nội dung trừ **SubProj**, mà chính nó là được sinh sôi đầy đủ. Subversion đã tạo ra các thư mục trung gian cho bạn mà không sinh sôi trong nó.



### Thay đổi chiều sâu bản sao làm việc

Một khi bạn đã kiểm xuất một bản sao làm việc đến độ sâu cụ thể, bạn có thể thay đổi độ sâu sau này để có được nội dung nhiều hơn hoặc ít hơn bằng cách sử dụng **Trình đơn ngữ cảnh** → **Cập nhật mục để sửa đổi ...**. Trong hộp thoại đó, hãy chắc chắn để đánh dấu hộp kiểm **Làm độ sâu dính**.



## Sử dụng một máy chủ cũ

Máy chủ trước-1.5 không hiểu được yêu cầu chiều sâu của bản sao làm việc, vì vậy chúng có thể không phải lúc nào cũng đối phó với các yêu cầu đó một cách hiệu quả. Các lệnh sẽ vẫn làm việc, nhưng một máy chủ cũ có thể gửi tất cả các dữ liệu, để lại khách hàng để lọc ra những gì không cần thiết, có thể có nghĩa là gây ra rất nhiều lưu lượng mạng. Nếu có thể bạn nên nâng cấp máy chủ của bạn tới ít nhất là 1.5.

Nếu dự án có chứa các tài liệu tham khảo đến các dự án bên ngoài mà bạn *không* muốn kiểm xuất tại cùng một thời gian, sử dụng hộp kiểm **Bỏ qua thành phần ngoài** .



## Quan trọng

Nếu phần **Bỏ qua thành phần ngoài** được kiểm, hoặc nếu bạn muốn tăng giá trị chiều sâu, bạn sẽ phải thực hiện cập nhật bản sao làm việc của bạn bằng cách sử dụng TortoiseSVN → **Cập nhật đến bản sửa đổi ...** thay vì TortoiseSVN → **Cập nhật** . Bản cập nhật tiêu chuẩn sẽ bao gồm tất cả các phần bên ngoài và giữ độ sâu hiện tại.

Chúng tôi đề nghị bạn chỉ nên kiểm xuất phần *thân cây* của cây thư mục, hoặc thấp hơn. Nếu bạn chỉ định đường dẫn cha mẹ của cây thư mục trong URL sau đó bạn có thể kết thúc với một đĩa cứng đầy đủ bởi vì khi bạn sẽ nhận được một bản sao của toàn bộ cây kho lưu trữ bao gồm tất cả các ngành và thẻ của dự án của bạn!



## Xuất khẩu

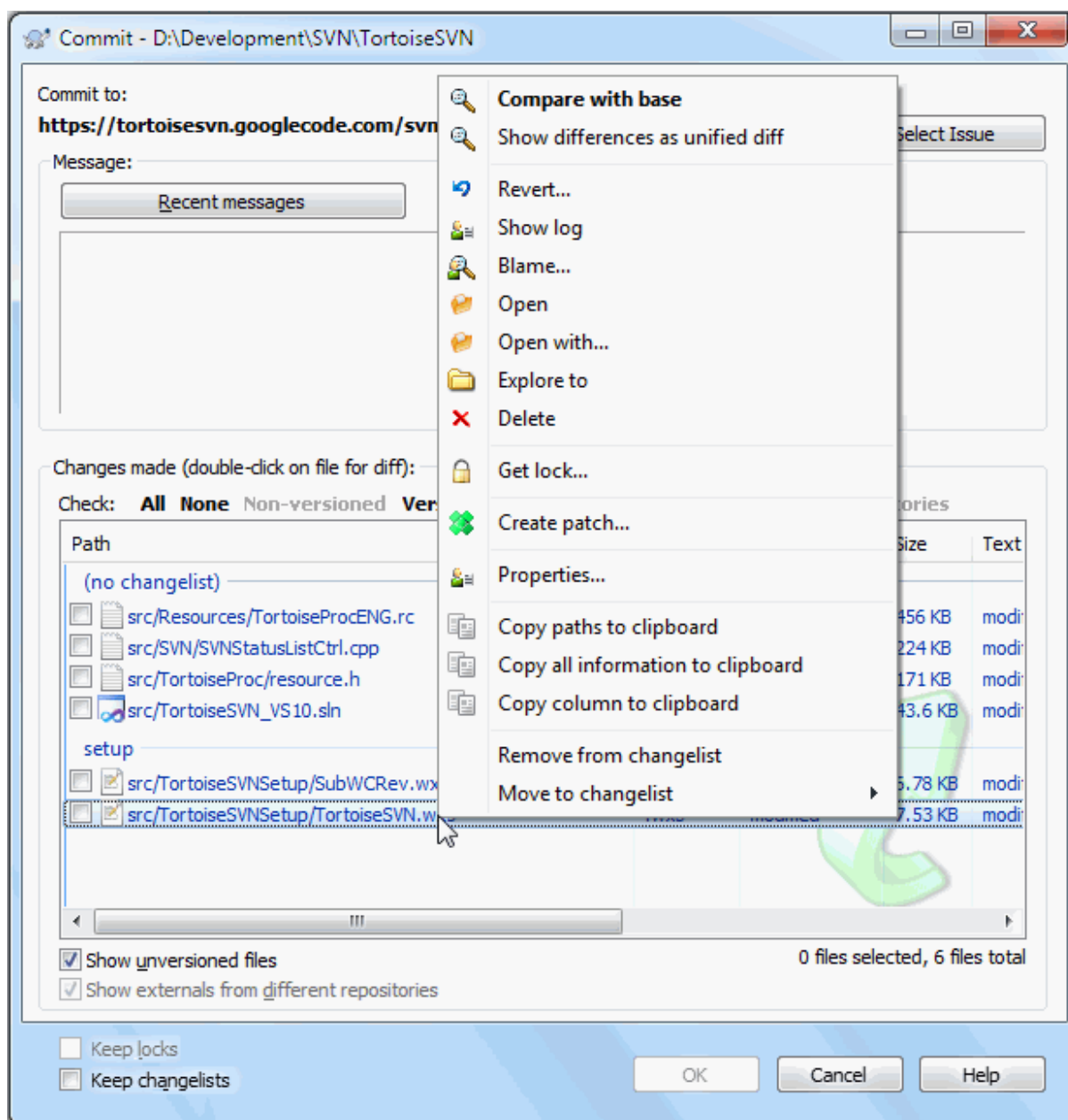
Đôi khi bạn có thể muốn tạo một bản sao địa phương mà không có bất kỳ của những thư mục `.svn`, ví dụ như để tạo ra một tarball nén của mã nguồn của bạn. Đọc [Phần 4.26, “Xuất khẩu một bản sao làm việc Subversion”](#) để tìm hiểu làm thế nào để làm điều đó.

## 4.4. Cam kết các Thay đổi của bạn vào Kho

Gửi thay đổi bạn đã thực hiện đến bản làm việc của bạn được gọi là *cam kết* các thay đổi. Nhưng trước khi bạn cam kết bạn phải đảm bảo rằng bản sao làm việc của bạn được cập nhật. Bạn có thể sử dụng TortoiseSVN → **Cập nhật trực tiếp**. Hoặc bạn có thể sử dụng TortoiseSVN → **Kiểm tra cho sửa đổi** trước, để xem các tập tin đã thay đổi cục bộ hoặc trên máy chủ.

### 4.4.1. Hộp Thoại Cam kết

Nếu bản sao làm việc của bạn được cập nhật và không có xung đột, bạn đã sẵn sàng cam kết thay đổi của bạn. Chọn bất kỳ tập tin và / hoặc thư mục mà bạn muốn cam kết, sau đó TortoiseSVN → **Cam kết ...** .



**Hình 4.8. Hộp thoại Cam kết**

Hộp thoại cam kết sẽ cho bạn thấy tất cả các tập tin thay đổi, bao gồm cả các tập tin được thêm, xoá và không được phiên bản. Nếu bạn không muốn một tập tin thay đổi được cam kết, chỉ cần bỏ chọn tập tin đó. Nếu bạn muốn để bao gồm một tập tin không được phiên bản, chỉ cần kiểm tra tập tin để thêm vào cam kết.

Để biết thêm thông tin về màu sắc và lớp phủ của các mục tùy theo trạng thái của chúng, xin xem [Phần 4.7.4, “Trạng thái địa phương và từ xa”](#).

Các mục đã được chuyển sang một đường dẫn kho lưu trữ khác cũng được chỉ định sử dụng một đánh dấu (s) . Bạn có thể đã chuyển một cái gì đó trong khi làm việc trên một chi nhánh và quên trở về với thân cây. Đây là dấu hiệu cảnh báo của bạn!



### Cam kết các tập tin hoặc thư mục?

Khi bạn cam kết các tập tin, hộp thoại cam kết chỉ hiển thị các tập tin mà bạn đã chọn. Khi bạn cam kết một thư mục hộp thoại cam kết sẽ tự động lựa chọn các tập tin thay đổi. Nếu bạn quên một tập

tin mới mà bạn tạo ra, cam kết các thư mục sẽ tìm thấy nó. Cam kết một thư mục *không* có nghĩa là mỗi tập tin được đánh dấu là thay đổi, chỉ làm cho cuộc sống của bạn dễ dàng hơn bằng cách thực hiện công việc nhiều hơn cho bạn.



## Nhiều tập tin không phiên bản trong hộp thoại cam kết

Nếu bạn thấy hộp thoại cam kết cho thấy bạn quá nhiều các tập tin không phiên bản (ví dụ như trình biên dịch tạo ra hoặc sao lưu dự phòng biên tập), có một số cách để xử lý việc này. Bạn có thể:

- thêm các tập tin (hoặc một phần mở rộng ký tự đại diện) vào danh sách các tập tin để loại trừ trên các trang thiết lập. Điều này sẽ ảnh hưởng đến tất cả các bản sao làm việc mà bạn có.
- thêm các tập tin vào danh sách `svn:ignore` sử dụng TortoiseSVN → Thêm vào danh sách bỏ qua Điều này sẽ chỉ ảnh hưởng đến thư mục mà bạn thiết lập thuộc tính `svn:ignore`. Sử dụng hộp thoại Thuộc Tính SVN, bạn có thể thay đổi thuộc tính `svn:ignore` cho một thư mục.
- 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.

Đọc [Phần 4.13](#), “[Bỏ qua các Tập Tin và Thư mục](#)” để biết thêm thông tin.

Nhấp đúp chuột trên bất kỳ tập tin được sửa đổi trong hộp thoại cam kết sẽ khởi động công cụ tìm khác biệt bên ngoài để hiển thị các thay đổi của bạn. Trình đơn ngữ cảnh sẽ cung cấp cho bạn nhiều lựa chọn hơn, như thể hiện trong ảnh chụp màn hình. Bạn cũng có thể kéo tập tin từ ở đây vào một ứng dụng khác chẳng hạn như một trình soạn thảo văn bản hoặc một IDE.

Bạn có thể chọn hoặc bỏ chọn các mục bằng cách nhấp vào hộp kiểm bên trái của mục đó. Trong thư mục, bạn có thể sử dụng **Shift**-chọn để làm cho hành động đệ quy.

Các cột hiển thị ở khung bên dưới có thể tùy chỉnh. Nếu bạn click chuột phải trên bất kỳ tiêu đề cột, bạn sẽ thấy một trình đơn ngữ cảnh cho phép bạn chọn các cột được hiển thị. Bạn cũng có thể thay đổi chiều rộng cột bằng cách sử dụng xử lý kéo xuất hiện khi bạn di chuyển chuột lên một ranh giới cột. Các tùy chỉnh được bảo tồn, vì vậy bạn sẽ thấy cùng các tiêu đề trong thời gian tiếp theo.

Theo mặc định khi bạn cam kết thay đổi, bất kỳ ổ khóa mà bạn giữ trên các tập tin được phát hành tự động sau khi cam kết thành công. Nếu bạn muốn giữ những ổ khóa, đảm bảo hộp kiểm **Giữ khóa** được đánh dấu. Trạng thái mặc định của hộp kiểm này được lấy từ tùy chọn `no_unlock` trong file cấu hình Subversion. Đọc [Phần 4.30.1](#), “[Các Thiết Lập Tổng Qu](#)” để có thông tin về việc làm thế nào để chỉnh sửa các file cấu hình Subversion.



## Kéo và thả

Bạn có thể kéo tập tin vào hộp thoại cam kết từ các nơi khác, miễn là bản sao làm việc được kiểm tra ra từ cùng kho lưu trữ. Ví dụ, bạn có thể có một bản sao làm việc rất lớn với một số nhà thám hiểm cửa sổ đang mở nhìn vào thư mục xa của hệ thống phân cấp. Nếu bạn muốn tránh cam kết từ các thư mục cấp cao nhất (với một thư thập dữ liệu thư mục dài để kiểm tra xem có thay đổi), bạn có thể mở hộp thoại cam kết cho một thư mục và kéo vào các mục từ các cửa sổ khác bao gồm trong các cam kết nguyên tử.

Bạn có thể kéo các tập tin không phiên bản cư trú trong một bản sao làm việc vào hộp thoại cam kết, và chúng sẽ được SVN tự động thêm vào.

Kéo tập tin từ danh sách ở dưới cùng của hộp thoại cam kết vào hộp thông điệp tường trình sẽ chèn các đường dẫn dưới dạng văn bản vào đó hộp chỉnh sửa. Điều này rất hữu ích nếu bạn muốn viết tin nhắn thông điệp tường trình cam kết bao gồm những đường dẫn bị ảnh hưởng bởi các cam kết.





### Sửa chữa đổi tên bên ngoài

Đôi khi các tập tin được đổi tên từ bên ngoài của Subversion, và chúng hiển thị trong danh sách tập tin như một tập tin bị mất và một tập tin không phiên bản. Để tránh mất lịch sử bạn cần phải thông báo cho Subversion về kết nối. Đơn giản chỉ cần chọn cả hai tên (bị mất) và tên mới (không phiên bản) và sử dụng Trình Đơn Ngữ Cảnh → Sửa chữa Di chuyển để ghép hai tập tin như đổi tên.



### Sửa chữa bản sao bên ngoài

Nếu bạn đã thực hiện một bản sao của một tập tin nhưng quên sử dụng lệnh Subversion để làm như vậy, bạn có thể sửa chữa bản sao để các tập tin mới không bị mất lịch sử của nó. Đơn giản chỉ cần chọn tên cũ (bình thường hoặc sửa đổi) và tên mới (không phiên bản) và sử dụng Trình Đơn Ngữ Cảnh → Sửa chữa Sao chép ghép hai tập tin như một bản.

## 4.4.2. Danh sách Thay đổi

Hộp thoại cam kết hỗ trợ tính năng danh sách thay đổi Subversion để giúp đỡ với nhóm các tập tin liên quan với nhau. Tìm hiểu về tính năng này trong [Phần 4.8, “Danh sách Thay đổi”](#).

## 4.4.3. Chỉ cam kết các phần của tập tin

Thỉnh thoảng bạn chỉ muốn cam kết vài phần của các thay đổi bạn thực hiện tới một tập tin. Tình huống như vậy thường xảy ra khi bạn làm việc trên vài thứ nhưng rồi một việc sửa chữa khẩn cấp cần được cam kết, và việc sửa chữa đó tình cờ xảy ra trong cùng tập tin bạn đang làm việc.

nhấn chuột phải trên tập tin và dùng Trình Đơn Ngữ Cảnh → Khôi phục sau khi cam kết. Việc này sẽ tạo ra một bản sao của tập tin. Rồi thì bạn có thể biên tập tập tin, ví dụ như với TortoiseMerge và hoàn tác các thay đổi bạn không muốn cam kết. Sau khi lưu các thay đổi đó bạn có thể cam kết tập tin.

Sau khi việc cam kết xong, bản sao của tập tin được khôi phục một cách tự động, và bạn nhận lại một tập tin với tất cả các chỉnh sửa không được cam kết.

## 4.4.4. Loại trừ mục khỏi Danh mục Cam kết

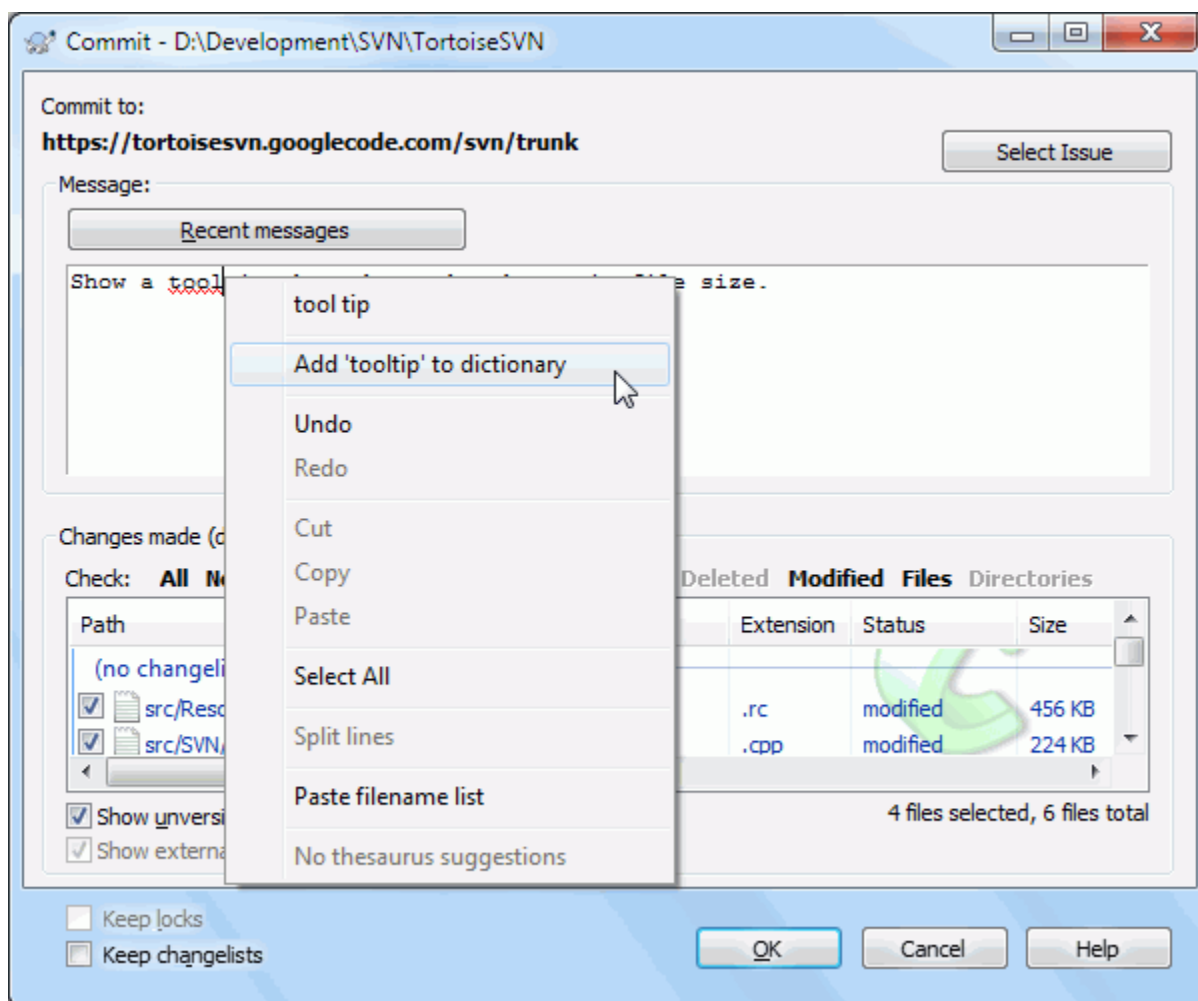
Đôi khi bạn có các tập tin được phiên bản mà thay đổi thường xuyên nhưng mà bạn thực sự không muốn cam kết. Đôi khi điều này cho thấy một lỗ hổng trong quá trình xây dựng của bạn - tại sao các tập tin được phiên bản? bạn có nên sử dụng tập tin mẫu? Nhưng đôi khi nó là không thể tránh khỏi. Một lý do cổ điển là IDE của bạn thay đổi dấu thời gian trong các tập tin dự án mỗi khi bạn xây dựng. Các hồ sơ dự án phải được phiên bản vì nó bao gồm tất cả các thiết lập xây dựng, nhưng nó không cần phải được cam kết chỉ vì các dấu thời gian đã thay đổi.

Để giúp đỡ trong các trường hợp khó xử như thế này, chúng tôi đã dành một danh sách thay đổi gọi là bỏ qua cam kết. Bất kỳ tập tin thêm vào danh sách thay đổi này sẽ tự động được đánh dấu trong hộp thoại cam kết. Bạn vẫn có thể cam kết thay đổi, nhưng bạn phải chọn nó trong hộp thoại cam kết.

## 4.4.5. Thông Điệp Tường Trình Cam Kết

Hãy chắc chắn để nhập vào một thông điệp tường trình trong đó mô tả những thay đổi bạn đang cam kết. Điều này sẽ giúp bạn xem những gì đã xảy ra và khi nào, khi bạn duyệt qua các thông điệp tường trình dự án vào một ngày sau đó. Các thông điệp có thể dài hoặc ngắn gọn như bạn thích, nhiều dự án có hướng dẫn cho những gì nên được bao gồm, ngôn ngữ sử dụng, và đôi khi thậm chí là một định dạng nghiêm ngặt.

Bạn có thể áp dụng định dạng đơn giản cho thông điệp tường trình của bạn bằng cách sử dụng một quy ước tương tự như được sử dụng trong email. Để áp dụng phong cách vào văn bản, sử dụng \*Văn bản\* cho chữ đậm, `_text_` gạch chân, và `^Văn bản^` cho chữ in nghiêng.



**Hình 4.9. Bộ kiểm tra chính tả của Hộp Thoại Cam Kết**

TortoiseSVN bao gồm một bộ kiểm tra chính tả để giúp bạn có thông điệp tường trình của bạn đúng. Điều này sẽ làm nổi bật bất kỳ từ sai chính tả. Sử dụng trình đơn ngữ cảnh để truy cập vào chỉnh sửa được đề nghị. Tất nhiên, nó không biết *tất cả* thuật ngữ kỹ thuật mà bạn biết, vì vậy từ được đánh vắn một cách chính xác đôi khi sẽ xuất hiện như là lỗi. Nhưng đừng lo lắng. Bạn chỉ có thể thêm vào từ điển cá nhân của bạn bằng cách sử dụng trình đơn ngữ cảnh.

Cửa sổ thông điệp tường trình cũng bao gồm một tên tập tin và cơ sở chức năng tự động hoàn thành. Điều này sử dụng các biểu thức thông thường để trích xuất các tên lớp và chức năng từ các tập tin (văn bản) bạn đang cam kết, cũng như các tên tập tin. Nếu một từ mà bạn đang gõ phù hợp với bất cứ điều gì trong danh sách (sau khi bạn đã gõ ít nhất 3 ký tự, hoặc nhấn **Ctrl+Space**), Một danh sách thả xuống xuất hiện cho phép bạn chọn tên đầy đủ. Các biểu thức thông thường được cung cấp với TortoiseSVN được tổ chức trong thư mục `bin` của lúc cài đặt TortoiseSVN. Bạn cũng có thể xác định biểu thức thông thường của riêng bạn và lưu trữ chúng trong `%APPDATA%\TortoiseSVN\autolist.txt`. Tất nhiên autolist của bạn sẽ không được ghi đè khi bạn cập nhật cài đặt TortoiseSVN của bạn. Nếu bạn không quen với các biểu thức thông thường, hãy nhìn vào việc giới thiệu tại [http://en.wikipedia.org/wiki/Regular\\_expression](http://en.wikipedia.org/wiki/Regular_expression), và tài liệu trực tuyến và hướng dẫn tại <http://www.regular-expressions.info/>.

Để có regex đúng có thể được học búa, do đó, để giúp bạn sắp xếp một biểu thức phù hợp có một hộp thoại kiểm tra cho phép bạn nhập một biểu thức và sau đó gõ vào tên tập tin để kiểm tra trên đó. Bắt đầu từ dấu nhắc lệnh bằng cách sử dụng lệnh `TortoiseProc.exe/command:autotexttest`.

Bạn có thể sử dụng lại thông điệp tường trình được nhập vào trước đó. Chỉ cần nhấp vào **Các thông điệp gần đây** để xem một danh sách các tin nhắn cuối cùng mà bạn nhập vào cho bản sao làm việc này. Số lượng thông điệp được lưu trữ có thể được tùy chỉnh trong hộp thoại cài đặt TortoiseSVN.

Bạn có thể dọn dẹp tất cả thông điệp tường trình cam kết được lưu trữ từ trang **Dữ liệu được lưu** của các thiết lập TortoiseSVN, hoặc bạn có thể xóa từng thông điệp từ bên trong hộp thoại **Các thông điệp gần đây** bằng cách sử dụng phím **Xóa**.

Nếu bạn muốn bao gồm các đường dẫn được kiểm tra trong thông điệp tường trình của bạn, bạn có thể sử dụng lệnh **Trình đơn ngữ cảnh** → **Dán danh sách tên tập tin** trong hộp biên tập.

Một cách khác để chèn đường dẫn vào thông điệp tường trình là chỉ cần kéo các tập tin từ danh sách tập tin vào hộp biên tập.



### Các Thuộc Tính Thư Mục Đặc Biệt

Có một số thuộc tính thư mục đặc biệt có thể được sử dụng để giúp cung cấp nhiều kiểm soát hơn trên các định dạng của thông điệp tường trình cam kết và ngôn ngữ được sử dụng bởi các mô đun kiểm tra chính tả. Đọc **Phần 4.17, “Thiết lập Dự án”** để biết thêm thông tin.

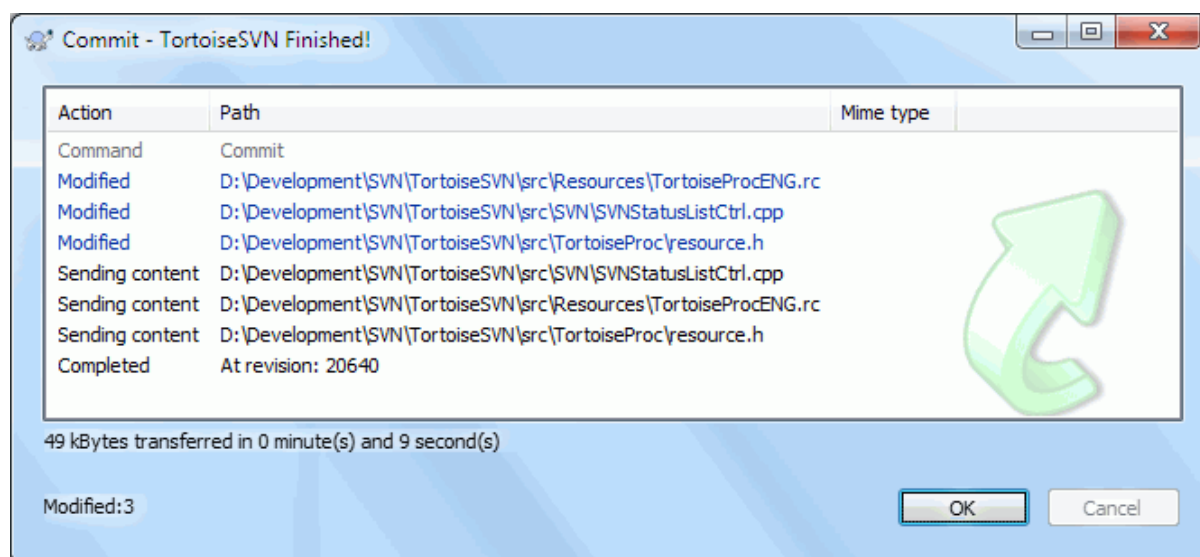


### Tích hợp với Công cụ Theo Dõi Lỗi

Nếu bạn đã kích hoạt hệ thống theo dõi lỗi, bạn có thể thiết lập một hoặc nhiều vấn đề trong hộp văn bản **Bug-ID/Vấn đề-Nr.** Nhiều vấn đề cần được phân tách bằng dấu phẩy. Ngoài ra, nếu bạn đang sử dụng hỗ trợ theo dõi lỗi dựa trên regex, chỉ cần thêm tài liệu tham khảo vấn đề của bạn như là một phần của thông điệp tường trình. Tìm hiểu thêm trong **Phần 4.28, “Tích hợp với Hệ thống Theo dõi Lỗi / Trình theo dõi Vấn đề”**.

## 4.4.6. Tiến độ Cam kết

Sau khi nhấn **OK**, một hộp thoại sẽ xuất hiện hiển thị tiến độ cam kết.



**Hình 4.10. Hộp thoại tiến độ thể hiện một cam kết đang tiến triển**

Hộp thoại tiến triển sử dụng màu sắc mã hóa để làm nổi bật hành động cam kết khác nhau

Màu xanh dương  
Cam kết một sửa đổi

Màu tím  
Cam kết một bổ sung mới

Màu đỏ sẫm

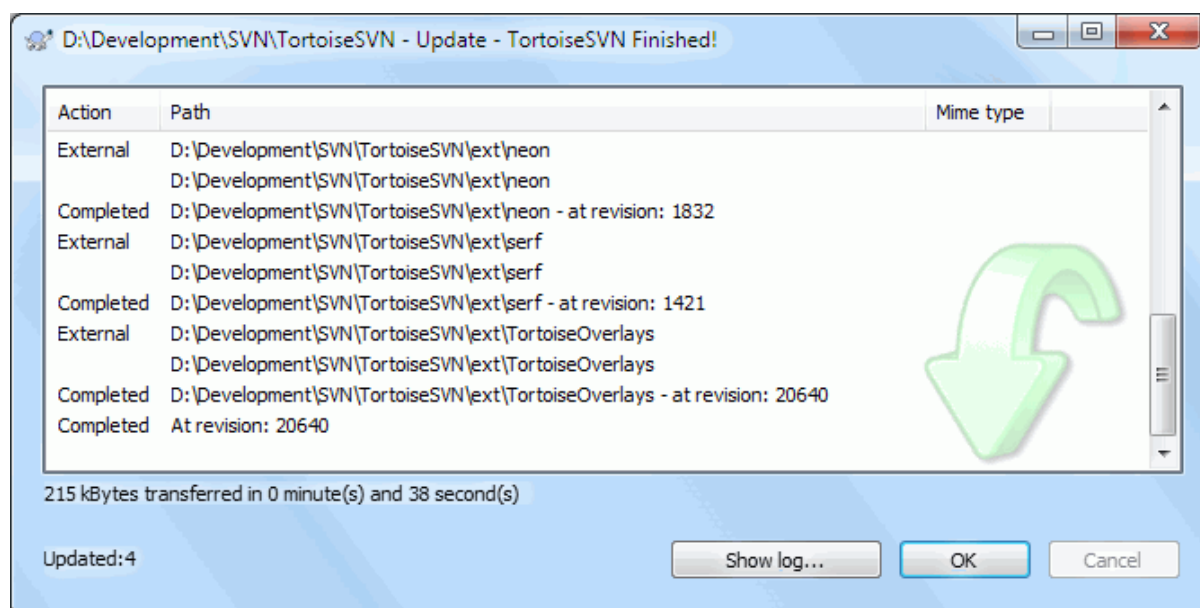
Cam kết một xoá bỏ hoặc thay thế.

Đen

Tất cả các mục khác.

Đây là sơ đồ màu mặc định, nhưng bạn có thể tùy chỉnh những màu sắc bằng cách sử dụng hộp thoại thiết lập. Đọc [Phần 4.30.1.5, “Thiết Lập Màu TortoiseSVN”](#) để biết thêm thông tin.

## 4.5. Cập nhật bản sao làm việc của bạn với thay đổi từ những người khác



**Hình 4.11. Hộp thoại tiến độ hiển thị cập nhật hoàn thành**

Định kỳ, bạn nên đảm bảo rằng những thay đổi được thực hiện bởi những người khác được kết hợp trong bản sao của bạn làm việc cục bộ. Quá trình lấy thay đổi từ máy chủ đến bản sao địa phương của bạn được gọi là *cập nhật*. Cập nhật có thể được thực hiện trên các tập tin đơn lẻ, một tập hợp các tập tin được lựa chọn, hoặc đệ quy trên toàn bộ hệ thống phân cấp thư mục. Để cập nhật, chọn các tập tin và / hoặc thư mục bạn muốn, click chuột phải và chọn **TortoiseSVN** → **Cập nhật** trong trình đơn ngữ cảnh thám hiểm. Một cửa sổ sẽ bật lên hiển thị tiến độ của bản cập nhật lúc nó chạy. Thay đổi được thực hiện bởi những người khác sẽ được sáp nhập vào tập tin của bạn, giữ bất kỳ thay đổi bạn có thể thực hiện trên các tập tin tương tự. Kho *không* bị ảnh hưởng bởi bản cập nhật.

Hộp thoại tiến triển sử dụng màu sắc mã hóa để làm nổi bật các hành động cập nhật khác

Màu tím

Thêm mục mới vào WC của bạn.

Màu đỏ sẫm

Mục dự phòng đã bị xóa từ WC của bạn, hoặc mục mất tích được thay thế trong WC của bạn.

Xanh lá cây

Thay đổi từ kho được sáp nhập thành công với những thay đổi địa phương của bạn.

Màu đỏ tươi

Thay đổi từ kho lưu trữ hợp nhất với những thay đổi của địa phương, kết quả là tạo ra các cuộc xung đột mà bạn cần phải giải quyết.

Đen

Mục không thay đổi trong WC của bạn được cập nhật với phiên bản mới hơn từ kho.

Đây là sơ đồ màu mặc định, nhưng bạn có thể tùy chỉnh những màu sắc bằng cách sử dụng hộp thoại thiết lập. Đọc [Phần 4.30.1.5](#), “[Thiết Lập Màu TortoiseSVN](#)” để biết thêm thông tin.

Nếu bạn nhận được bất kỳ *xung đột* trong một bản cập nhật (điều này có thể xảy ra nếu những người khác thay đổi dòng tương tự trong cùng một tập tin như bạn đã làm và những thay đổi đó không khớp) sau đó hộp thoại cho thấy những xung đột bằng màu đỏ. Bạn có thể nhấp đúp chuột trên những dòng này để bắt đầu công cụ sáp nhập bên ngoài để giải quyết xung đột.

Khi cập nhật hoàn tất, hộp thoại tiến độ cho thấy một bản tóm tắt về số lượng các mục được cập nhật, bổ sung, gỡ bỏ mâu thuẫn, vv bên dưới danh sách tập tin. Thông tin tóm tắt này có thể được sao chép vào clipboard bằng cách sử dụng **Ctrl+C** .

Các lệnh Cập Nhật tiêu chuẩn không có các tùy chọn và chỉ cập nhật bản sao làm việc của bạn đến sửa đổi CHÍNH của kho lưu trữ, đó là trường hợp sử dụng phổ biến nhất. Nếu bạn muốn kiểm soát nhiều hơn đối với quá trình cập nhật, thay vào đó bạn nên sử dụng TortoiseSVN → Cập Nhật đến Sửa Đổi ... . Điều này cho phép bạn cập nhật bản sao làm việc của bạn tới một phiên bản cụ thể, không chỉ là một bản gần đây nhất. Giả sử bản sao làm việc của bạn là tại sửa đổi 100, nhưng bạn muốn nó phản ánh tình trạng mà nó đã có trong bản chính sửa 50 - thì chỉ cần cập nhật đến sửa đổi 50.

Trong cùng một hộp thoại, bạn cũng có thể chọn *độ sâu* để cập nhật thư mục hiện hành. Các điều khoản được sử dụng được mô tả trong [Phần 4.3.1](#), “[Độ sâu Kiểm xu](#)”. Độ sâu mặc định là **Bản Sao Làm Việc** bảo tồn các thiết lập độ sâu hiện có . Bạn cũng có thể thiết lập độ sâu đích có nghĩa là bản cập nhật tiếp theo sẽ sử dụng độ sâu mới đó, tức là độ sâu mà sau đó được sử dụng như là độ sâu mặc định.

Để làm cho nó dễ dàng hơn để bao gồm hoặc loại trừ các mục cụ thể từ kiểm xuất nhấp chuột vào nút bấm **Chọn mục ...** . Điều này sẽ mở ra một hộp thoại mới, nơi bạn có thể kiểm tra tất cả các mục mà bạn muốn trong bản sao làm việc của bạn và bỏ chọn tất cả các mục bạn không muốn.

Bạn cũng có thể chọn để bỏ qua bất kỳ dự án bên ngoài trong bản cập nhật (có nghĩa là dự án được tham chiếu sử dụng `svn:externals` ).



## Cẩn thận

Nếu bạn cập nhật một tập tin hoặc thư mục vào một phiên bản cụ thể, bạn không nên thay đổi các tập tin. Bạn sẽ nhận được thông báo lỗi “*lỗi thời*” khi bạn cố gắng để thực hiện chúng! Nếu bạn muốn lùi lại thay đổi trong một tập tin và khởi đầu lại từ một phiên bản trước đó, bạn có thể cuộn lại một phiên bản trước đó từ hộp thoại đăng nhập sửa đổi. Hãy xem [Phần B.4](#), “[Phục hồi \(Hoàn tác\) các sửa đổi trong kho](#)” được hướng dẫn thêm, và phương pháp thay thế.

Cập nhật đến bản sửa đổi đôi khi có thể hữu ích để xem dự án của bạn trông giống những gì tại một số điểm trước đó trong lịch sử của nó. Nhưng nói chung, cập nhật các tập tin cá nhân đến một phiên bản trước đó không phải là một ý tưởng tốt vì nó để lại bản sao của bạn làm việc trong một tình trạng không phù hợp. Nếu các tập tin đang được cập nhật đã thay đổi tên, thậm chí bạn có thể tìm thấy tập tin đó chỉ biến mất từ bản sao làm việc của bạn bởi vì không có tập tin tên này đã tồn tại trong phiên bản trước đó. Bạn cũng nên lưu ý rằng các mục sẽ hiển thị một lớp phủ màu xanh lá cây bình thường, vì vậy không thể phân biệt nó từ các tập tin đã được cập nhật.

Nếu bạn chỉ đơn giản là muốn có một bản sao địa phương của một phiên bản cũ của một tập tin nó thì tốt hơn là sử dụng lệnh **Trình đơn ngữ cảnh** → **Lưu sửa đổi vào ...** từ hộp thoại tương trình cho tập tin đó.



## Nhiều Tập tin / thư mục

Nếu bạn chọn nhiều tập tin và thư mục trong trình thám hiểm và sau đó chọn **Cập nhật** , tất cả các tập tin / thư mục được cập nhật từng cái một. TortoiseSVN đảm bảo rằng tất cả các tập tin / thư mục từ kho lưu trữ được cập nhật đến bản sửa đổi chính xác! Ngay cả khi giữa những cập nhật đó một cam kết khác xảy ra.

## 4.6. Giải quyết mâu thuẫn

Lâu lâu một lần, bạn sẽ nhận được một *xung đột* khi bạn cập nhật / sáp nhập các tập tin của bạn từ kho hoặc khi bạn chuyển bản sao của bạn làm việc đến một URL khác nhau. Có hai loại xung đột:

xung đột tập tin

Sự xung đột giữa các tập tin xảy ra nếu có hai (hoặc nhiều hơn) nhà phát triển đã thay đổi một vài dòng của một tập tin.

các xung đột cây

Sự xung đột giữa các cây xảy ra khi một nhà phát triển di chuyển / đổi tên / xóa một tập tin hoặc thư mục, mà một nhà phát triển khác hoặc cũng đã di chuyển / đổi tên / xóa hoặc chỉ sửa đổi.

#### 4.6.1. Các xung đột tập tin

Sự xung đột giữa các tập tin xảy ra khi hai hoặc nhiều nhà phát triển đã thay đổi một vài dòng cùng của một tập tin. Do Subversion không biết gì về dự án của bạn, nó để việc giải quyết những cuộc xung đột cho các nhà phát triển. Khu vực xung đột trong một tập tin văn bản được đánh dấu như thể này:

```
<<<<<<< filename
    các thay đổi của bạn
=====
    mã sáp nhập từ kho
>>>>>>> revision
```

Ngoài ra, đối với mỗi tập tin mâu thuẫn Subversion đặt thêm ba tập tin trong thư mục của bạn:

filename.ext.mine

Đây là của bạn vì nó tồn tại trong bản sao làm việc của bạn trước khi bạn cập nhật bản sao của bạn làm việc - có nghĩa là, không có đánh dấu xung đột. tập tin này có những thay đổi mới nhất của bạn trong đó và không có gì khác.

filename.ext.OLDREV

Đây là tập tin sửa đổi CO BẢN trước khi bạn cập nhật bản sao của bạn làm việc. Nghĩa là, các tập tin mà bạn kiểm tra ra trước khi bạn đã thực hiện chỉnh sửa mới nhất của bạn.

filename.ext.rNEWREV

Đây là tập tin mà máy khách Subversion của bạn chỉ nhận được từ máy chủ khi bạn cập nhật bản sao làm việc của bạn. Tập tin này tương ứng với bản sửa đổi CHÍNH của kho.

Bạn có thể khởi động một công cụ hợp nhất / biên tập xung đột bên ngoài với TortoiseSVN → Chỉnh Sửa Xung Đột hoặc bạn có thể sử dụng bất kỳ trình soạn thảo văn bản để giải quyết cuộc xung đột bằng tay. Bạn nên quyết định các mã nên trông như thế nào, làm những thay đổi cần thiết và các tập tin. Sử dụng một công cụ hợp nhất như TortoiseMerge hoặc một trong những công cụ phổ biến khác là lựa chọn dễ dàng hơn bởi chúng nói chung trình bày các tập tin liên quan trong một cửa sổ xem 3-bảng và bạn không phải lo lắng về những dấu hiệu xung đột. Nếu bạn sử dụng một trình soạn thảo văn bản thì bạn nên tìm kiếm các dòng bắt đầu với chuỗi <<<<<<< .

Sau đó thực hiện các lệnh TortoiseSVN → Giải quyết và cam kết sửa đổi của bạn vào kho lưu trữ. Xin lưu ý rằng lệnh Giải Quyết không thực sự giải quyết cuộc xung đột. Chỉ cần loại bỏ các tập tin filename.ext.mine và filename.ext.r \* , sẽ cho phép bạn cam kết thay đổi của bạn.

Nếu bạn có xung đột với các tập tin nhị phân, Subversion không cố gắng để hợp nhất các tập tin. Các tập tin địa phương vẫn không thay đổi (chính xác như lần cuối bạn thay đổi nó) và bạn có các tập tin filename.ext.r \* . Nếu bạn muốn loại bỏ các thay đổi của bạn và giữ các phiên bản trong kho lưu trữ, chỉ cần sử dụng lệnh Hoàn Nguyên. Nếu bạn muốn giữ phiên bản của bạn và ghi đè lên phiên bản kho lưu trữ, sử dụng lệnh Giải Quyết, sau đó cam kết phiên bản của bạn.

Bạn có thể sử dụng lệnh Giải Quyết cho nhiều tập tin nếu bạn kích chuột phải vào thư mục mẹ và chọn TortoiseSVN → Giải Quyết ... Điều này sẽ mang đến một hộp thoại liệt kê tất cả các tập tin mâu thuẫn trong thư mục đó, và bạn có thể chọn những cái để đánh dấu là đã giải quyết.

## 4.6.2. Xung đột về thuộc tính

Sự xung đột giữa các thuộc tính xảy ra khi hai hay nhiều nhà phát triển đã thay đổi cùng một thuộc tính. Như với nội dung tập tin, giải quyết cuộc xung đột chỉ có thể được thực hiện bởi các nhà phát triển.

Nếu một trong những thay đổi phải ghi đè lên nhau thì chọn tùy chọn để **Giải quyết bằng cách sử dụng tính chất địa phương** hoặc **Giải quyết bằng cách sử dụng tính chất từ xa**. Nếu thay đổi phải được sáp nhập sau đó chọn **Tự chỉnh sửa tính chất**, sắp xếp những gì giá trị tính chất nên là và đánh dấu như đã giải quyết.

## 4.6.3. Mâu Thuẫn Cây

Sự xung đột giữa các cây xảy ra khi một nhà phát triển di chuyển / đổi tên / xóa một tập tin hoặc thư mục, mà một nhà phát triển khác hoặc cũng đã di chuyển / đổi tên / xóa hoặc chỉ cần sửa đổi. Có rất nhiều tình huống khác nhau có thể dẫn đến một cuộc xung đột cây, và tất cả chúng yêu cầu các bước khác nhau để giải quyết cuộc xung đột.

Khi một tập tin bị xóa tại địa phương trong Subversion, tập tin đó cũng bị xóa từ hệ thống tập tin địa phương, do đó, ngay cả nếu nó là một phần của một cuộc xung đột cây nó không thể hiển thị một lớp phủ mâu thuẫn và bạn không thể nhấp chuột phải vào nó để giải quyết cuộc xung đột. Sử dụng hộp thoại **Kiểm tra xem có sửa đổi thay vì truy cập vào tùy chọn Chỉnh sửa xung đột**.

TortoiseSVN có thể giúp tìm đúng nơi để sáp nhập các thay đổi, nhưng các việc cần thiết có thể cần thêm để sắp xếp những cuộc xung đột. Hãy nhớ rằng sau khi cập nhật một bản làm việc CỐ SỞ thì sẽ luôn luôn có các sửa đổi ở từng hạng mục như trong kho tại thời điểm cập nhật. Nếu bạn trở lại một sự thay đổi sau khi cập nhật thì nó đi trở lại tình trạng kho, không giống như nó trước đó khi bạn bắt đầu tạo ra những thay đổi của địa phương của bạn.

### 4.6.3.1. Xóa tại địa phương, biên tập sắp tới khi cập nhật

1. Nhà phát triển A thay đổi `Foo.c` và cam kết nó vào kho lưu trữ.
2. Nhà phát triển B đã đồng thời di chuyển `Foo.c` thành `Bar.c` trong bản sao làm việc của mình, hoặc chỉ đơn giản là xóa `Foo.c` hoặc thư mục cha mẹ của nó.

Bản cập nhật bản sao kết quả của nhà phát triển B dẫn đến một cuộc xung đột cây:

- `Foo.c` đã được xóa từ bản sao làm việc, nhưng được đánh dấu bằng một cuộc xung đột cây.
- Nếu kết quả cuộc xung đột từ việc đổi tên một chữ không phải là một việc xóa thì `Bar.c` được đánh dấu là thêm vào, nhưng không chứa các sửa đổi của nhà phát triển A.

Nhà phát triển B đã chọn giữ các thay đổi của Nhà phát triển A. Trong trường hợp đổi tên của một tập tin, ông có thể sáp nhập các thay đổi `Foo.c` vào các tập tin đổi tên `Bar.c`. Đối với việc xóa bỏ tập tin hoặc thư mục đơn giản, ông có thể chọn giữ cho các mục với thay đổi của nhà phát triển A và loại bỏ việc xóa. Hoặc, bằng cách đánh dấu cuộc xung đột được giải quyết mà không làm bất cứ điều gì, ông ta loại bỏ thay đổi của nhà phát triển A một cách hiệu quả.

Hộp thoại soạn thảo xung đột đề xuất sáp nhập các thay đổi nếu nó có thể tìm thấy các tập tin nguyên thủy của tập tin đổi tên `Bar.c`. Tùy thuộc vào nơi cập nhật đã được gọi, nó có thể không tìm được tập tin nguồn.

### 4.6.3.2. Biên tập tại địa phương, việc xóa sắp tới trên cập nhật

1. Nhà phát triển A di chuyển `Foo.c` thành `Bar.c` và cam kết nó vào kho lưu trữ.
2. Nhà phát triển B thay đổi `Foo.c` trong bản sao làm việc của mình.

Hoặc trong trường hợp của một di chuyển thư mục ...

1. Nhà phát triển A di chuyển thư mục mẹ `FooFolder` sang `BarFolder` và cam kết nó vào kho lưu trữ.
2. Nhà phát triển B thay đổi `Foo.c` trong bản sao làm việc của mình.

Cập nhật các bản sao kết quả làm việc của nhà phát triển B dẫn đến một cuộc xung đột cây. Đối với một xung đột tập tin đơn giản:

- `Bar.c` được thêm vào bản sao làm việc như một tập tin bình thường.
- `Foo.c` được đánh dấu là thêm vào (với lịch sử) và có một cuộc xung đột cây.

Đối với một cuộc xung đột thư mục:

- `BarFolder` được thêm vào bản sao làm việc như một thư mục bình thường.
- `FooFolder` được đánh dấu là thêm vào (với lịch sử) và có một cuộc xung đột cây.

`Foo.c` được đánh dấu là đã sửa đổi.

Nhà phát triển B bây giờ phải quyết định xem có nên đi với việc tổ chức lại nhà phát triển A và hợp nhất các thay đổi của bà vào tập tin tương ứng trong cơ cấu mới, hoặc chỉ đơn giản là trở lại các thay đổi của A và giữ các tập tin địa phương.

Để nhập các thay đổi của mình với các cải tổ, nhà phát triển B trước tiên phải tìm ra tên tập tin gì mà tập tin mâu thuẫn `Foo.c` đã được đổi tên / di chuyển trong kho. Điều này có thể được thực hiện bằng cách sử dụng hộp thoại tương trình. Những thay đổi sau đó phải được sáp nhập bằng tay bởi hiện nay vẫn không có cách nào để tự động hoặc thậm chí đơn giản hóa quá trình này. Một khi các thay đổi đã được chuyển an toàn, các đường dẫn xung đột là không cần thiết và có thể bị xóa. Trong trường hợp này sử dụng nút **Loại bỏ** trong hộp thoại biên tập xung đột để làm sạch và đánh dấu cuộc xung đột là đã được giải quyết.

Nếu nhà phát triển B quyết định rằng thay đổi của A sai thì cô phải chọn nút **Giữ** trong hộp thoại biên tập xung đột. Điều này đánh dấu các tập tin / thư mục như đã giải quyết mâu thuẫn, nhưng thay đổi của nhà phát triển A cần phải được loại bỏ bằng tay. Một lần nữa, hộp thoại tương trình sẽ giúp theo dõi những gì đã được di chuyển.

#### 4.6.3.3. Việc xóa địa phương, việc xóa sắp đến khi cập nhật

1. Nhà phát triển A di chuyển `Foo.c` thành `Bar.c` và cam kết nó vào kho lưu trữ.
2. Nhà phát triển B di chuyển `Foo.c` sang `Bix.c`.

Bản cập nhật bản sao kết quả của nhà phát triển B dẫn đến một cuộc xung đột cây:

- `Bix.c` được đánh dấu là thêm vào với lịch sử.
- `Bar.c` được thêm vào bản sao làm việc với trạng thái "bình thường".
- `Foo.c` được đánh dấu là bị xóa và có một cuộc xung đột cây.

Để giải quyết cuộc xung đột này, nhà phát triển B phải tìm ra tên tập tin gì mà tập tin mâu thuẫn `Foo.c` đã được đổi tên / di chuyển trong kho. Điều này có thể được thực hiện bằng cách sử dụng hộp thoại tương trình.

Sau đó, nhà phát triển B phải quyết định tên tập tin mới của `Foo.c` để giữ - tập tin được thực hiện bởi nhà phát triển A hoặc đổi tên được thực hiện bởi ông ta.

Sau khi nhà phát triển B giải quyết bằng tay cuộc xung đột, cuộc xung đột cây đã được đánh dấu là giải quyết với các nút trong hộp thoại biên tập xung đột.

#### 4.6.3.4. Không có tại địa phương, chỉnh sửa sắp đến lúc hợp nhất

1. Nhà phát triển A làm việc trên thân cây thay đổi `Foo.c` và cam kết nó vào kho lưu trữ
2. Nhà phát triển B làm việc trên một chi nhánh di chuyển `Foo.c` sang `Bar.c` và cam kết nó vào kho lưu trữ

Một sự hợp nhất của những thay đổi trên thân cây của nhà phát triển A đến bản sao làm việc chi nhánh của nhà phát triển B dẫn đến một cuộc xung đột cây:



- `Bar.c` đã có trong bản sao làm việc với trạng thái "bình thường".
- `Foo.c` được đánh dấu là bị mất với một cuộc xung đột cây.

Để giải quyết cuộc xung đột này, nhà phát triển B phải đánh dấu các tập tin như đã được giải quyết trong hộp thoại biên tập xung đột, mà sẽ loại bỏ nó từ danh sách xung đột. Sau đó, cô đã quyết định liệu có sao chép các tập tin bị thiếu `Foo.c` từ kho lưu trữ các bản sao làm việc, cho dù kết hợp phát triển thay đổi của nhà phát triển A `Foo.c` vào tập tin được đổi tên `Bar.c` hoặc bỏ qua các thay đổi bằng cách đánh dấu cuộc xung đột được giải quyết và không làm gì khác.

Lưu ý rằng nếu bạn sao chép các tập tin bị mất từ kho và sau đó đánh dấu như giải quyết, sao chép của bạn sẽ được gỡ bỏ một lần nữa. Bạn phải giải quyết cuộc xung đột trước tiên.

#### 4.6.3.5. Biên tập tại địa phương, việc xóa đến lúc hợp nhất

1. Nhà phát triển A làm việc trên thân cây di chuyển `Foo.c` sang `Bar.c` và cam kết nó vào kho lưu trữ.
2. Nhà phát triển B làm việc trên một chi nhánh thay đổi `Foo.c` và cam kết nó vào kho lưu trữ.

Có một trường hợp tương đương với di chuyển thư mục, nhưng nó chưa được phát hiện trong Subversion 1.6 ...

1. Nhà phát triển A làm việc trên thân cây di chuyển thư mục mẹ `FooFolder` sang `BarFolder` và cam kết nó vào kho lưu trữ.
2. Nhà phát triển B làm việc trên một chi nhánh thay đổi `Foo.c` trên bản sao làm việc của mình.

Một sự hợp nhất của những thay đổi trên thân cây của nhà phát triển A đến bản sao làm việc chi nhánh của nhà phát triển B dẫn đến một cuộc xung đột cây:

- `Bar.c` được đánh dấu là đã được thêm vào.
- `Foo.c` được đánh dấu là đã sửa đổi với một cuộc xung đột cây.

Nhà phát triển B bây giờ phải quyết định xem có nên đi với việc tổ chức lại nhà phát triển A và hợp nhất các thay đổi của bà vào tập tin tương ứng trong cơ cấu mới, hoặc chỉ đơn giản là trở lại các thay đổi của A và giữ các tập tin địa phương.

Để sáp nhập các thay đổi của mình với các cải tổ, nhà phát triển B trước tiên phải tìm ra những tên tập tin gì mà tập tin mâu thuẫn `Foo.c` đã được đổi tên / di chuyển trong kho. Điều này có thể được thực hiện bằng cách sử dụng hộp thoại tường trình cho các nguồn hợp nhất. Các trình biên tập xung đột chỉ cho thấy các tường trình cho các bản sao làm việc vì nó không biết đường dẫn nào được sử dụng trong hợp nhất, do đó bạn sẽ phải tự mình tìm. Những thay đổi sau đó phải được sáp nhập bằng tay như hiện nay vẫn không có cách nào để tự động hoặc thậm chí đơn giản hóa quá trình này. Một khi các thay đổi đã được chuyển an toàn, các đường dẫn xung đột là không cần thiết và có thể bị xóa. Trong trường hợp này sử dụng nút **Hủy bỏ** trong hộp thoại biên tập xung đột để làm sạch và đánh dấu cuộc xung đột như đã được giải quyết.

Nếu nhà phát triển B quyết định rằng thay đổi của A sai thì cô phải chọn nút **Giữ** trong hộp thoại biên tập xung đột. Điều này đánh dấu các tập tin / thư mục như đã được giải quyết, nhưng thay đổi của nhà phát triển A cần phải được loại bỏ bằng tay. Một lần nữa, hộp thoại tường trình cho các nguồn hợp nhất giúp theo dõi những gì đã được di chuyển.

#### 4.6.3.6. Việc xóa địa phương, xóa đến khi hợp nhất

1. Nhà phát triển A làm việc trên thân cây di chuyển `Foo.c` sang `Bar.c` và cam kết nó vào kho lưu trữ.
2. Nhà phát triển B làm việc trên một chi nhánh di chuyển `Foo.c` sang `Bix.c` và cam kết nó vào kho lưu trữ.

Một sự hợp nhất của những thay đổi trên thân cây của nhà phát triển A đến bản sao làm việc chi nhánh của nhà phát triển B dẫn đến một cuộc xung đột cây:

- `Bix.c` được đánh dấu với trạng thái bình thường (chưa sửa đổi).
- `Bar.c` được đánh dấu là đã được thêm vào với lịch sử.
- `Foo.c` được đánh dấu là mất tích và có một cuộc xung đột cây.

Để giải quyết cuộc xung đột này, nhà phát triển B phải tìm ra những tên tập tin gì mà tập tin mâu thuẫn `Foo.c` đã được đổi tên / di chuyển trong kho. Điều này có thể được thực hiện bằng cách sử dụng hộp thoại tương trình cho các nguồn hợp nhất. Các trình biên tập xung đột chỉ cho thấy các trường trình cho các bản sao làm việc vì nó không biết đường dẫn nào được sử dụng trong hợp nhất, do đó bạn sẽ phải tự mình tìm.

Sau đó, nhà phát triển B phải quyết định tên tập tin mới của `Foo.c` để giữ - tập tin được thực hiện bởi nhà phát triển A hoặc đổi tên được thực hiện bởi ông ta.

Sau khi nhà phát triển B giải quyết bằng tay cuộc xung đột, cuộc xung đột cây đã được đánh dấu là giải quyết với các nút trong hộp thoại biên tập xung đột.

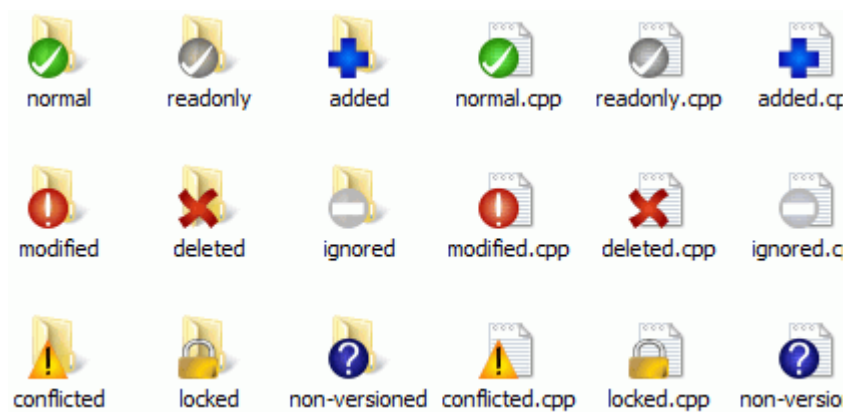
#### 4.6.3.7. Các xung đột cây khác

Có những trường hợp khác được dán nhãn là các cuộc xung đột cây chỉ đơn giản là vì cuộc xung đột liên quan đến một thư mục chứ không phải là một tập tin. Ví dụ nếu bạn thêm một thư mục có cùng tên cho cả hai thân và chi nhánh và sau đó cố gắng hợp nhất, bạn sẽ có được một cuộc xung đột cây. Nếu bạn muốn giữ lại các thư mục từ mục tiêu hợp nhất, chỉ cần đánh dấu cuộc xung đột là đã được giải quyết. Nếu bạn muốn sử dụng một trong các nguồn hợp nhất sau đó bạn cần phải dùng SVN xóa một trong những mục tiêu trước tiên và chạy hợp nhất một lần nữa. Nếu bạn cần bất cứ điều gì phức tạp hơn sau đó bạn phải giải quyết bằng tay.

### 4.7. Lấy Thông tin Trạng thái

Trong khi bạn đang làm việc trên bản sao làm việc của bạn, bạn thường cần phải biết những tập tin mà bạn đã thay đổi / bổ sung / gỡ bỏ hay đổi tên, hoặc thậm chí các tập tin đã thay đổi và cam kết bởi những người khác.

#### 4.7.1. Biểu tượng Lớp phủ



**Hình 4.12. Explorer hiển thị lớp phủ biểu tượng**

Bởi vì bây giờ bạn đã kiểm xuất một bản sao làm việc từ một kho lưu trữ Subversion bạn có thể xem các tập tin của bạn trong nhà thám hiểm cửa sổ với các biểu tượng đã được thay đổi. Đây là một trong những lý do tại sao TortoiseSVN là rất phổ biến. TortoiseSVN thêm vào một biểu tượng lớp phủ lên mỗi biểu tượng tập tin chồng lên biểu tượng tập tin ban đầu. Tùy thuộc vào tình trạng Subversion của tập tin mà biểu tượng lớp phủ sẽ là khác nhau.



Một bản sao làm việc mới kiểm xuất có một dấu kiểm màu xanh lá cây làm lớp phủ. Điều đó có nghĩa là tình trạng Subversion là *bình thường*.



Ngay sau khi bạn bắt đầu chỉnh sửa một tập tin, trạng thái đổi sang *sửa đổi* và lớp phủ biểu tượng sau đó thay đổi thành một dấu chấm than màu đỏ. Bằng cách đó bạn có thể dễ dàng nhìn thấy các tập tin đã được thay đổi kể từ lần cuối bạn cập nhật bản sao làm việc của bạn và cần phải được cam kết.



Nếu trong quá trình cập nhật một *xung đột* xảy ra thì biểu tượng thay đổi thành một dấu chấm than màu vàng.



Nếu bạn đã thiết lập thuộc tính `svn:needs-lock` vào một tập tin, Subversion làm cho tập tin đó chỉ đọc cho đến khi bạn nhận được một khóa trên tập tin đó. Các tập tin như vậy có lớp phủ để cho biết rằng bạn phải có được khóa trước khi bạn có thể chỉnh sửa tập tin đó.



Nếu bạn giữ một khóa trên một tập tin, và tình trạng Subversion là *bình thường*, lớp phủ biểu tượng này nhắc nhở bạn rằng bạn nên phát hành các khóa nếu bạn không sử dụng nó để cho phép người khác thực hiện các thay đổi của họ lên tập tin.



Biểu tượng này cho thấy rằng một số tập tin hoặc thư mục bên trong thư mục hiện hành đã được dự kiến sẽ được *xóa* từ kiểm soát phiên bản hoặc một tập tin dưới sự kiểm soát phiên bản đã mất trong một thư mục.



Dấu cộng cho bạn biết rằng một tập tin hoặc thư mục đã được dự kiến sẽ được *thêm* vào kiểm soát phiên bản.



Dấu hiệu thanh nói với bạn rằng một tập tin hoặc thư mục bị *bỏ qua* cho mục đích kiểm soát phiên bản. Lớp phủ này là tùy chọn.



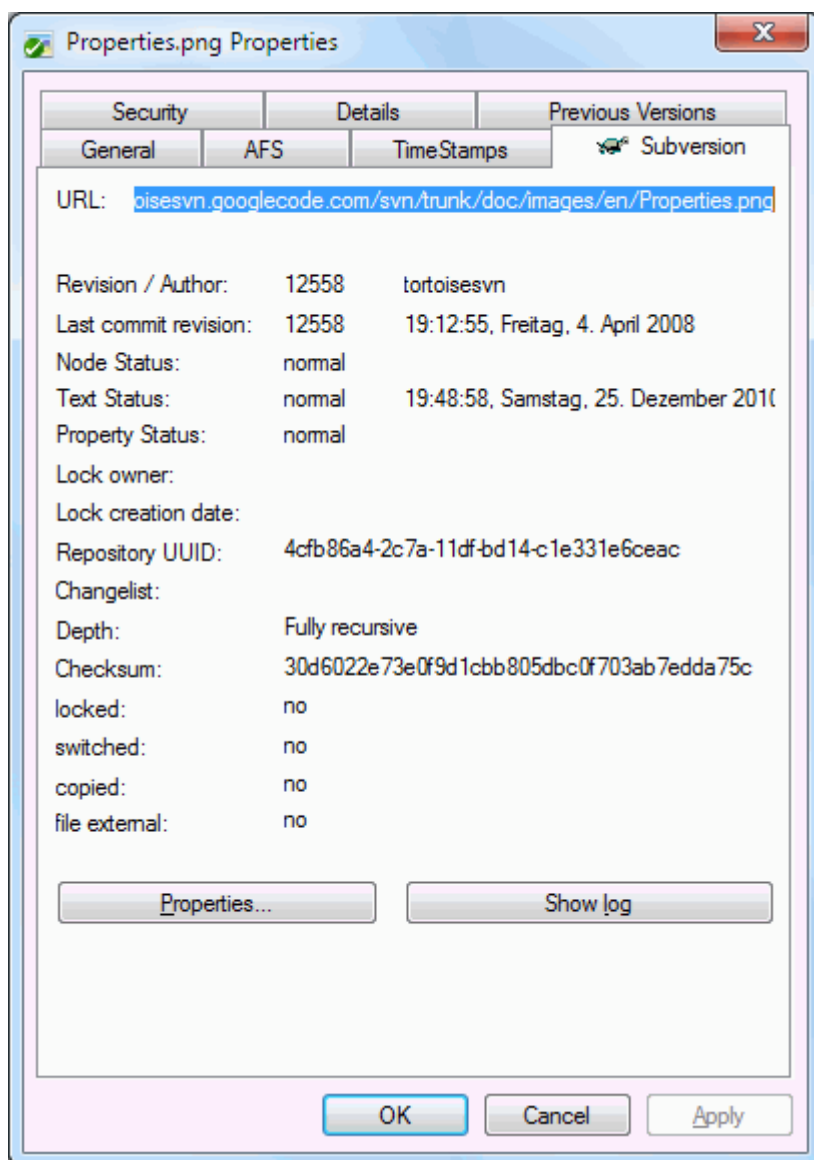
Biểu tượng này cho thấy các tập tin và thư mục không phải nằm dưới sự kiểm soát phiên bản, nhưng đã không được bỏ qua. Lớp phủ này là tùy chọn.

Trong thực tế, bạn có thể thấy rằng không phải tất cả các biểu tượng này được sử dụng trên hệ thống của bạn. Điều này là bởi vì số lượng của lớp phủ cho phép Windows là rất hạn chế và nếu bạn cũng sử dụng một phiên bản cũ của TortoiseCVS, sau đó có không đủ chỗ cho lớp phủ có sẵn. TortoiseSVN cố gắng là một “Công Dân Tốt (TM)” và hạn chế sử dụng lớp phủ để cung cấp cho các ứng dụng khác một cơ hội.

Bởi vì bây giờ có nhiều trình khách Tortoise xung quanh (TortoiseCVS, TortoiseHg, ...) giới hạn của biểu tượng sẽ trở thành một vấn đề thực sự. Để làm việc này, dự án TortoiseSVN giới thiệu một tập hợp biểu tượng chung chia sẻ, được nạp như một DLL, có thể được sử dụng bởi tất cả các trình khách Tortoise. Kiểm tra với nhà cung cấp dịch vụ khách hàng của bạn để xem nếu cái này đã được tích hợp chưa :-)

Đối với một mô tả của lớp phủ biểu tượng tương ứng với tình trạng Subversion và các chi tiết kỹ thuật khác, đọc [Phần F.1, “Biểu tượng Lớp phủ”](#).

## 4.7.2. Tình trạng Chi tiết



**Hình 4.13.** trang thuộc tính Explorer, tab Subversion

Đôi khi bạn muốn có thêm thông tin chi tiết về một tập tin / thư mục hơn là với lớp phủ biểu tượng. Bạn có thể nhận được tất cả các thông tin Subversion cung cấp trong hộp thoại thám hiểm thuộc tính. Chỉ cần chọn các tập tin hoặc thư mục và chọn Trình đơn Cửa sổ → thuộc tính trong trình đơn ngữ cảnh (lưu ý: đây là các thuộc tính trình đơn bình thường mà trình thám hiểm cung cấp, không phải là một trong các trình đơn phụ của TortoiseSVN). Trong hộp thoại thuộc tính TortoiseSVN thêm một trang thuộc tính mới cho các tập tin / thư mục do Subversion kiểm soát, nơi bạn có thể nhìn thấy tất cả các thông tin có liên quan về các tập tin / thư mục được lựa chọn.

### 4.7.3. Cột TortoiseSVN Trong Windows Explorer

Cùng một thông tin có sẵn từ các lớp phủ biểu tượng (và nhiều hơn nữa) có thể được hiển thị như cột bổ sung trong phần xem chi tiết của Windows Explorer.

Đơn giản chỉ cần kích chuột phải vào một trong các tiêu đề của một cột, chọn Nhiều hơn nữa ... từ trình đơn ngữ cảnh được hiển thị. Một hộp thoại sẽ xuất hiện nơi bạn có thể chỉ định các cột và thứ tự của chúng, được hiển thị trong “Xem chi tiết”. Di chuyển xuống cho đến khi các mục bắt đầu với SVN có thể xem được. Kiểm tra những cái bạn muốn hiển thị và đóng hộp thoại bằng cách nhấn OK. Các cột sẽ được nối thêm vào bên phải của những mục đang hiển thị. Bạn có thể sắp xếp lại chúng bằng cách kéo và thả, hoặc thay đổi kích thước, để chúng phù hợp với nhu cầu của bạn.



## Quan trọng

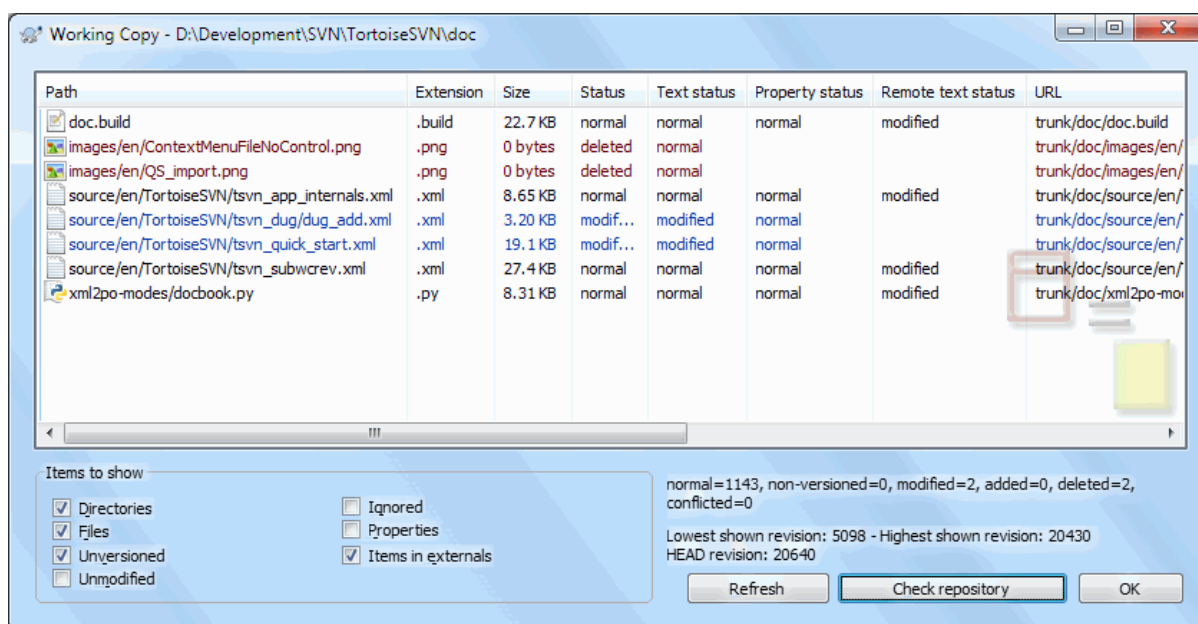
Các cột bổ sung trong Windows Explorer không có sẵn trên Vista, kể từ khi Microsoft quyết định không cho phép các cột như vậy cho *tất cả* tập tin nữa nhưng chỉ trừ các loại tập tin nhất định.



## Mẹo

Nếu bạn muốn bố trí hiện tại được hiển thị trong tất cả các bản sao làm việc của bạn, bạn có thể muốn nó làm giao diện mặc định.

### 4.7.4. Trạng thái địa phương và từ xa



**Hình 4.14. Kiểm tra xem có sửa đổi**

Thường rất hữu ích để biết những tập tin mà bạn đã thay đổi và cũng có các tập tin đã thay đổi và cam kết bởi những người khác. Đó là nơi mà các lệnh TortoiseSVN → Kiểm tra xem có sửa đổi ... trở nên có ích. Hộp thoại này sẽ cho bạn thấy tất cả các tập tin đã thay đổi trong bất kỳ cách nào trong bản sao làm việc của bạn, cũng như bất kỳ tập tin không được phiên bản mà bạn có thể có.

Nếu bạn bấm vào Kiểm tra Kho thì bạn cũng có thể tìm những thay đổi trong kho. Bằng cách đó bạn có thể kiểm tra trước khi cập nhật nếu xung đột có thể xảy ra. Bạn cũng có thể cập nhật các tập tin được lựa chọn từ kho lưu trữ mà không cần cập nhật toàn bộ thư mục. Theo mặc định, nút Kiểm tra Kho chỉ lấy tình trạng từ xa với độ sâu kiểm xuất của các bản sao làm việc. Nếu bạn muốn xem tất cả các tập tin và thư mục trong kho lưu trữ, ngay cả những thứ bạn đã không kiểm xu, thì bạn phải giữ phím **Shift** khi bạn bấm vào nút Kiểm tra Kho .

Hộp thoại sử dụng mã màu để làm nổi bật tình trạng.

Màu xanh dương

Các mục thay đổi tại địa phương.

Màu tím

Các mục được thêm. Các mục đã được thêm vào lịch sử có một dấu + ghi trong cột Tình trạng Văn bản, và một tooltip cho thấy nơi mà mục này đã được sao chép.

Màu đỏ sẫm

Các mục bị xóa hoặc mất.

### Xanh lá cây

Mục được sửa đổi tại địa phương và trong kho lưu trữ. Những thay đổi sẽ được sáp nhập lúc cập nhật. Những việc này *có thể* tạo các xung đột lúc cập nhật.

### Màu đỏ tươi

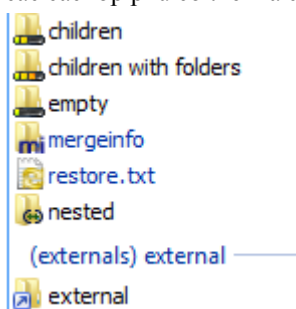
Mục sửa đổi tại địa phương và bị xóa trong kho, hoặc sửa đổi trong kho lưu trữ và xóa địa phương. Những việc này *sẽ* tạo các xung đột lúc cập nhật.

### Đen

Các mục không thay đổi và không được phiên bản.

Đây là sơ đồ màu mặc định, nhưng bạn có thể tùy chỉnh những màu sắc bằng cách sử dụng hộp thoại thiết lập. Đọc [Phần 4.30.1.5, “Thiết Lập Màu TortoiseSVN”](#) để biết thêm thông tin.

Biểu tượng lớp phủ cũng được dùng để thể hiện các trạng thái khác. Ảnh chụp màn hình dưới đây cho thấy tất cả các lớp phủ có thể mà được hiển thị khi cần.



Các lớp phủ được hiển thị cho các trạng thái sau đây:

- Độ sâu kiểm xuất để trống, có nghĩa là chỉ bản thân món đó.
- Độ sâu kiểm xuất các tập tin, có nghĩa chỉ là bản thân món đó và tất cả các tập tin con mà không có các thư mục con.
- Độ sâu kiểm xuất immediates, có nghĩa là chỉ bản thân mục đó và tất cả các tập tin và thư mục con, nhưng không có con của các thư mục con.
- Các món lồng ghép, nghĩa là các bản sao làm việc bên trong một bản sao làm việc.
- Các món từ bên ngoài, nghĩa là, tất cả các món được thêm và qua thuộc tính `svn:externals`.
- Các món được phục hồi sau một lệnh cam kết. Xem chi tiết ở [Phần 4.4.3, “Chi cam kết các phần của tập tin”](#).
- Các mục có sửa đổi thuộc tính, nhưng chỉ với thuộc tính `svn:mergeinfo`. Nếu có thuộc tính khác được sửa đổi, lớp phủ sẽ không được dùng.

Các mục đã được chuyển sang một đường dẫn kho lưu trữ khác cũng được chỉ định sử dụng một đánh dấu (s) . Bạn có thể đã chuyển một cái gì đó trong khi làm việc trên một chi nhánh và quên để chuyển đổi trở lại với thân cây. Đây là dấu hiệu cảnh báo của bạn! Trình đơn ngữ cảnh cho phép bạn chuyển đổi chúng trở lại đường dẫn bình thường.

Từ trình đơn ngữ cảnh của hộp thoại, bạn có thể hiển thị khác biệt trong những thay đổi. Kiểm tra các thay đổi cục bộ *bạn* thực hiện bằng cách sử dụng Trình đơn ngữ cảnh → So sánh với Cơ Sở . Kiểm tra những thay đổi của người khác trong các kho lưu trữ sử dụng Trình đơn ngữ cảnh → Hiển thị sự khác biệt dưới dạng khác biệt thống nhất .

Bạn cũng có thể trở lại những thay đổi trong các tập tin. Nếu bạn đã vô tình xóa một tập tin, nó sẽ hiển thị như là *Thiếu* và bạn có thể sử dụng *Hoàn nguyên* để phục hồi nó.

Các tập tin không phiên bản và bỏ qua có thể được gửi vào thùng rác từ đây bằng cách sử dụng Trình đơn ngữ cảnh → Xóa . Nếu bạn muốn xóa các tập tin vĩnh viễn (bỏ qua thùng rác) thì bấm phím **Shift** trong khi nhấp chuột vào Xóa .

Nếu bạn muốn kiểm tra chi tiết một tập tin, bạn có thể kéo nó từ đây vào một ứng dụng khác như một trình soạn thảo văn bản hay IDE, hoặc bạn có thể lưu một bản sao đơn giản bằng cách kéo nó vào một thư mục trong explorer.

Các cột có thể tùy chỉnh. Nếu bạn click chuột phải trên bất kỳ cột tiêu đề, bạn sẽ thấy một trình đơn ngữ cảnh cho phép bạn chọn các cột được hiển thị. Bạn cũng có thể thay đổi chiều rộng cột bằng cách sử dụng xử lý kéo xuất hiện khi bạn di chuyển chuột lên một ranh giới cột. Các tùy chỉnh được bảo tồn, vì vậy bạn sẽ thấy cùng các tiêu đề trong thời gian tiếp theo.

Nếu bạn đang làm việc trên một số nhiệm vụ không liên quan cùng một lúc, bạn cũng có thể nhóm các tập tin với nhau vào danh sách thay đổi. Đọc [Phần 4.4.2, “Danh sách Thay đổi”](#) để biết thêm thông tin.

Ở phía dưới của hộp thoại, bạn có thể nhìn thấy một bản tóm tắt của các chỉnh sửa trong kho lưu trữ sử dụng trong các bản sao làm việc của bạn. Đây là những *cam kết* sửa đổi, chứ không phải *cập nhật* sửa đổi, chúng đại diện cho một loạt các phiên bản mà các tập tin này được cam kết cuối cùng, không phải là phiên bản mà chúng đã được cập nhật. Lưu ý rằng đây là phạm vi sửa đổi thể hiện chỉ áp dụng cho các mục hiển thị, chứ không phải toàn bộ bản sao làm việc. Nếu bạn muốn xem thông tin đó cho toàn bộ bản sao làm việc, bạn phải đánh hộp kiểm **Hiển thị các tập tin chưa sửa đổi**.



### Mẹo

Nếu bạn muốn xem theo cách phẳng bản sao làm việc của bạn, tức là hiển thị tất cả các file và thư mục ở mọi cấp của hệ thống phân cấp thư mục, thì hộp thoại **Kiểm tra cho sửa đổi** là cách dễ nhất để đạt được điều đó. Chỉ cần kiểm tra hộp kiểm **Hiển thị các tập tin chưa sửa đổi** để hiển thị tất cả các tập tin trong bản sao làm việc của bạn.



### Sửa chữa đổi tên bên ngoài

Đôi khi các tập tin được đổi tên từ bên ngoài của Subversion, và chúng hiển thị trong danh sách tập tin như một tập tin bị mất và một tập tin không phiên bản. Để tránh mất lịch sử bạn cần phải thông báo cho Subversion về kết nối. Đơn giản chỉ cần chọn cả hai tên (bị mất) và tên mới (không phiên bản) và sử dụng **Trình Đơn Ngữ Cảnh** → **Sửa chữa Di chuyển** để ghép hai tập tin như đổi tên.



### Sửa chữa bản sao bên ngoài

Nếu bạn đã thực hiện một bản sao của một tập tin nhưng quên sử dụng lệnh Subversion để làm như vậy, bạn có thể sửa chữa bản sao để các tập tin mới không bị mất lịch sử của nó. Đơn giản chỉ cần chọn tên cũ (bình thường hoặc sửa đổi) và tên mới (không phiên bản) và sử dụng **Trình Đơn Ngữ Cảnh** → **Sửa chữa Sao chép** ghép hai tập tin như một bản.

## 4.7.5. Xem các khác biệt

Thường thì bạn muốn nhìn bên trong các tập tin của bạn, để có một cái nhìn vào những gì bạn đã thay đổi. Bạn có thể thực hiện điều này bằng cách chọn một tập tin đã thay đổi, và chọn **Khác** từ trình đơn ngữ cảnh của TortoiseSVN. Điều này bắt đầu trình xem khác biệt bên ngoài, mà sau đó sẽ so sánh các tập tin hiện tại cùng với bản sao nguyên sơ (bản sửa đổi BASE), đã được lưu trữ sau khi kiểm tra ra hoặc cập nhật cuối cùng.



### Mẹo

Ngay cả khi không phải bên trong một bản sao làm việc hoặc khi bạn có nhiều phiên bản của tập tin nằm xung quanh, bạn vẫn có thể hiển thị khác biệt:

Chọn hai tập tin mà bạn muốn so sánh trong trình thám hiểm (ví dụ như sử dụng **Ctrl** và con chuột) và chọn **Khác** từ trình đơn ngữ cảnh của TortoiseSVN. Các tập tin được click chuột cuối cùng (tập tin được tập trung, tức là hình chữ nhật với nét rời) sẽ được coi là tập tin đến sau.

## 4.8. Danh sách Thay đổi

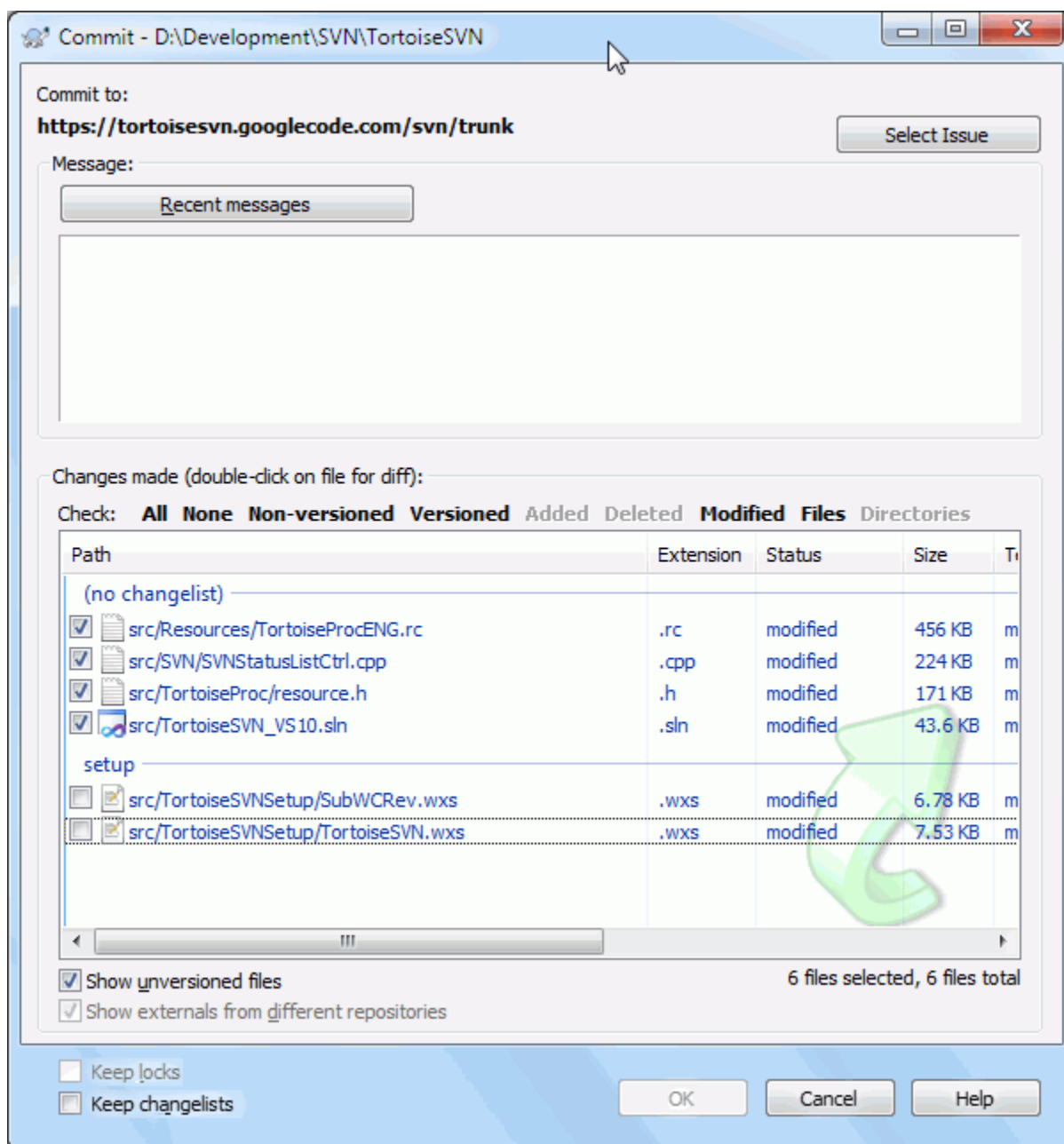
Trong một thế giới lý tưởng, bạn bao giờ cũng chỉ làm một việc tại một thời điểm, và bản sao làm việc của bạn chỉ chứa một tập hợp các thay đổi hợp lý. OK, trở lại với thực tế. Thường xảy ra rằng bạn phải làm việc trên một số nhiệm vụ không liên quan cùng một lúc, và khi bạn nhìn vào hộp thoại cam kết, tất cả các thay đổi được trộn lẫn trong với nhau. Tính năng *danh sách thay đổi* giúp bạn nhóm các tập tin với nhau, làm cho nó dễ dàng hơn để xem những gì bạn đang làm. Tất nhiên điều này chỉ có thể được nếu những thay đổi không chồng chéo lên nhau. Nếu hai nhiệm vụ khác nhau ảnh hưởng đến cùng một tập tin, không có cách nào để tách các thay đổi.

Bạn có thể thấy danh sách thay đổi ở một vài nơi, nhưng những cái quan trọng nhất là hộp thoại cam kết và hộp thoại kiểm tra sửa đổi mới. Chúng ta hãy bắt đầu trong hộp thoại kiểm tra, sửa đổi sau khi bạn đã làm việc trên một số tính năng và nhiều tập tin. Khi bạn mở hộp thoại đầu tiên, tất cả các tập tin thay đổi được liệt kê với nhau. Giả sử bây giờ bạn muốn tổ chức các sự vật và nhóm các tập tin theo tính năng.

Chọn một hoặc nhiều tập tin và sử dụng **Trình đơn ngữ cảnh** → **Di chuyển đến danh sách thay đổi** để thêm một mục vào danh sách thay đổi. Ban đầu sẽ có không danh sách thay đổi, do đó, lần đầu tiên bạn làm điều này, bạn sẽ tạo ra một danh sách thay đổi mới. Cung cấp cho nó tên trong đó mô tả mục đích bạn đang sử dụng nó, và nhấn **OK**. Hộp thoại sẽ thay đổi để hiển thị các nhóm hạng mục.

Một khi bạn đã tạo ra một danh sách thay đổi bạn có thể kéo và thả các mục vào nó, hoặc từ danh sách thay đổi khác, hoặc từ Windows Explorer. Kéo từ Explorer có thể có ích vì nó cho phép bạn thêm các mục vào một danh sách thay đổi trước khi tập tin được sửa đổi. Bạn có thể làm điều đó từ hộp thoại kiểm tra-cho-thay đổi, nhưng chỉ bằng cách hiển thị tất cả các tập tin chưa sửa đổi.





**Hình 4.15. Hộp thoại cam kết với danh sách thay đổi**

Trong hộp thoại cam kết, bạn có thể nhìn thấy những tập tin tương tự, nhóm theo danh sách thay đổi. Ngoài việc đưa ra một chỉ dấu của các nhóm ngay lập tức bằng hình ảnh, bạn cũng có thể sử dụng các tiêu đề của nhóm để lựa chọn các tập tin vào cam kết.

Trên XP, có một trình đơn ngữ cảnh khi bạn nhấp chuột phải vào một nhóm trong đó cung cấp cho bạn sự lựa chọn để chọn hoặc bỏ chọn tất cả các mục của nhóm. Trên Vista tuy nhiên trình đơn ngữ cảnh là không cần thiết. Click vào tiêu đề nhóm để chọn tất cả các mục, sau đó chọn một trong các mục được lựa chọn để chọn tất cả.

TortoiseSVN để dành một tên danh sách thay đổi để sử dụng riêng của nó, cụ thể là bỏ qua lúc cam kết. Điều này được sử dụng để đánh dấu các tập tin là phiên bản mà bạn gần như không bao giờ muốn cam kết mặc dù chúng có thay đổi cục bộ. Tính năng này được mô tả trong [Phần 4.4.4, “Loại trừ mục khỏi Danh mục Cam kết”](#).

Khi bạn thực hiện cam kết các tập tin thuộc một danh sách thay đổi thì thông thường bạn sẽ mong đợi rằng việc là thành viên của danh sách thay đổi không còn cần thiết. Vì vậy, theo mặc định, các tập tin được gỡ bỏ từ danh

sách thay đổi tự động lúc cam kết. Nếu bạn muốn giữ lại các tập tin trong danh sách thay đổi của nó, sử dụng hộp kiểm **Giữ danh sách thay đổi** ở phía dưới của hộp thoại cam kết.



## Mẹo

Danh sách thay đổi hoàn toàn là một tính năng ở máy khách địa phương. Tạo ra và loại bỏ danh sách thay đổi sẽ không ảnh hưởng đến kho lưu trữ, cũng không ảnh hưởng bất cứ bản sao làm việc của ai khác. Chúng chỉ đơn giản là một cách thuận tiện để bạn có thể tổ chức các tập tin của bạn.

## 4.9. Hộp thoại Tường trình Chính sửa

Đối với mỗi thay đổi bạn thực hiện và cam kết, bạn nên cung cấp một thông báo tường trình cho sự thay đổi đó. Bằng cách đó sau này bạn có thể tìm hiểu những thay đổi bạn đã thực hiện và lý do tại sao, và bạn có một bản tường trình chi tiết cho quá trình phát triển của bạn.

Hộp thoại tường trình sửa đổi lấy tất cả những thông báo tường trình và hiển thị chúng cho bạn. Màn hình hiển thị được chia thành 3 bảng.

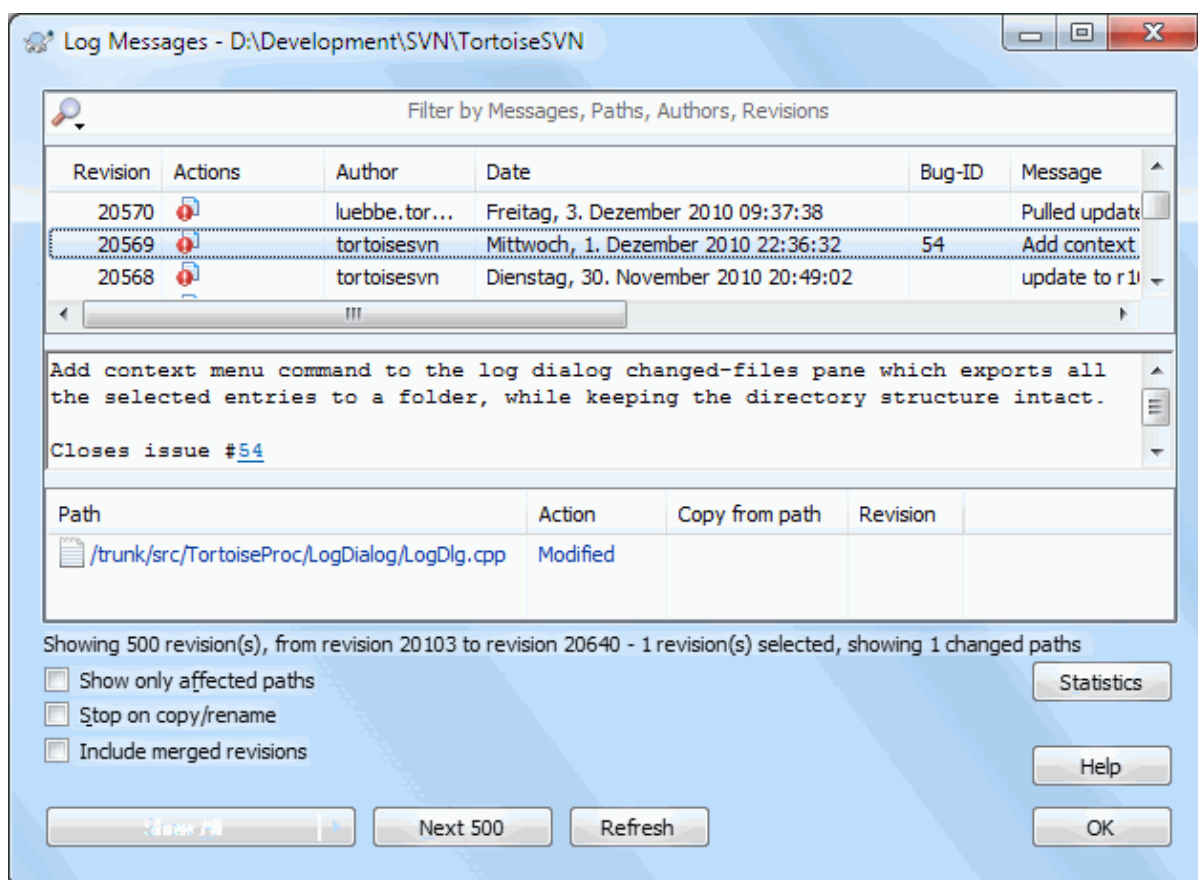
- Các cửa sổ đầu cho thấy một danh sách các phiên bản thay đổi tập tin / thư mục đã được cam kết. Bản tóm tắt này bao gồm ngày tháng và thời gian, người cam kết sửa đổi và bắt đầu của các thông báo tường trình.

Các dòng hiển thị trong màu xanh chỉ ra rằng một cái gì đó đã được sao chép vào dòng phát triển này (có lẽ từ một chi nhánh).

- Bảng ở giữa cho thấy các thông báo tường trình đầy đủ cho các bản sửa đổi đã chọn.
- Khung phía dưới hiển thị danh sách tất cả các tập tin và thư mục đã được thay đổi như là một phần của bản sửa đổi được lựa chọn.

Nhưng nó còn làm nhiều hơn thế - nó cung cấp các lệnh trình đơn ngữ cảnh mà bạn có thể sử dụng để có được thông tin nhiều hơn về lịch sử của dự án.

### 4.9.1. Hộp thoại Tường Trình Sửa Đổi



**Hình 4.16. Hộp thoại Tường Trình Sửa Đổi**

Có nhiều nơi mà từ đó bạn có thể hiển thị hộp thoại Tường Trình:

- Từ trình đơn ngữ cảnh phụ của TortoiseSVN
- Từ trang thuộc tính
- Từ hộp thoại tiến độ sau khi một bản cập nhật đã hoàn tất. Sau đó hộp thoại Tường Trình chỉ thể hiện các phiên bản đã được thay đổi kể từ khi cập nhật cuối cùng của bạn

Nếu kho lưu trữ không sẵn sàng, bạn sẽ thấy hộp thoại **Bạn muốn ra ngoài mạng không?**, được mô tả trong [Phần 4.9.10, “Chế Độ Ngoại Tuyến”](#).

### 4.9.2. Hoạt động Tường Trình Sửa Đổi

Các bảng trên cùng có một cột **Hoạt động** chứa các biểu tượng tóm tắt những gì đã được thực hiện trong bản chỉnh sửa đó. Có bốn biểu tượng khác nhau, được hiển thị trong cột riêng của mình.



Nếu một phiên bản sửa đổi một tập tin hoặc thư mục, biểu tượng *sửa đổi* được hiển thị trong cột đầu tiên.



Nếu một sửa đổi được thêm vào một tập tin hoặc thư mục, biểu tượng *thêm* được hiển thị trong cột thứ hai.

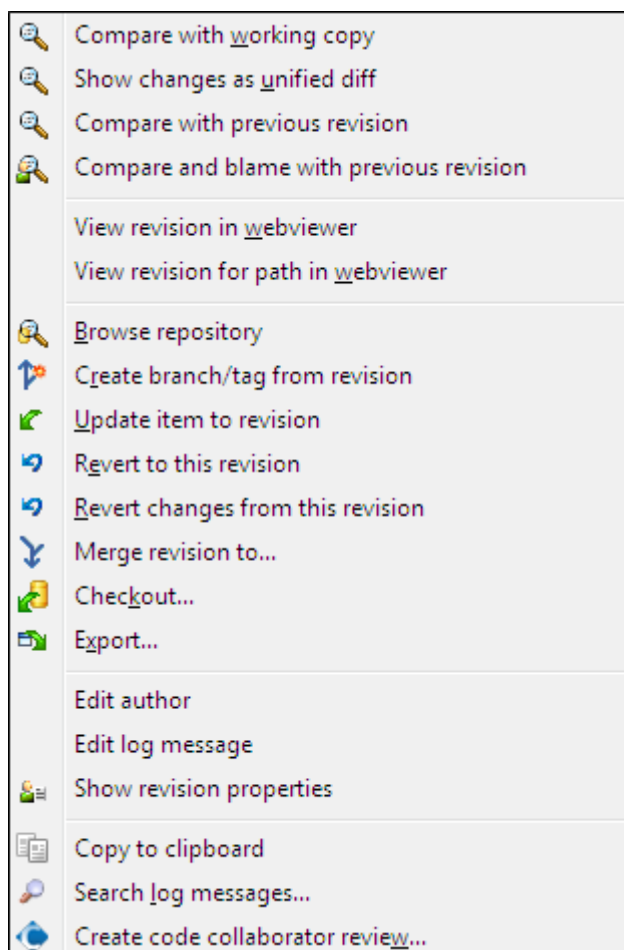


Nếu một phiên bản đã xóa một tập tin hoặc thư mục, *biểu tượng xóa* được hiển thị trong cột thứ ba.



Nếu một phiên bản thay thế một tập tin hoặc thư mục, *biểu tượng thay thế* được hiển thị trong cột thứ tư.

### 4.9.3. Lấy thông tin thêm



**Hình 4.17. Bảng trên cùng của hộp thoại tường trình sửa đổi với trình đơn ngữ cảnh**

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.

#### So sánh với bản sao làm việc

So sánh các bản sửa đổi được lựa chọn với bản sao làm việc của bạn. Diff-Tool mặc định của là TortoiseMerge được cung cấp với TortoiseSVN. Nếu hộp thoại tường trình là cho một thư mục, điều này sẽ chỉ cho bạn một danh sách các tập tin thay đổi, và cho phép bạn xem lại các thay đổi được thực hiện đến từng tập tin riêng lẻ.

#### So sánh và đồ lỗi với bản làm việc Cơ Sở

Đồ lỗi cho bản sửa đổi được lựa chọn, và tập tin trong bản làm việc cơ sở của bạn và so sánh các báo cáo đồ lỗi bằng cách sử dụng một công cụ trực quan khác. Đọc [Phần 4.23.2, “Đồ lỗi cho khác biệt”](#) để biết thêm chi tiết. (chỉ có các tập tin)

#### Hiển thị những thay đổi như khác biệt thống nhất

Xem các thay đổi được thực hiện trong bản sửa đổi được lựa chọn như là một tập Diff hợp nhất (định dạng vá lỗi GNU). Điều này cho thấy chỉ có sự khác biệt với một vài dòng của bối cảnh. Đó là khó đọc hơn so với so sánh tập tin một cách trực quan, nhưng sẽ hiển thị tất cả thay đổi tập tin với nhau trong một định dạng nhỏ gọn.

Nếu bạn giữ phím **Shift** khi nhấp chuột vào các mục trình đơn, một hộp thoại hiện lên trước tiên, nơi bạn có thể thiết lập các tùy chọn khác biệt thông nhất. Các tùy chọn này bao gồm khả năng để bỏ qua những thay đổi trong các kết thúc dòng và khoảng trắng.

#### So sánh với phiên bản trước

So sánh các sửa đổi được lựa chọn với các bản trước. Điều này làm việc một cách tương tự như so sánh với bản sao làm việc của bạn. Đối với các thư mục tùy chọn này đầu tiên sẽ hiển thị hộp thoại các tập tin thay đổi cho phép bạn chọn tập tin để so sánh.

#### So sánh và đổ lỗi với phiên bản trước

Hiển thị hộp thoại tập tin thay đổi cho phép bạn chọn các tập tin. Đổ lỗi cho các phiên bản được lựa chọn, và sửa đổi trước đây, và so sánh các kết quả bằng cách sử dụng một công cụ trực quan khác. (chỉ cho các thư mục)

#### Lưu sửa đổi ...

Lưu sửa đổi được lựa chọn vào một tập tin để bạn có một phiên bản cũ của tập tin đó. (chỉ cho tập tin)

#### Mở / Mở với ...

Mở tập tin được lựa chọn, hoặc là với trình xem mặc định cho loại tập tin, hoặc với một chương trình mà bạn chọn. (chỉ cho tập tin)

#### Đổ lỗi cho ...

Đổ lỗi cho các tập tin đến sửa đổi được lựa chọn. (chỉ cho tập tin)

#### Duyệt kho lưu trữ

Mở trình duyệt kho để kiểm tra các tập tin hoặc thư mục được chọn trong kho lưu trữ như nó đã được làm với bản sửa đổi được lựa chọn.

#### Tạo chi nhánh / thẻ từ sửa đổi

Tạo một chi nhánh hoặc thẻ từ một phiên bản được lựa chọn. Điều này rất hữu ích ví dụ như nếu bạn quên để tạo ra một thẻ và đã cam kết một số thay đổi mà lẽ ra không được vào bản phát hành đó.

#### Cập nhật mục đến bản sửa đổi

Cập nhật bản sao làm việc của bạn với phiên bản được lựa chọn. Hữu ích nếu bạn muốn có bản sao của bạn làm việc phản ánh một thời gian trong quá khứ, hoặc nếu có tiếp tục được cam kết vào kho lưu trữ và bạn muốn cập nhật bản sao làm việc từng bước một tại một thời điểm. Tốt nhất là cập nhật toàn bộ thư mục bản trong bản sao làm việc của bạn, không phải chỉ là một tập tin, nếu không bản sao làm việc của bạn có thể không phù hợp.

Nếu bạn muốn phục hồi lại một sự thay đổi trước đó vĩnh viễn, thay vào đó sử dụng **Hoàn nguyên về phiên bản này**.

#### Hoàn nguyên về phiên bản này

Hoàn nguyên về một phiên bản trước đó. Nếu bạn có một số thay đổi, và sau đó quyết định rằng bạn thực sự muốn mọi thứ trở lại như trong bản sửa đổi N, đây là lệnh bạn cần. Những thay đổi được hoàn tác trong bản sao làm việc của bạn, do đó hoạt động này *không* ảnh hưởng đến kho lưu trữ cho đến khi bạn cam kết những thay đổi. Lưu ý rằng điều này sẽ lùi lại *tất cả* thay đổi được thực hiện sau trong bản sửa đổi được lựa chọn, thay thế các tập tin / thư mục với các phiên bản trước đó.

Nếu bản sao làm việc của bạn trong trạng thái chưa sửa đổi, sau khi bạn thực hiện hành động này, bản sao làm việc của bạn sẽ hiển thị như đã được sửa đổi. Nếu bạn đã có thay đổi địa phương, lệnh này sẽ hợp nhất các thay đổi *hoàn tác* vào bản sao làm việc của bạn.

Những gì đang xảy ra trong nội bộ là Subversion thực hiện một đảo ngược hợp nhất của tất cả các thay đổi được thực hiện sau khi bản sửa đổi được lựa chọn, hoàn tác các tác động của những cam kết trước đó.

Nếu sau khi thực hiện hành động này, bạn quyết định rằng bạn muốn *hoàn tác các hoàn tác* và nhận được bản sao làm việc của bạn trở lại trạng thái chưa sửa đổi của nó trước đó, bạn nên sử dụng **TortoiseSVN** → **Hoàn nguyên** từ bên trong Windows Explorer, sẽ loại bỏ các thay đổi được thực hiện bởi hành động đảo kết hợp này.

Nếu bạn chỉ đơn giản là muốn nhìn thấy những gì một tập tin hoặc thư mục trông giống như thế nào tại một phiên bản trước đó, thay vào đó sử dụng **Cập nhật đến sửa đổi** hoặc **Lưu sửa đổi như ...** .

#### Hoàn nguyên các thay đổi từ phiên bản này

Hoàn tác thay đổi từ đó đã được thực hiện trong phiên bản được lựa chọn. Những thay đổi được hoàn tác trong bản sao làm việc của bạn, do đó hoạt động này *không* ảnh hưởng đến kho lưu trữ ở tất cả! Lưu ý rằng điều này sẽ chỉ hoàn tác các thay đổi được thực hiện trong bản chỉnh sửa; nó không thay thế bản sao của bạn làm việc với toàn bộ tập tin trong bản sửa đổi trước đó. Điều này là rất hữu ích để hoàn tác một sự thay đổi trước đó kể từ khi các thay đổi khác không liên quan đã được thực hiện.

Nếu bản sao làm việc của bạn trong trạng thái chưa sửa đổi, sau khi bạn thực hiện hành động này, bản sao làm việc của bạn sẽ hiển thị như đã được sửa đổi. Nếu bạn đã có thay đổi địa phương, lệnh này sẽ hợp nhất các thay đổi *hoàn tác* vào bản sao làm việc của bạn.

Những gì đang xảy ra trong nội bộ là Subversion thực hiện đảo ngược một hợp nhất của bản sửa đổi, hoàn tác tác động của nó từ một cam kết trước đó.

Bạn có thể *Hoàn tác các hoàn tác* như mô tả ở trên trong **Hoàn nguyên về phiên bản này** .

#### Kết hợp các sửa đổi ...

Hợp nhất các bản sửa đổi được lựa chọn vào một bản sao làm việc khác nhau. Một hộp thoại lựa chọn thư mục cho phép bạn chọn các bản sao làm việc để sáp nhập vào, nhưng sau đó không có hộp thoại xác nhận, cũng không có bất kỳ cơ hội để thử nghiệm hợp nhất. Đó là một ý tưởng tốt để kết hợp thành một bản sao hoạt động chưa sửa đổi để bạn có thể phục hồi các thay đổi nếu nó không hoạt động! Đây là một tính năng hữu ích nếu bạn muốn kết hợp các phiên bản được lựa chọn từ một chi nhánh khác.

#### Kiểm tra ra ...

Thực hiện một kiểm tra ra mới của thư mục được lựa chọn tại bản sửa đổi được lựa chọn. Việc này sẽ trả về một hộp thoại cho bạn để xác nhận địa chỉ URL và sửa đổi, và chọn một vị trí cho kiểm tra ra.

#### Xuất khẩu ...

Xuất khẩu các tập tin / thư mục được lựa chọn tại bản sửa đổi được lựa chọn. Việc này sẽ trả về một hộp thoại cho bạn để xác nhận địa chỉ URL và sửa đổi, và chọn một vị trí để xuất khẩu.

#### Chỉnh sửa tác giả / thông điệp tường trình

Chỉnh sửa thông điệp tường trình hoặc tác giả gắn liền với một cam kết trước đó. Đọc **Phần 4.9.7, “Thay đổi Thông điệp Tường trình và Tác giả”** để tìm hiểu cách làm việc này.

#### Hiển thị thuộc tính chỉnh sửa

Xem và chỉnh sửa bất kỳ thuộc tính chỉnh sửa, chứ không phải chỉ thông điệp tường trình và tác giả. Tham khảo **Phần 4.9.7, “Thay đổi Thông điệp Tường trình và Tác giả”** .

#### Sao chép vào clipboard

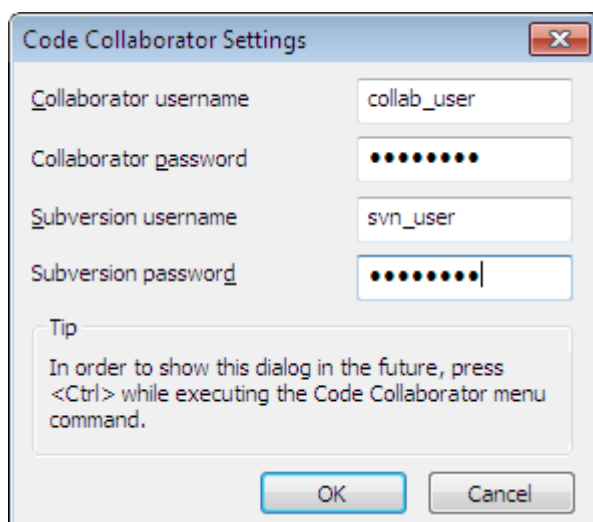
Sao chép các chi tiết tường trình của các phiên bản được lựa chọn vào clipboard. Điều này sẽ sao chép số sửa đổi, tác giả, ngày, thông điệp tường trình và danh sách các mục thay đổi cho mỗi chỉnh sửa.

#### Tìm kiếm các thông điệp tường trình ...

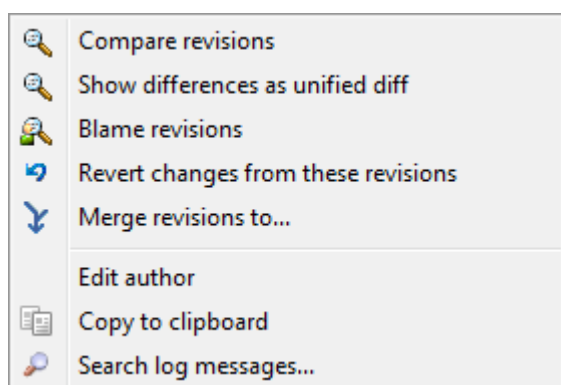
Tìm kiếm thông điệp tường trình cho các văn bản bạn nhập vào. Điều này tìm kiếm các thông điệp tường trình mà bạn đã nhập và cả các tóm tắt hành động được tạo ra bởi Subversion (được hiển thị ở khung bên dưới). Tìm kiếm không phân biệt chữ hoa.

#### 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.



**Hình 4.18. The Code Collaborator Settings Dialog**



**Hình 4.19. Bảng trên cùng của trình đơn ngữ cảnh trong 2 bản sửa đổi được lựa chọn**

Nếu bạn chọn hai phiên bản cùng một lúc (bằng cách sử dụng phím thông thường **Ctrl** -Sửa đổi), trình đơn ngữ cảnh thay đổi và cung cấp cho bạn ít tùy chọn hơn:

#### So sánh các bản sửa đổi

So sánh hai bản sửa đổi được lựa chọn bằng cách sử dụng một công cụ tìm khác biệt trực quan. Diff-Tool mặc định là TortoiseMerge được cung cấp với TortoiseSVN.

Nếu bạn chọn tùy chọn này cho một thư mục, một hộp thoại tiếp tục bật lên danh sách các tập tin thay đổi và cung cấp cho bạn các tùy chọn khác hơn nữa. Tìm hiểu thêm về hộp thoại So sánh Sửa đổi [Phần 4.10.3](#), “So sánh các thư mục”.

#### Đồ lỗi cho các phiên bản

Đồ lỗi cho hai phiên bản và so sánh các báo cáo đồ lỗi bằng cách sử dụng một công cụ khác biệt trực quan. Đọc [Phần 4.23.2](#), “Đồ lỗi cho khác biệt” để biết thêm chi tiết.

#### Hiện thị khác biệt như là khác biệt thống nhất

Xem sự khác biệt giữa hai phiên bản được lựa chọn như là một tập tin Diff hợp nhất. Điều này làm tốt cho các tập tin và thư mục.

#### Sao chép vào clipboard

Sao chép thông báo tường trình vào clipboard như mô tả ở trên.

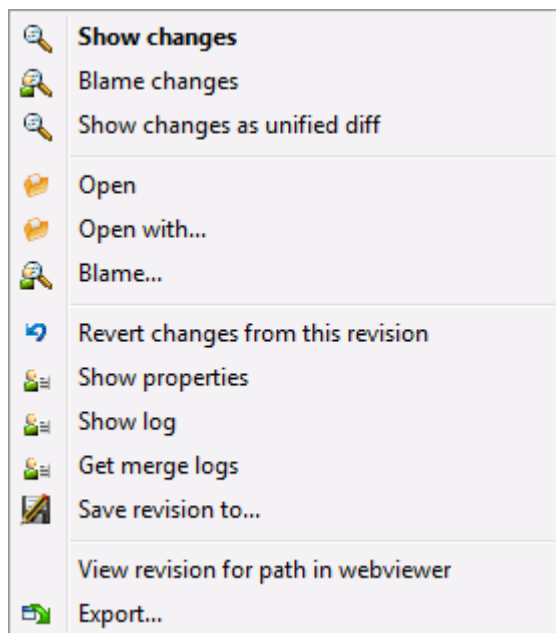
#### Tìm kiếm các thông điệp tường trình ...

Tìm kiếm thông điệp tường trình như đã mô tả ở trên.

Nếu bạn chọn hai hoặc nhiều sửa đổi (bằng cách sử dụng thông thường **Ctrl** hoặc **Shift** bô), trình đơn ngữ cảnh sẽ bao gồm một mục vào Hoàn nguyên tất cả các thay đổi đã được thực hiện trong các phiên bản được lựa chọn. Đây là cách dễ nhất để quay lui một nhóm các phiên bản trong một lần.

Bạn cũng có thể chọn để kết hợp các phiên bản được lựa chọn thành một bản sao làm việc, như mô tả ở trên.

Nếu tất cả các sửa đổi được lựa chọn có cùng tác giả, bạn có thể chỉnh sửa các tác giả của tất cả những sửa đổi trong một lần.



**Hình 4.20. Khung hộp thoại tường trình dưới đáy với trình đơn ngữ cảnh**

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

Hiện thị những thay đổi

Hiện thị những thay đổi được thực hiện trong phiên bản được lựa chọn cho tập tin được lựa chọn.

Đồ lỗi cho sự thay đổi

Đồ lỗi cho bản sửa đổi được lựa chọn và bản sửa đổi trước cho các tập tin được lựa chọn, và so sánh các báo cáo đồ lỗi bằng cách sử dụng một công cụ tìm khác biệt trực quan khác. Đọc [Phần 4.23.2, “Đồ lỗi cho khác biệt”](#) để biết thêm chi tiết.

Hiện thị như khác biệt thống nhất

Xem những thay đổi tập tin ở định dạng khác biệt thống nhất. Trình đơn ngữ cảnh này chỉ có sẵn cho các tập tin được hiển thị như *sửa đổi* .

Mở / Mở với ...

Mở tập tin được lựa chọn, hoặc là với trình xem mặc định cho loại tập tin, hoặc với một chương trình mà bạn chọn.

Đồ lỗi cho ...

Mở hộp thoại đồ lỗi, cho phép bạn đồ lỗi cho việc sửa đổi được lựa chọn.

Hoàn nguyên các thay đổi từ phiên bản này

Phục hồi các thay đổi được thực hiện trong các tập tin được chọn trong bản chỉnh sửa đó.

Hiện thị thuộc tính

Xem các thuộc tính Subversion cho các mục được chọn.

Hiện tường trình

Hiện thị các tường trình sửa đổi cho các tập tin được lựa chọn duy nhất.



#### Lấy các trường trình hợp nhất

Hiển thị trường trình sửa đổi cho các tập tin duy nhất được lựa chọn, bao gồm cả những thay đổi sáp nhập. Tìm hiểu nhiều hơn trong [Phần 4.9.6, “Hợp nhất tính năng theo dõi”](#).

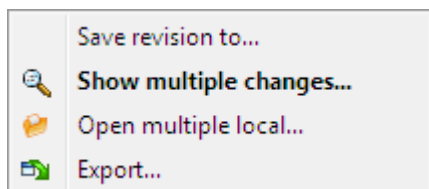
#### Lưu sửa đổi ...

Lưu sửa đổi được lựa chọn vào một tập tin để bạn có một phiên bản cũ của tập tin đó.

#### Xuất khẩu ...

Xuất khẩu các mục được lựa chọn trong phiên bản này vào một thư mục, bảo quản hệ thống phân cấp tập tin.

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



### Hình 4.21. The Log Dialog Bottom Pane with Context Menu When Multiple Files Selected.

#### Lưu sửa đổi ...

Lưu sửa đổi được lựa chọn vào một tập tin để bạn có một phiên bản cũ của tập tin đó.

#### Show multiple changes...

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

#### Open multiple local...

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

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

#### Xuất khẩu ...

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.



### Mẹo

Bạn có thể nhận thấy rằng đôi khi chúng ta nhắc đến thay đổi và lúc khác là khác biệt. Sự khác biệt là gì?

Subversion sử dụng số sửa đổi cho 2 việc khác nhau. Sửa đổi thường đại diện cho trạng thái của các kho lưu trữ tại một điểm trong thời gian, nhưng nó cũng có thể được sử dụng để đại diện cho tập

hợp thay đổi mà tạo ra rằng sửa đổi, ví dụ như “ Làm trong r1234 ” có nghĩa thay đổi cam kết trong r1234 thực hiện tính năng X. Để làm cho nó rõ ràng hơn là nghĩa nào đang được sử dụng, chúng tôi sử dụng hai thuật ngữ khác nhau.

Nếu bạn chọn hai phiên bản N và M, trình đơn ngữ cảnh sẽ cung cấp để hiển thị *sự khác biệt* giữa hai phiên bản. Trong thuật ngữ của Subversion đây là `diff-r M: N` .

Nếu bạn chọn một sửa đổi N, trình đơn ngữ cảnh sẽ cung cấp để hiển thị *thay đổi* được thực hiện trong bản chỉnh sửa đó. Trong thuật ngữ Subversion đây là `diff-r N-1:N` hoặc `diff -c N` .

Khung phía dưới hiển thị các tập tin thay đổi trong tất cả các sửa đổi được lựa chọn, do đó, các trình đơn ngữ cảnh luôn luôn cung cấp để hiển thị *thay đổi* .

#### 4.9.4. Lấy thêm thông điệp tường trình

Hộp thoại Tường trình không phải luôn luôn hiển thị tất cả thay đổi từng được thực hiện vì một số lý do:

- Đối với một kho lớn có thể có hàng trăm hoặc thậm chí hàng ngàn những thay đổi và lấy tất cả có thể mất một thời gian dài. Thông thường bạn chỉ quan tâm đến những thay đổi gần đây. Theo mặc định, số lượng thông điệp tường trình lấy được giới hạn đến 100, nhưng bạn có thể thay đổi giá trị này trong TortoiseSVN → Thiết lập ( Phần 4.30.1.2, “Các Thiết Lập Hộp Thoại TortoiseSVN 1” ),
- Khi hộp Dừng trên sao chép/đổi tên được kiểm tra, Hiển Thị Tường Trình sẽ dừng lại ở điểm mà tập tin hoặc thư mục đã được sao chép từ một nơi nào đó khác trong kho. Điều này có thể hữu ích khi tìm kiếm tại các chi nhánh (hoặc tags) như nó dừng lại ở thư mục gốc của chi nhánh, và đưa ra một chỉ dẫn nhanh chóng của các thay đổi được thực hiện trong đó chi nhánh chỉ.

Thông thường bạn sẽ muốn để tùy chọn này không được kiểm tra. TortoiseSVN nhớ trạng thái của hộp kiểm, vì vậy nó sẽ tôn trọng sở thích của bạn.

Khi hộp thoại Hiển thị Tường trình là gọi từ trong hộp thoại Hợp Nhất, hộp luôn được chọn theo mặc định. Điều này là do việc sáp nhập thường xuyên nhất là nhìn vào những thay đổi trên các nhánh, và vượt ra ngoài thư mục gốc của chi nhánh không có ý nghĩa trong trường hợp đó.

Lưu ý rằng Subversion hiện đang thực hiện đổi tên như một việc sao chép / xóa cặp, do đó, đặt lại tên tập tin hoặc thư mục cũng sẽ gây ra màn hình tường trình dừng lại nếu tùy chọn này được kiểm tra.

Nếu bạn muốn để xem nhiều thông điệp tường trình hơn, nhấp vào 100 tiếp theo để lấy 100 thông điệp tường trình tiếp theo. Bạn có thể lặp lại điều này nhiều lần khi cần thiết.

Bên cạnh nút này là nút đa chức năng nhớ lựa chọn cuối cùng bạn sử dụng nó cho. Nhấp chuột vào mũi tên để xem các tùy chọn khác được cung cấp.

Sử dụng Hiện Phạm Vi ... nếu bạn muốn xem một phạm vi cụ thể các phiên bản. Một hộp thoại sau đó sẽ nhắc bạn nhập vào bản sửa đổi bắt đầu và kết thúc.

Sử dụng Hiện tất cả nếu bạn muốn xem *tất cả* thông điệp tường trình từ HEAD tới lại phiên bản 1.

Để làm mới phiên bản mới nhất trong trường hợp có các cam kết trong khi hộp thoại tường trình đã được mở, nhấn phím **F5** .

Để làm mới bộ nhớ cache tường trình, nhấn phím **Ctrl-F5** .

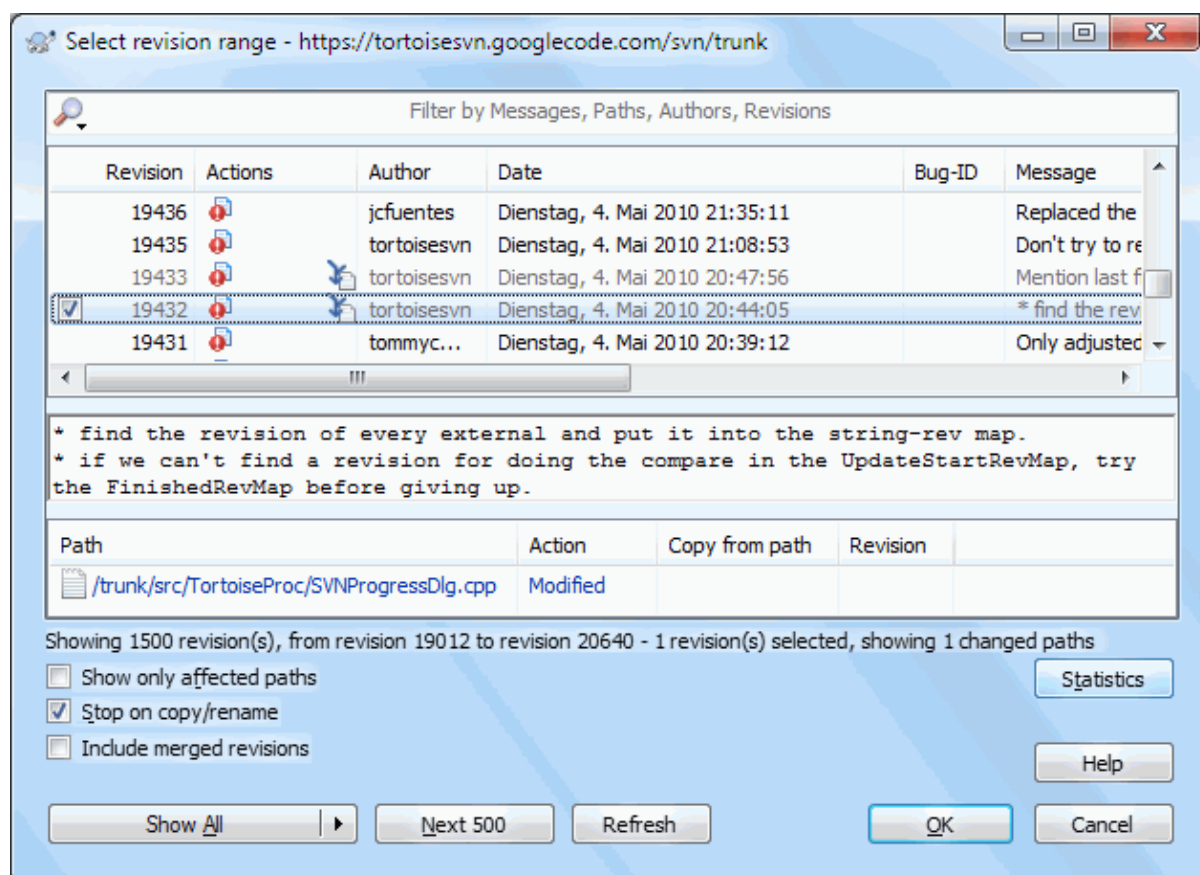
#### 4.9.5. Bản sửa đổi làm việc hiện tại

Bởi vì các hộp thoại tường trình hiển thị cho bạn các tường trình từ HEAD, không phải từ bản sao làm việc hiện tại, nó thường xảy ra rằng có những thông điệp tường trình được hiển thị cho nội dung chưa được cập nhật trong bản sao làm việc của bạn. Để giúp thực hiện điều này rõ ràng hơn, thông báo cam kết tương ứng với sửa đổi bản có trong bản sao làm việc của bạn sẽ được in đậm.

Khi bạn tường trình cho một thư bản mục sửa đổi được nhấn mạnh là bản sửa đổi cao nhất tìm thấy bất cứ nơi nào trong thư mục đó, đòi hỏi thu thập dữ liệu của bản sao làm việc. Thu thập dữ liệu diễn ra trong một mạch riêng biệt để không trì hoãn hiển thị các tường trình, nhưng là một kết quả nổi bật cho các thư mục có thể không xuất hiện ngay lập tức.

#### 4.9.6. Hợp nhất tính năng theo dõi

Subversion 1.5 và gần hơn giữ một hồ sơ của các kết hợp sử dụng thuộc tính. Điều này cho phép chúng tôi có được một lịch sử chi tiết hơn về những thay đổi được sáp nhập. Ví dụ, nếu bạn phát triển một tính năng mới trên một chi nhánh và sau đó hợp nhất chi nhánh trở lại với thân cây, sự phát triển tính năng sẽ xuất hiện trên nhật ký thân cây như một cam kết duy nhất cho việc hợp nhất, mặc dù có thể có 1000 cam kết trong quá trình phát triển chi nhánh.



**Hình 4.22. Hộp Thoại Tường Trình Hiển Thị Theo Dõi Hợp Nhất Các Bản Sửa Đổi**

Nếu bạn muốn xem chi tiết trong đó các bản sửa đổi đã được sáp nhập như là một phần của cam kết, sử dụng hộp kiểm Bao gồm các phiên bản được sáp nhập. Điều này sẽ lấy các thông điệp tường trình một lần nữa, nhưng cũng sẽ xen kẽ các thông điệp tường trình từ các phiên bản đã được sáp nhập. Sửa đổi được sáp nhập được thể hiện trong màu xám bởi vì chúng đại diện cho các thay đổi được thực hiện trên một phần khác nhau của cây.

Tất nhiên, sáp nhập không bao giờ là đơn giản! Trong quá trình phát triển tính năng trên chi nhánh có thể thỉnh thoảng sẽ có sáp nhập vào lại thân cây để giữ cho các chi nhánh đồng bộ với dòng mã chính. Vì vậy, lịch sử hợp nhất của chi nhánh cũng sẽ bao gồm một lớp lịch sử hợp nhất khác. Những lớp khác nhau được hiển thị trong hộp thoại tường trình bằng cách sử dụng các mức độ thụt đầu dòng.

#### 4.9.7. Thay đổi Thông Điệp Tường Trình và Tác giả

Các thuộc tính sửa đổi hoàn toàn khác với các thuộc tính Subversion của từng hạng mục. Revprops là các mục mô tả được kết hợp với một con số của bản sửa đổi cụ thể trong kho, chẳng hạn như thông điệp tường trình, cam kết ngày tháng và tên người cam kết (tác giả).

Đôi khi bạn có thể muốn thay đổi một thông điệp tường trình mà bạn đã từng nhập vào, có thể bởi vì có một lỗi chính tả trong đó hoặc bạn muốn cải thiện hoặc thay đổi nó vì lý do khác. Hoặc bạn muốn thay đổi tác giả của cam kết bởi vì bạn quên thiết lập chứng thực hoặc ...

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>] in the Subversion Book for details about how to do that. Read **Phần 3.3, “kịch bản móc phía máy chủ”** to find some further notes on implementing hooks on a Windows machine.

Một khi bạn đã thiết lập máy chủ của bạn với các móc cần thiết, bạn có thể thay đổi các tác giả và tin nhắn tường trình (hoặc bất kỳ revprop khác) của bất kỳ bản sửa đổi nào, sử dụng trình đơn ngữ cảnh từ cửa sổ trên cùng của hộp thoại Tường Trình. Bạn cũng có thể chỉnh sửa một tin thông điệp tường trình bằng cách sử dụng trình đơn ngữ cảnh cho các cửa sổ giữa.



### Cảnh báo

Bởi vì các thuộc tính sửa đổi Subversion không phải là phiên bản, thay đổi thuộc tính đó (ví dụ, thuộc tính thông điệp cam kết `svn:log`) sẽ ghi đè lên các giá trị trước đó của thuộc tính đó *mãi mãi* .



### Quan trọng

Kể từ khi TortoiseSVN giữ một bộ nhớ cache của tất cả các thông tin tường trình, biên tập thực hiện cho tác giả và thông điệp tường trình sẽ chỉ hiển thị trên bản cài đặt địa phương của bạn. Người sử dụng khác sử dụng TortoiseSVN vẫn sẽ thấy tác giả (cũ) và tin nhắn tường trình lưu trữ trong cache cho đến khi họ làm mới bộ nhớ cache tường trình.

## 4.9.8. Lọc Thông Điệp Tường Trình

Nếu bạn muốn hạn chế các thông điệp tường trình để chỉ hiển thị những người bạn đang quan tâm đến chứ không phải di chuyển qua một danh sách của hàng trăm, bạn có thể sử dụng các điều khiển bộ lọc ở phía trên cùng của Hộp Thoại Tường Trình. Kiểm soát ngày bắt đầu và kết thúc cho phép bạn hạn chế đầu ra cho một phạm vi ngày được biết đến. Hộp tìm kiếm cho phép bạn hiển thị các tin nhắn có chứa một cụm từ cụ thể.

Nhấp chuột vào biểu tượng tìm kiếm để chọn thông tin mà bạn muốn tìm kiếm, và để lựa chọn *regex* chế độ. Bình thường, bạn chỉ cần một cách tìm kiếm chuỗi con đơn giản, nhưng nếu bạn cần cách tìm kiếm linh hoạt hơn, bạn có thể sử dụng biểu thức thông thường. Nếu bạn di chuyển chuột trên hộp, một tooltip sẽ đưa ra gợi ý về làm thế nào để sử dụng chức năng regex, hoặc chức năng chuỗi con. Các bộ lọc hoạt động bằng cách kiểm tra xem chuỗi bộ lọc của bạn phù hợp với các mục tường trình không, và sau đó chỉ có những mục *phù hợp* với chuỗi lọc được hiển thị.

Tìm kiếm chuỗi con đơn giản làm việc trong một cách tương tự như một công cụ tìm kiếm. Các chuỗi để tìm kiếm được phân cách bởi các khoảng trống, và tất cả các chuỗi phải khớp. Bạn có thể sử dụng một dấu `-` ở đầu để xác định một chuỗi con xác định mà không tìm thấy (đảo ngược khớp cho thuật ngữ đó), và bạn có thể sử dụng `!` vào đầu biểu thức để đảo ngược khớp cho toàn bộ biểu thức. Bạn có thể sử dụng một dấu `+` ở đầu để xác định một chuỗi con mà nên được bao gồm, ngay cả khi loại trừ với một dấu `-`. Lưu ý rằng thứ tự bao gồm / loại trừ có ý nghĩa quan trọng ở đây. Bạn có thể sử dụng dấu trích dẫn nội dung bài viết để bao quanh một chuỗi có chứa khoảng trống, và nếu bạn muốn tìm kiếm một dấu ngoặc kép, bạn có thể sử dụng hai dấu ngoặc kép với nhau như là một chuỗi tự-thoát. Lưu ý rằng các dấu gạch chéo ngược *không* được sử dụng như là một ký tự thoát và không có ý nghĩa đặc biệt trong tìm kiếm chuỗi con đơn giản. Các ví dụ này sẽ làm cho việc này dễ dàng hơn:

```
Alice Bob -Eve
```

tim các chuỗi chứa cả hai Alice và Bob nhưng không chứa Eve

Alice -Bob +Eve

tim các chuỗi chứa cả hai Alice nhưng không chứa Bob, hoặc các chuỗi chứa Eve.

-Case +SpecialCase

tim các chuỗi không chứa Case, nhưng vẫn bao gồm các chuỗi chứa SpecialCase.

!Alice Bob

tim các chuỗi không chứa cả hai Alice và Bob

!-Alice -Bob

bạn có nhớ định lý De Morgan's? NOT(NOT Alice AND NOT Bob) lược giản thành (Alice OR Bob).

"Alice and Bob"

tim các biểu thức thực "Alice and Bob"

""

tim dấu ngoặc kép bất cứ đâu trong văn bản

"Alice says ""hi"" to Bob"

tim biểu thức thực "Alice says "hi" to Bob".

Mô tả việc sử dụng tìm kiếm bằng các biểu thức thông thường là vượt quá phạm vi của sổ tay này, nhưng bạn có thể tìm thấy tài liệu trực tuyến và hướng dẫn tại <http://www.regular-expressions.info/> [<http://www.regular-expressions.info/>].

Lưu ý rằng các bộ lọc này hoạt động trên các thông điệp đã được lấy ra. Chúng không kiểm soát việc tải xuống các thông điệp từ kho lưu trữ.

Bạn cũng có thể lọc tên đường dẫn ở khung bên dưới bằng cách sử dụng hộp kiểm **Chỉ hiển thị đường dẫn bị ảnh hưởng**. Đường dẫn bị ảnh hưởng là những cái có chứa các đường dẫn được sử dụng để hiển thị các tường trình. Nếu bạn lấy các tường trình cho một thư mục, có nghĩa là bất cứ điều gì trong thư mục đó hoặc dưới nó. Đối với một tập tin, nó có nghĩa là chỉ là một tập tin. Thông thường danh sách đường dẫn cho thấy bất kỳ đường dẫn khác bị ảnh hưởng bởi các cam kết tương tự, nhưng bằng màu xám. Nếu hộp được kiểm tra, những đường dẫn sẽ được ẩn.

Đôi khi cách thực hành làm việc của bạn sẽ yêu cầu các tin nhắn đăng nhập vào phải theo một định dạng cụ thể, điều đó có nghĩa rằng các văn bản mô tả những thay đổi không nhìn thấy được trong phần tóm tắt viết tắt được hiển thị trong cửa sổ trên cùng. Thuộc tính `TSVN: logsummary` có thể được sử dụng để trích xuất một phần của thông điệp tường trình được hiển thị trong cửa sổ trên cùng. Đọc [Phần 4.17.2, "Thuộc tính dự án TortoiseSVN"](#) để tìm hiểu làm thế nào để sử dụng thuộc tính này.



## Không có Định Dạng Tường Trình từ Trình Duyệt Kho

Bởi vì việc định dạng phụ thuộc vào truy cập các thuộc tính Subversion, bạn sẽ chỉ thấy các kết quả khi sử dụng một bản sao làm việc được kiểm xuất. Lấy thuộc tính từ xa là một hoạt động chậm, vì vậy bạn sẽ không thấy tính năng này trong hành động từ trình duyệt kho.

## 4.9.9. Thông Tin Thống kê

Nút Thống kê sẽ trả về một hộp hiển thị một số thông tin thú vị về các phiên bản được hiển thị trong hộp thoại Tường Trình. Điều này cho thấy có bao nhiêu tác giả đã tại nơi làm việc, bao nhiêu cam kết mà họ đã thực hiện, tiến độ theo tuần, và nhiều hơn nữa. Bây giờ bạn có thể nhìn thấy trong nháy mắt ai đã làm việc chăm chỉ nhất và ai là người đang uể oải ;-)

### 4.9.9.1. Trang Thống Kê

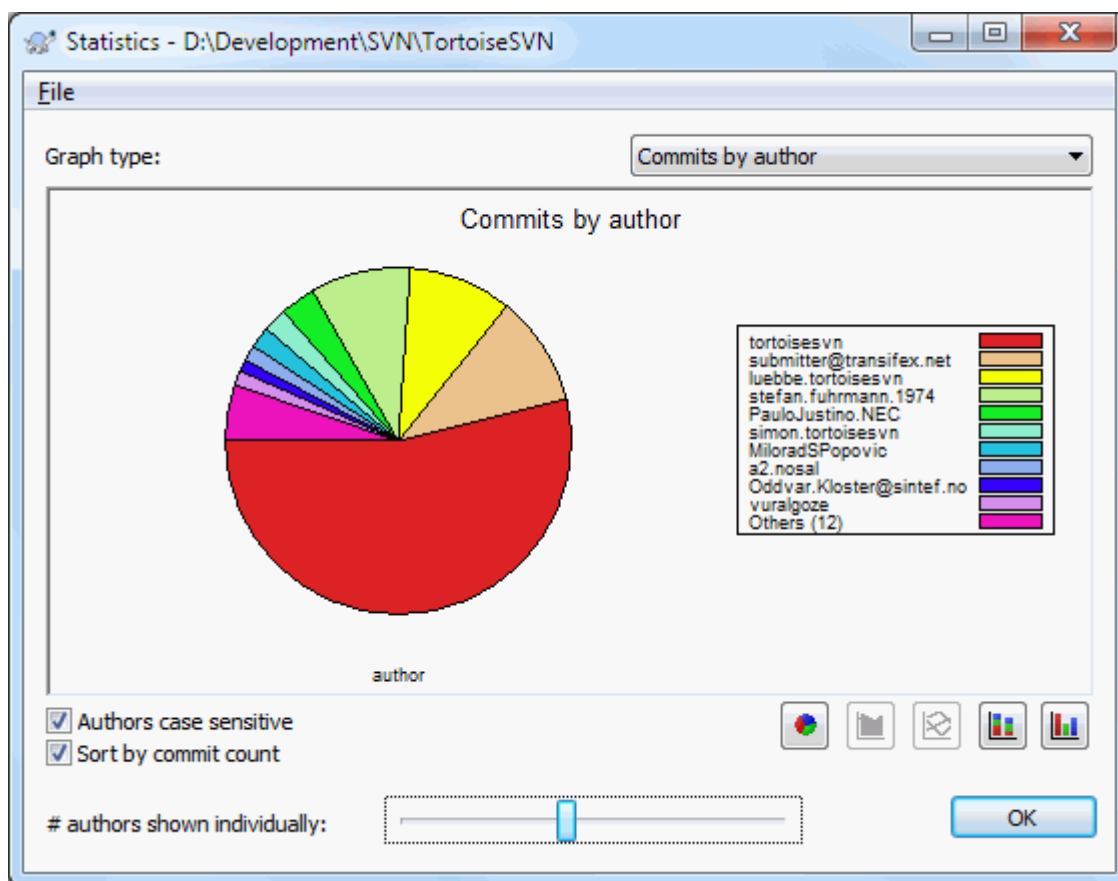
Trang này cho bạn tất cả các con số bạn có thể nghĩ đến, trong khoảng thời gian cụ thể và số lượng các phiên bản được kể đến, và một số giá trị nhỏ nhất / lớn nhất / trung bình.

### 4.9.9.2. Trang các Cam kết của Tác giả



**Hình 4.23. Biểu đồ phân phối của các Cam kết-bỏ-Tác giả**

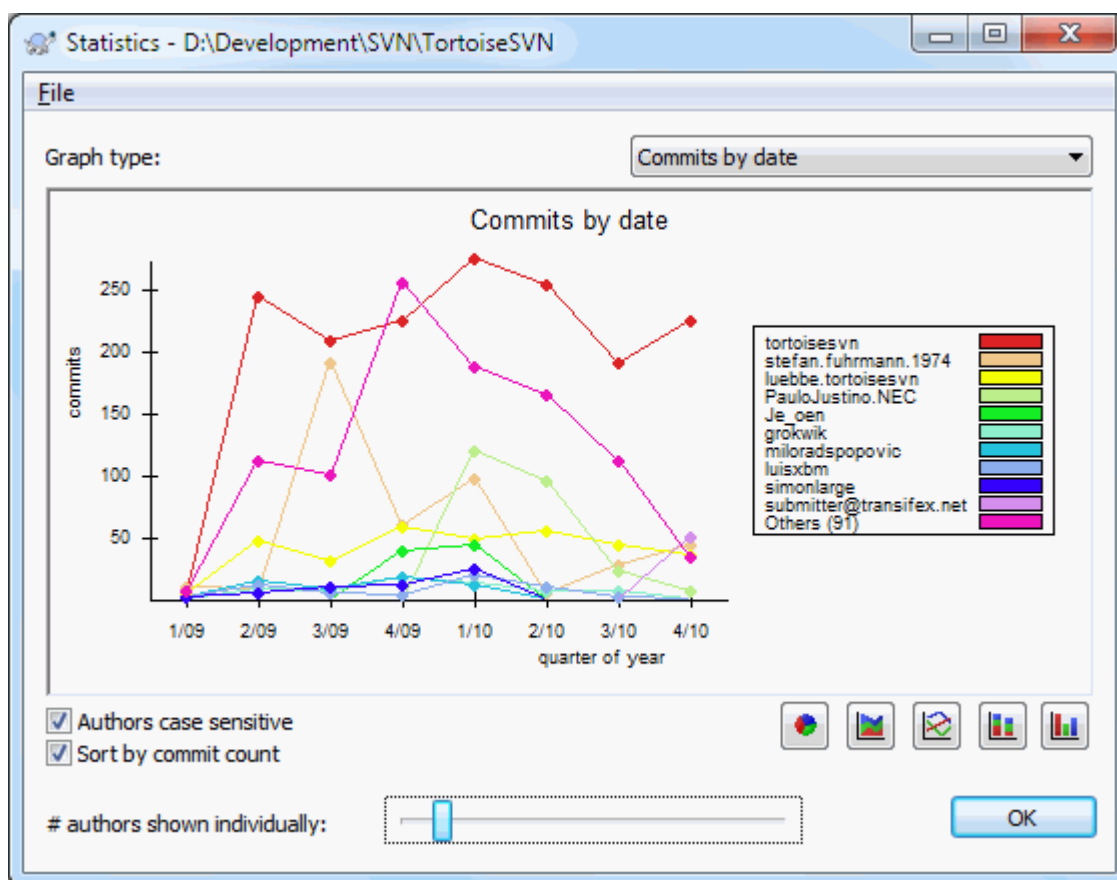
Biểu đồ này cho bạn thấy các tác giả nào đã hoạt động tích cực trên dự án như một biểu đồ phân phối đơn giản, biểu đồ phân phối xếp chồng lên nhau hoặc biểu đồ hình bánh.



**Hình 4.24. Biểu đồ hình bánh của các Cam kết-bởi-Tác giả**

Trường hợp có một vài tác giả chính và nhiều người đóng góp nhỏ, số lượng các phân đoạn nhỏ có thể làm cho đồ thị khó đọc hơn. Thanh trượt ở phía dưới cho phép bạn thiết lập một ngưỡng (như là một tỷ lệ phần trăm của tổng số cam kết) mà dưới ngưỡng đó các hoạt động bất kỳ được nhóm lại thành một danh mục có tên *Những phần khác*.

### 4.9.9.3. Trang cam kết theo ngày



**Hình 4.25. Đồ thị Cam kết-theo-ngày**

Trang này cung cấp cho bạn một trình bày đồ họa của các hoạt động trong dự án tính bằng số các cam kết và tác giả. Điều này một số ý tưởng khi nào dự án đang có người làm việc, và những người đang làm việc là ai tại thời gian nào.

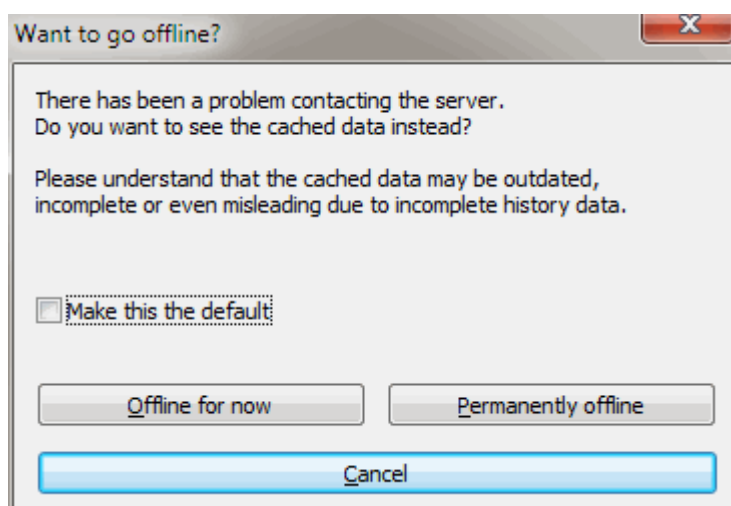
Khi có nhiều tác giả, bạn sẽ nhận được nhiều dòng trên đồ thị. Có hai cách xem có sẵn ở đây:  *bình thường* , nơi mà mỗi tác giả hoạt động liên quan đến đường cơ sở, và  *xếp chồng lên nhau* , nơi hoạt động của mỗi tác giả có liên quan đến đường bên dưới. Tùy chọn thứ hai tránh làm các dòng chéo qua nhau, có thể làm cho đồ thị dễ đọc hơn, nhưng không dễ dàng để thấy công sức bỏ ra của một tác giả.

Theo mặc định, việc phân tích có phân biệt chữ hoa, vì vậy người sử dụng PeterEgan và PeteRegan được coi là các tác giả khác nhau. Tuy nhiên, trong nhiều trường hợp tên người dùng không phân biệt chữ hoa, và đôi khi được nhập vào không thống nhất, vì vậy bạn có thể muốn DavidMorgan và davidmorgan được coi là cùng một người. Sử dụng hộp kiểm **Tác giả phân biệt chữ hoa** để kiểm soát việc này được xử lý như thế nào.

Lưu ý rằng các số liệu thống kê bao gồm các giai đoạn tương tự như hộp thoại Tường Trình. Nếu đó là chi hiển thị một bản sửa đổi thì các số liệu thống kê sẽ không nói lên gì nhiều cho bạn.



## 4.9.10. Chế Độ Ngoại Tuyến



**Hình 4.26. Đến Hộp Thoại Ngoại Tuyến**

Nếu máy chủ không thể truy cập được, và bạn có bộ nhớ đệm cho tường trình được kích hoạt, bạn có thể sử dụng hộp thoại tường trình và biểu đồ sửa đổi trong chế độ ngoại tuyến. Điều này sử dụng dữ liệu từ bộ nhớ cache, cho phép bạn tiếp tục làm việc mặc dù các thông tin có thể không được cập nhật hoặc thậm chí không hoàn chỉnh.

Ở đây bạn có ba lựa chọn:

**Bây giờ thì ngoại tuyến**

Hoàn thành các tác vụ hiện tại trong chế độ ngoại tuyến, nhưng thử lại kho lưu trữ trong lần tiếp theo khi dữ liệu tường trình được yêu cầu.

**Ngoại tuyến thường trực**

Vẫn ở trong chế độ ngoại tuyến cho đến khi kiểm tra kho lưu trữ được yêu cầu cụ thể. Xem [Phần 4.9.11, “Làm mới màn hình xem”](#).

**Hủy bỏ**

Nếu bạn không muốn tiếp tục hoạt động với dữ liệu có thể cũ, chỉ cần hủy bỏ.

**Hộp kiểm Lấy nó làm mặc định** ngăn chặn các hộp thoại này xuất hiện lại và luôn luôn chọn các tùy chọn mà bạn chọn tiếp theo. Bạn vẫn có thể thay đổi (hoặc loại bỏ) mặc định sau khi làm điều này từ **TortoiseSVN** → **Thiết lập**.

## 4.9.11. Làm mới màn hình xem

Nếu bạn muốn kiểm tra máy chủ một lần nữa cho thông điệp tường trình mới, bạn đơn giản là có thể làm mới màn hình xem bằng cách sử dụng **F5**. Nếu bạn đang sử dụng bộ nhớ cache tường trình (kích hoạt mặc định), điều này sẽ kiểm tra các kho lưu trữ cho các thông điệp mới và chỉ lấy những cái mới. Nếu bộ nhớ cache tường trình trong chế độ ngoại tuyến, nó cũng sẽ cố gắng để trở lại trực tuyến.

Nếu bạn đang sử dụng bộ nhớ cache tường trình và bạn nghĩ nội dung thông điệp hay tác giả có thể đã thay đổi, bạn có thể sử dụng **Shift-F5** hoặc **Ctrl-F5** để nạp lại các thông điệp được hiển thị từ máy chủ và cập nhật bộ nhớ cache tường trình. Lưu ý rằng điều này chỉ ảnh hưởng đến các thông điệp hiện đang thể hiện và không làm mất hiệu lực toàn bộ bộ nhớ cache cho kho lưu trữ đó.

## 4.10. Xem Các Khác Biệt

Một trong những yêu cầu phổ biến nhất trong việc phát triển dự án là để xem những gì đã thay đổi. Bạn có thể muốn xem xét sự khác biệt giữa hai phiên bản của cùng một tập tin, hoặc sự khác biệt giữa hai tập tin riêng biệt. TortoiseSVN cung cấp một công cụ được xây dựng sẵn tên TortoiseMerge để xem sự khác biệt của các tập tin văn bản. Để xem sự khác biệt của các tập tin hình ảnh, TortoiseSVN cũng có một công cụ được đặt tên TortoiseIDiff. Tất nhiên, bạn có thể sử dụng chương trình tìm khác biệt yêu thích của bạn nếu muốn.

### 4.10.1. Khác biệt của tập tin

Các thay đổi địa phương

Nếu bạn muốn thấy những thay đổi *bạn* đã thực hiện trong bản sao làm việc của bạn, chỉ cần sử dụng trình đơn ngữ cảnh thám hiểm và chọn **TortoiseSVN** → **Khác** .

Sự khác biệt cho một chi nhánh / thê

Nếu bạn muốn xem những gì đã thay đổi trên thân cây (nếu bạn đang làm việc trên một chi nhánh) hoặc một chi nhánh cụ thể (nếu bạn đang làm việc trên thân cây), bạn có thể sử dụng trình đơn ngữ cảnh thám hiểm. Chỉ cần giữ phím **Shift** trong khi bạn kích chuột phải vào tập tin. Sau đó chọn **TortoiseSVN** → **Tim khác biệt với URL** . Trong hộp thoại tiếp theo, chỉ định các URL trong kho lưu trữ mà bạn muốn so sánh với tập tin địa phương của bạn.

Bạn cũng có thể sử dụng trình duyệt lưu trữ và chọn hai cây khác, có lẽ hai thê, hoặc một chi nhánh / thê và thân. Trình đơn ngữ cảnh ở đó cho phép bạn so sánh chúng bằng cách sử dụng So sánh các bản sửa đổi . Đọc thêm về [Phần 4.10.3, “So sánh các thư mục”](#) .

Sự khác nhau từ một phiên bản trước đó

Nếu bạn muốn thấy sự khác biệt giữa một phiên bản đặc biệt và bản sao làm việc của bạn, sử dụng hộp thoại Tường Trình Sửa Đổi, chọn việc bản sửa đổi mà bạn quan tâm, sau đó chọn So sánh với bản sao làm việc từ trình đơn ngữ cảnh.

Nếu bạn muốn thấy sự khác biệt giữa bản sửa đổi được cam kết cuối cùng và bản sao làm việc của bạn, giả sử rằng các bản sao làm việc đã không được sửa đổi, chỉ cần kích chuột phải vào tập tin. Sau đó chọn **TortoiseSVN** → **Khác biệt với phiên bản trước** . Điều này sẽ thực hiện một việc tìm khác biệt giữa bản sửa đổi trước ngày cam kết cuối cùng (được ghi trong bản sao làm việc của bạn) và bản làm việc CƠ SỞ. Điều này cho bạn thấy sự thay đổi cuối cùng đến tập tin đó mà đem nó đến trạng thái bây giờ bạn nhìn thấy trong bản sao làm việc của bạn. Nó sẽ không hiển thị những thay đổi mới hơn so với bản sao làm việc của bạn.

Sự khác biệt giữa hai phiên bản trước

Nếu bạn muốn thấy sự khác biệt giữa hai phiên bản đã cam kết, sử dụng hộp thoại Tường Trình Sửa Đổi và chọn hai phiên bản bạn muốn so sánh (sử dụng **Ctrl** -Sửa đổi). Sau đó chọn So sánh các phiên bản từ trình đơn ngữ cảnh.

Nếu bạn đã làm điều này từ tường trình sửa đổi cho một thư mục, một hộp thoại So sánh các Sửa đổi xuất hiện, hiển thị một danh sách các tập tin thay đổi trong thư mục đó. Đọc thêm tại [Phần 4.10.3, “So sánh các thư mục”](#) .

Tất cả các thay đổi được thực hiện trong một cam kết

Nếu bạn muốn xem các thay đổi được thực hiện tới tất cả các tập tin trong một phiên bản cụ thể trong một màn xem, bạn có thể sử dụng kết quả ra Diff Thống Nhất (định dạng và lỗi GNU). Điều này chỉ cho thấy sự khác biệt với một vài dòng của bối cảnh. Đó là khó đọc hơn so với việc so sánh tập tin trực quan, nhưng sẽ hiển thị tất cả các thay đổi với nhau. Từ hộp thoại Tường Trình Sửa Đổi lựa chọn bản sửa đổi mà bạn quan tâm, sau đó chọn Hiển thị Khác biệt như Diff Thống Nhất từ trình đơn ngữ cảnh.

Sự khác nhau giữa các tập tin

Nếu bạn muốn thấy sự khác biệt giữa hai tập tin khác nhau, bạn có thể làm điều đó trực tiếp trong trình thám hiểm bằng cách chọn cả hai tập tin (sử dụng **Ctrl** -Sửa đổi). Sau đó, từ trình đơn ngữ cảnh thám hiểm chọn **TortoiseSVN** → **Khác** .

Sự khác biệt giữa tập tin WC / thư mục và một URL

Nếu bạn muốn thấy sự khác biệt giữa một tập tin trong bản sao làm việc của bạn, và một tập tin trong bất kỳ kho lưu trữ Subversion, bạn có thể làm điều đó trực tiếp trong trình thám hiểm bằng cách chọn các tập tin sau đó giữ lại phím **Shift** trong khi cách nhấp chuột phải để có được trình đơn ngữ cảnh. Chọn **TortoiseSVN** → **Tim khác biệt với URL** . Bạn có thể làm điều tương tự cho một bản sao thư mục làm việc. **TortoiseMerge** hiển thị những khác biệt này trong cùng một cách như nó hiển thị một tập tin và - một danh sách các tập tin thay đổi mà bạn có thể xem một tại một thời điểm.

Sự khác biệt với thông tin đồ lỗi

Nếu bạn muốn xem không chỉ sự khác biệt mà còn là tác giả, sửa đổi và ngày thay đổi đã được thực hiện, bạn có thể kết hợp khác và đồ lỗi cho báo cáo từ bên trong hộp thoại tương trình sửa đổi. Đọc [Phần 4.23.2](#), “[Đồ lỗi cho khác biệt](#)” để biết thêm chi tiết.

Sự khác nhau giữa các thư mục

Các công cụ được xây dựng sẵn cung cấp với TortoiseSVN không hỗ trợ xem sự khác biệt giữa các hệ thống phân cấp thư mục. Nhưng nếu bạn có một công cụ bên ngoài mà hỗ trợ tính năng đó, bạn có thể sử dụng thay vào. Trong [Phần 4.10.6](#), “[Công cụ Khác Biệt / Hợp Nhất Bên ngoài](#)” chúng tôi cho bạn biết về một số công cụ mà chúng tôi đã sử dụng.

Nếu bạn đã cấu hình một công cụ bên thứ ba khác, bạn có thể sử dụng **Shift** khi chọn lệnh Diff để sử dụng công cụ thay thế. Đọc [Phần 4.30.5](#), “[Các Thiết Lập Cho Chương Trình Bên Ngoài](#)” để tìm hiểu về cấu hình các công cụ tìm khác biệt khác.

## 4.10.2. Tùy chọn cho Hết-Dòng và Khoảng Trắng

Đôi khi trong cuộc đời của một dự án bạn có thể thay đổi kết thúc dòng `CRLF` sang `LF`, hoặc bạn có thể thay đổi thụt đầu dòng của một phần. Tiếc là điều này sẽ đánh dấu một số lượng lớn các dòng như đã thay đổi, mặc dù không có thay đổi ý nghĩa của mã. Các tùy chọn ở đây sẽ giúp quản lý những thay đổi này khi nói đến việc so sánh và áp dụng sự khác biệt. Bạn sẽ thấy các thiết lập trong hộp thoại **Hợp nhất** và **Đồ lỗi**, cũng như trong các thiết lập cho TortoiseMerge.

Bỏ qua kết thúc dòng không bao gồm thay đổi do sự khác biệt trong phong cách cuối dòng.

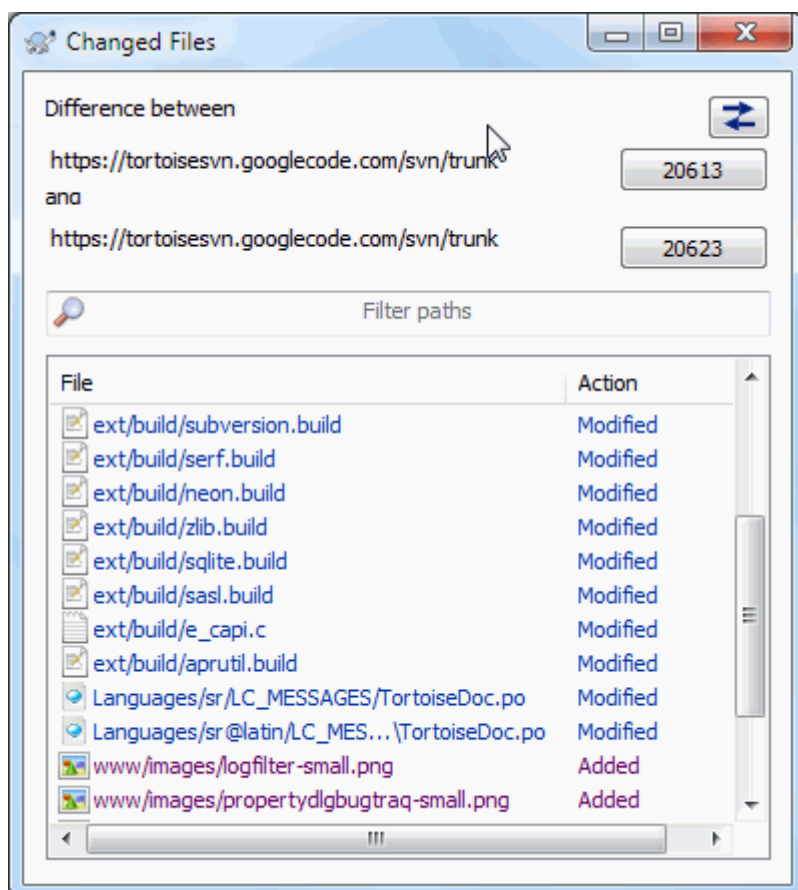
So sánh các khoảng trắng bao gồm tất cả các thay đổi trong thụt đầu dòng và khoảng trắng nội tuyến như các dòng được thêm / gỡ bỏ.

Bỏ qua khoảng trắng thay đổi không bao gồm thay đổi do sự thay đổi trong số lượng hoặc loại khoảng trắng, ví dụ như thay đổi thụt đầu dòng hoặc thay đổi các tab thành khoảng trắng. Thêm khoảng trắng nơi không có trước kia, hoặc loại bỏ một khoảng trắng hoàn toàn vẫn còn được hiển thị như là một thay đổi.

Bỏ qua tất cả các khoảng trắng không bao gồm tất cả thay đổi chỉ có khoảng trắng.

Đương nhiên, bất kỳ dòng nào với nội dung thay đổi luôn luôn bao gồm trong khác biệt.

## 4.10.3. So sánh các thư mục



**Hình 4.27. Hộp Thoại So Sánh Bản Sửa Đổi**

Khi bạn chọn hai cây trong trình duyệt kho lưu trữ, hoặc khi bạn chọn hai bản sửa đổi của một thư mục trong hộp thoại tương trình, bạn có thể **Trình đơn ngữ cảnh** → **So sánh các bản sửa đổi** .

Hộp thoại này cho thấy một danh sách của tất cả các tập tin đã thay đổi và cho phép bạn so sánh hoặc đổ lỗi cho từng tập tin một bằng cách sử dụng trình đơn ngữ cảnh.

Bạn có thể xuất khẩu một *cây thay đổi* , là hữu ích nếu bạn cần phải gửi tới một người khác cấu trúc cây dự án của bạn, nhưng chỉ chứa các tập tin đã thay đổi. Tác vụ này chỉ hoạt động trên các tập tin được lựa chọn, vì vậy bạn cần phải chọn các tập tin quan tâm - thường có nghĩa là tất cả trong số chúng - và sau đó **Trình đơn ngữ cảnh** → **Xuất khẩu lựa chọn ...** . Bạn sẽ được nhắc cho một vị trí để lưu cây thay đổi.

Bạn cũng có thể xuất khẩu *danh sách* các tập tin thay đổi vào một tập tin văn bản bằng cách sử dụng **Trình đơn ngữ cảnh** → **Lưu danh sách các file được chọn ...** .

Nếu bạn muốn xuất khẩu danh sách các tập tin và hành động (sửa đổi, bổ sung, xóa), bạn có thể làm điều đó bằng cách sử dụng **Trình đơn ngữ cảnh** → **Sao chép lựa chọn vào clipboard** .

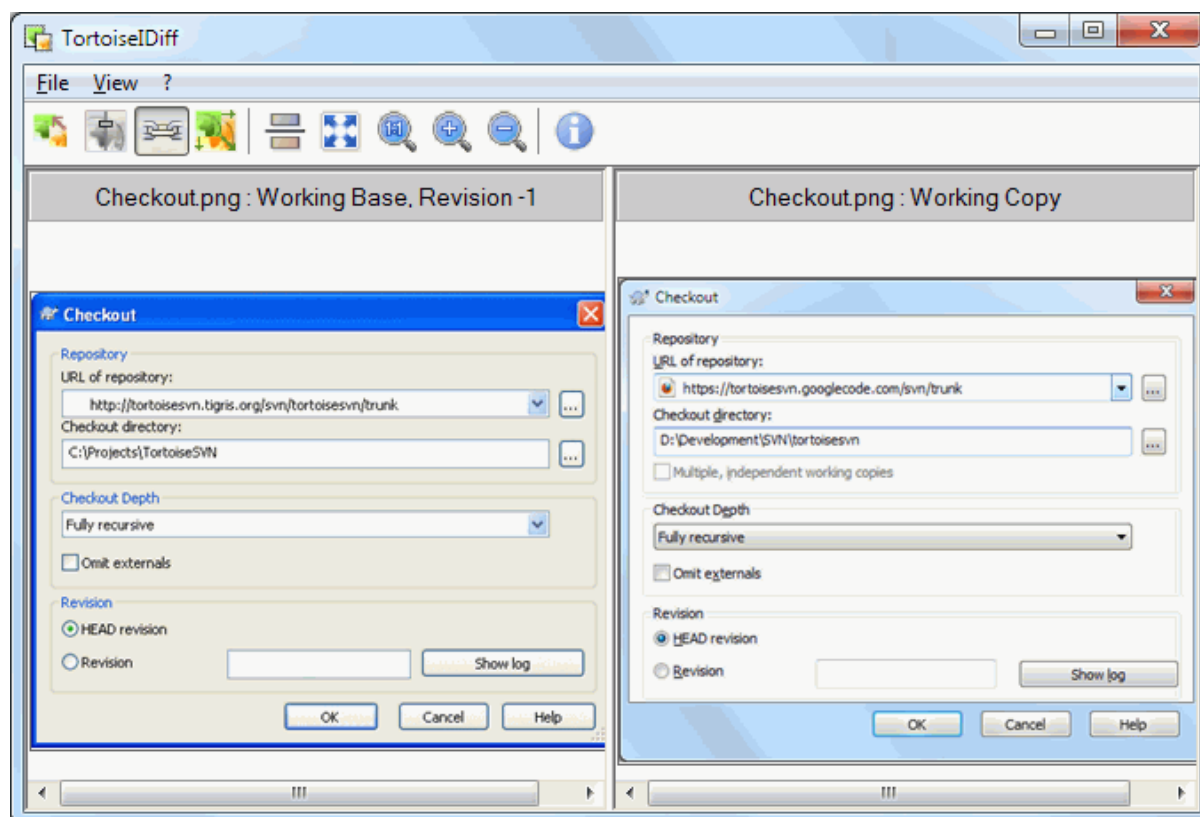
Các nút ở trên cùng cho phép bạn thay đổi hướng so sánh. Bạn có thể hiển thị những thay đổi cần phải nhận được từ A đến B, hoặc nếu bạn thích, từ B đến A.

Các nút với các con số sửa đổi trên có thể được sử dụng để thay đổi một loạt phiên bản khác nhau. Khi bạn thay đổi phạm vi, danh sách các hạng mục khác nhau giữa hai phiên bản sẽ được cập nhật tự động.

Nếu danh sách các tên tập tin là rất dài, bạn có thể sử dụng hộp tìm kiếm để giảm bớt danh sách tên tập tin có chứa văn bản cụ thể. Lưu ý rằng một cách tìm kiếm văn bản đơn giản được sử dụng, vì vậy nếu bạn muốn hạn chế danh sách các tập tin nguồn C bạn nên nhập `.c` chứ không phải là `*.c` .

#### 4.10.4. Tìm Khác Biệt Hình Ảnh Sử Dụng TortoiseIDiff

Có nhiều công cụ có sẵn cho tìm khác biệt tập tin văn bản, bao gồm cả TortoiseMerge của chúng ta, nhưng chúng ta thường thấy mình muốn xem làm thế nào một tập tin hình ảnh đã thay đổi. Đó là lý do tại sao chúng tôi tạo ra TortoiseIDiff.



**Hình 4.28. Trình xem khác biệt hình ảnh**

TortoiseSVN → Khác biệt cho bất kỳ các định dạng tập tin hình ảnh phổ biến sẽ bắt đầu TortoiseIDiff để hiển thị sự khác biệt hình ảnh. Theo mặc định, hình ảnh được hiển thị bên cạnh nhau nhưng bạn có thể sử dụng trình đơn Xem hoặc thanh công cụ để chuyển sang một góc nhìn từ trên xuống dưới, hoặc nếu bạn thích, bạn có thể che phủ những hình ảnh và giả vờ bạn đang sử dụng một hộp sáng.

Tự nhiên bạn cũng có thể phóng to và nhỏ và quay xung quanh hình ảnh. Bạn cũng có thể xoay hình ảnh đơn giản bằng cách kéo trái nó. Nếu bạn chọn tùy chọn **Liên kết hình ảnh với nhau**, sau đó điều khiển xoay (thanh cuộn, con lăn chuột) trên cả hai hình ảnh được liên kết.

Một hộp thông tin hình ảnh hiển thị chi tiết về các tập tin hình ảnh, chẳng hạn như kích thước, độ phân giải điểm ảnh và độ sâu màu. Nếu hộp này được hiện ra, sử dụng **Xem** → **Thông tin về Hình ảnh** để che giấu nó. Bạn có thể nhận được cùng một thông tin trong một tooltip nếu bạn di chuột lên trên thanh tiêu đề hình ảnh.

Khi những hình ảnh được phủ lên, cường độ tương đối của hình ảnh (sự pha trộn alpha) được điều khiển bởi một điều khiển thanh trượt ở phía bên trái. Bạn có thể bấm bất cứ nơi nào trên thanh trượt để thiết lập sự pha trộn trực tiếp, hoặc bạn có thể kéo thanh trượt để thay đổi sự pha trộn một cách tương tác. **Ctrl+Shift** - Bánh xe để thay đổi sự pha trộn.

Các nút trên thanh trượt bập bênh giữa 0% và hỗn hợp 100%, và nếu bạn nhấp đúp chuột vào nút, sự pha trộn các bập bênh tự động mỗi giây cho đến khi bạn nhấn vào nút này một lần nữa. Điều này có thể hữu ích khi tìm kiếm nhiều thay đổi nhỏ.

Đôi khi bạn muốn nhìn thấy một sự khác biệt chứ không phải là một sự pha trộn. Bạn có thể có các tập tin hình ảnh trong hai phiên bản của một bảng mạch in và muốn xem những rãnh nào đã thay đổi. Nếu bạn vô hiệu hóa

chế độ pha trộn alpha, sự khác biệt sẽ được hiển thị như một XOR của các giá trị màu điểm ảnh. Các khu vực không thay đổi sẽ có màu trắng đều và các thay đổi sẽ được tô màu.

#### 4.10.5. Tìm khác biệt các tài liệu văn phòng

Khi bạn muốn tìm khác biệt các tài liệu văn bản bạn thường phải sử dụng phần mềm được sử dụng để tạo ra các tài liệu bởi nó hiểu được định dạng tập tin. Đối với bộ ứng dụng thường được sử dụng như Microsoft Office và Open Office thực sự là có một số hỗ trợ để xem khác biệt và TortoiseSVN bao gồm các kịch bản để gọi những cái này với các thiết lập phù hợp khi bạn tìm khác biệt tập tin với các phần mở rộng tập tin được biết rộng rãi. Bạn có thể kiểm tra những phần mở rộng tập tin nào được hỗ trợ và thêm vào của riêng bạn bằng cách vào TortoiseSVN → Thiết Lập và nhấp Nâng Cao trong phần Chương Trình Bên Ngoài .



#### Các vấn đề với Office 2010

Nếu bạn cài đặt phiên bản *Nhấp Chuột-để-Chạy* của Office 2010 và bạn cố gắng để tìm khác biệt các tài liệu bạn có thể nhận được một thông báo lỗi từ Windows Script Host một cái gì đó như thế này: “ Thành phần ActiveX không thể tạo đối tượng: word.Application ”. Có vẻ như bạn phải sử dụng phiên bản Office dựa trên MSI để có được những chức năng khác.

#### 4.10.6. Công cụ Khác Biệt / Hợp Nhất Bên ngoài

Nếu các công cụ mà chúng tôi cung cấp không làm những gì bạn cần, hãy thử một trong các chương trình mã nguồn mở hoặc thương mại có sẵn. Mọi người đều có chương trình yêu thích của riêng họ, và danh sách này không có nghĩa là hoàn chỉnh, nhưng đây là một vài điều mà bạn có thể xem xét:

##### WinMerge

*WinMerge* [<http://winmerge.sourceforge.net/>] là một công cụ mã nguồn mở khác mà cũng có thể xử lý thư mục.

##### Perforce Merge

Perforce là một RCS thương mại, nhưng bạn có thể tải về các công cụ khác biệt / hợp nhất miễn phí. Nhận thông tin nhiều hơn từ *Perforce* [<http://www.perforce.com/perforce/products/merge.html>] .

##### KDiff3

KDiff3 là một công cụ tìm khác biệt miễn phí mà cũng có thể xử lý thư mục. Bạn có thể tải xuống từ *ở đây* [<http://kdiff3.sf.net/>] .

##### SourceGear DiffMerge

SourceGear Vault là một RCS thương mại, nhưng bạn có thể tải về công cụ tìm khác biệt / sáp nhập miễn phí. Nhận thông tin nhiều hơn từ *SourceGear* [<http://www.sourcegear.com/diffmerge/>].

##### ExamDiff

Tiêu Chuẩn ExamDiff là phần mềm miễn phí. Nó có thể xử lý các tập tin mà không phải thư mục. ExamDiff Pro là phần mềm chia sẻ và cho biết thêm một số tính năng tốt bao gồm tìm khác biệt thư mục và khả năng biên tập. Trong cả hai hướng vị, phiên bản 3.2 và ở trên có thể xử lý unicode. Bạn có thể tải về từ *PrestoSoft* [<http://www.prestosoft.com/>] .

##### Bên Ngoài So Sánh

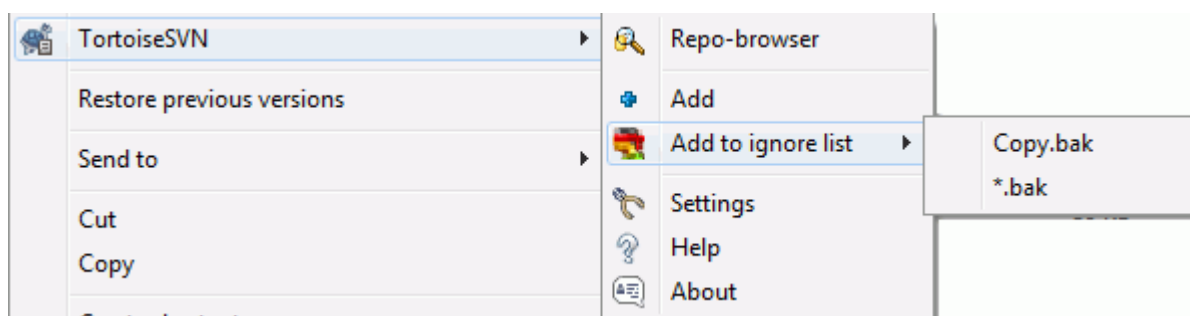
Tương tự như ExamDiff Pro, đây là một công cụ chia sẻ tìm khác biệt tuyệt vời khác mà có thể xử lý khác biệt thư mục và unicode. Tải xuống từ *Phần mềm Scooter* [<http://www.scootersoftware.com/>] .

##### Araxis Merge

Araxis Merge là một công cụ thương mại hữu ích cho khác biệt và sáp nhập cả hai tập tin và thư mục. Nó so sánh ba chiều trong việc hợp nhất và có liên kết đồng bộ hóa để sử dụng nếu bạn đã thay đổi thứ tự của các chức năng. Tải xuống từ *Araxis* [<http://www.araxis.com/merge/index.html>] .

Đọc [Phần 4.30.5, “Các Thiết Lập Cho Chương Trình Bên Ngoài ”](#) để lấy thông tin về làm thế nào để cài đặt TortoiseSVN để sử dụng những công cụ này.

## 4.11. Thêm Tập Tin Và Thư Mục Mới



**Hình 4.29. Trình đơn ngữ cảnh Explorer cho các tập tin không phiên bản**

Nếu bạn đã tạo tập tin mới và / hoặc thư mục trong quá trình phát triển của bạn thì bạn cần phải thêm chúng vào kiểm soát nguồn. Chọn các tập tin và / hoặc thư mục và sử dụng TortoiseSVN → Thêm .

Sau khi bạn đã thêm các tập tin / thư mục vào kiểm soát nguồn tập tin xuất hiện với một lớp phủ biểu tượng thêm mà có nghĩa là bạn phải cam kết bản sao của bạn làm việc để làm cho những tập tin / thư mục sẵn sàng cho các nhà phát triển khác. Thêm một tập tin / thư mục *không* ảnh hưởng đến kho lưu trữ!



### Nhiều Việc Thêm

Bạn cũng có thể sử dụng lệnh Thêm trên các thư mục đã được phiên bản. Trong trường hợp đó, hộp thoại thêm sẽ cho bạn thấy tất cả các file không phiên bản bên trong thư mục phiên bản đó. Điều này giúp nếu bạn có nhiều tập tin mới và cần thêm tất cả cùng một lúc.

Để thêm các tập tin từ bên ngoài bản sao làm việc của bạn, bạn có thể sử dụng xử lý kéo và thả:

1. lựa chọn các tập tin mà bạn muốn thêm vào
2. kéo phải chúng đến vị trí mới bên trong các bản sao làm việc
3. thả nút chuột phải
4. chọn Trình Đơn Ngữ Cảnh → SVN Thêm các tập tin vào WC này . Các tập tin sau đó sẽ được sao chép vào các bản sao làm việc và thêm vào kiểm soát phiên bản.

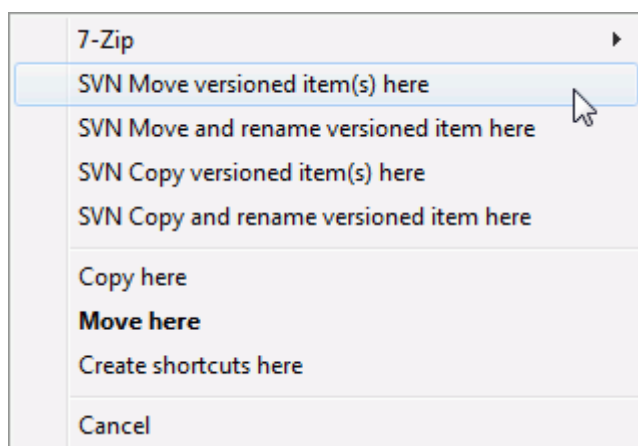
Bạn cũng có thể thêm các tập tin trong một bản sao làm việc chỉ đơn giản bằng trái cách kéo và thả chúng vào hộp thoại cam kết.

Nếu bạn thêm một tập tin hoặc thư mục do nhầm lẫn, bạn có thể hoàn tác việc thêm trước khi bạn cam kết sử dụng TortoiseSVN → Hoàn tác việc thêm ... .

## 4.12. Sao chép / di chuyển / Đổi tên tập tin và thư mục

Thường hay xảy ra rằng bạn đã có các tập tin bạn cần trong một dự án khác trong kho của bạn, và bạn chỉ đơn giản là muốn sao chép chúng qua. Thật đơn giản, bạn chỉ cần có thể sao chép các tập tin và thêm chúng như mô tả ở trên, nhưng điều đó sẽ không cung cấp cho bạn bất kỳ lịch sử nào. Và nếu bạn sau đó sửa chữa một lỗi trong các tập tin ban đầu, bạn chỉ có thể kết hợp sửa chữa tự động nếu các bản sao mới có liên quan với bản gốc trong Subversion.

Cách dễ nhất để sao chép các tập tin và thư mục từ trong một bản sao làm việc là để sử dụng trình đơn kéo phải. Khi bạn phải kéo một tập tin hoặc thư mục từ một bản sao làm việc khác, hoặc thậm chí trong cùng một thư mục, một trình đơn ngữ cảnh xuất hiện khi bạn thả chuột.



**Hình 4.30. Kéo phải trình đơn cho một thư mục dưới sự kiểm soát phiên bản**

Bây giờ bạn có thể sao chép nội dung là phiên bản hiện tại sang một vị trí mới, có thể đổi tên nó cùng một lúc.

Bạn cũng có thể sao chép hoặc di chuyển các tập tin là phiên bản trong một bản sao làm việc, hoặc giữa hai bản sao làm việc, sử dụng phương pháp cắt-và-dán quen thuộc. Sử dụng Sao chép hoặc Cắt tiêu chuẩn của Windows để sao chép một hoặc nhiều mục là phiên bản vào clipboard. Nếu clipboard chứa các đồ vật được phiên bản như vậy, sau đó bạn có thể sử dụng TortoiseSVN → Dán (Lưu ý: không phải Dán tiêu chuẩn của Windows) để sao chép hoặc di chuyển các mục vào vị trí bản sao làm việc mới.

Bạn có thể sao chép các tập tin và thư mục từ bản sao làm việc của bạn đến một vị trí khác trong kho lưu trữ bằng cách sử dụng TortoiseSVN → Nhánh / Thẻ . Tham khảo [Phần 4.19.1, “Tạo một Chi nhánh hoặc Thẻ”](#) để tìm hiểu thêm.

Bạn có thể xác định vị trí một phiên bản cũ của một tập tin hoặc thư mục trong hộp thoại tương trình và sao chép nó vào một vị trí mới trong kho lưu trữ trực tiếp từ hộp thoại tương trình sử dụng Trình đơn ngữ cảnh → Tạo chi nhánh / thẻ từ sửa đổi . Tham khảo [Phần 4.9.3, “Lấy thông tin thêm”](#) để tìm hiểu thêm.

Bạn cũng có thể sử dụng trình duyệt kho lưu trữ để xác định vị trí nội dung mà bạn muốn, và sao chép nó vào bản sao làm việc của bạn trực tiếp từ kho lưu trữ, hoặc sao chép giữa hai vị trí trong kho. Tham khảo [Phần 4.24, “Trình duyệt Kho”](#) để tìm hiểu thêm.



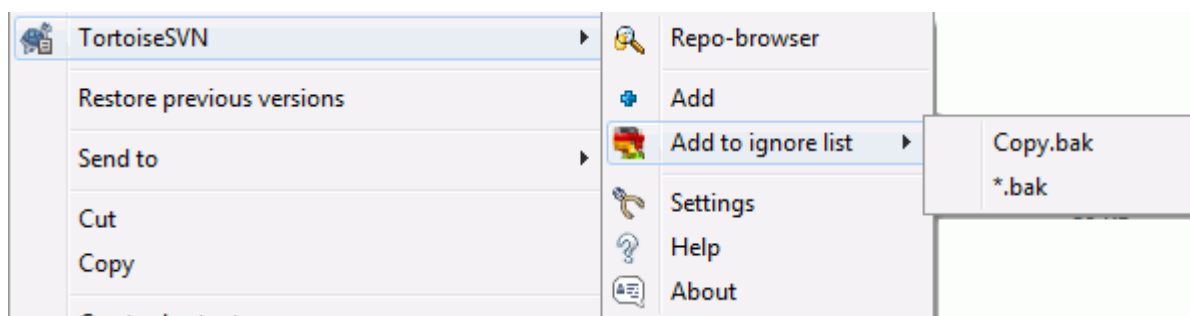
### Không thể sao chép giữa các kho

Trong khi bạn có thể sao chép hoặc di chuyển các tập tin và thư mục *trong* một kho lưu trữ, bạn *không thể* sao chép hoặc di chuyển từ một kho lưu trữ khác trong khi vẫn giữ lịch sử bằng cách sử dụng TortoiseSVN. Không, ngay cả khi các kho lưu trữ tồn tại trên cùng một máy chủ. Tất cả những gì bạn có thể làm là sao chép nội dung trong trạng thái hiện tại của nó và thêm nó như là nội dung mới vào kho lưu trữ thứ hai.

Nếu bạn không chắc chắn là hai URL trên cùng một máy chủ chỉ đến các kho giống nhau hay khác nhau, sử dụng trình duyệt kho để mở một URL và tìm ra gốc của kho. Nếu bạn có thể nhìn thấy cả hai địa điểm trong một cửa sổ trình duyệt kho thì chúng ở trong cùng một kho lưu trữ.

## 4.13. Bỏ qua các Tập Tin và Thư mục





**Hình 4.31. Trình đơn ngữ cảnh Explorer cho các tập tin không phiên bản**

Trong hầu hết các dự án bạn sẽ có các tập tin và thư mục mà không nên là đối tượng để kiểm soát phiên bản. Đây có thể bao gồm các tập tin được tạo ra bởi trình biên dịch, \*.obj, \*.lst, có thể là một thư mục đầu ra được sử dụng để lưu trữ thực thi. Bất cứ khi nào bạn cam kết thay đổi, TortoiseSVN cho thấy các tập tin không phiên bản của bạn, đẩy lên danh sách tập tin trong hộp thoại cam kết. Tất nhiên bạn có thể tắt hiển thị này, nhưng sau đó bạn có thể quên để thêm một tập tin nguồn mới.

Cách tốt nhất để tránh những vấn đề này là thêm các tập tin có nguồn gốc từ danh sách bỏ qua của dự án. Bằng cách đó chúng sẽ không bao giờ xuất hiện trong hộp thoại cam kết, nhưng các tập tin nguồn chính gốc chưa phiên bản vẫn sẽ được gắn cờ lên.

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.

Nếu bạn muốn loại bỏ một hoặc nhiều mục từ danh sách bỏ qua, nhấp chuột phải trên các mục và chọn TortoiseSVN → Gỡ bỏ khỏi danh sách bỏ qua. Bạn cũng có thể truy cập vào thuộc tính một thư mục của svn:ignore một cách trực tiếp. Điều đó cho phép bạn chỉ định mô hình tổng quát hơn bằng cách sử dụng toàn cục hóa tên tập tin, được mô tả trong phần dưới đây. Đọc [Phần 4.17, “Thiết lập Dự án”](#) để biết thêm thông tin về việc thiết lập các thuộc tính trực tiếp. Xin hãy nhận thức rằng mỗi mô hình bỏ qua phải được đặt trên một dòng riêng biệt. Tách chúng bằng dấu cách sẽ không có tác dụng.



### Các Danh sách bỏ qua toàn cục

Một cách khác để bỏ qua các tập tin là thêm chúng vào *danh sách bỏ qua toàn cục*. Sự khác biệt lớn ở đây là các danh sách bỏ qua toàn cục là một thuộc tính của máy khách. Nó áp dụng cho *tất cả* các dự án Subversion, nhưng chỉ trên máy khách. Nói chung, tốt hơn sử dụng thuộc tính svn:ignore trong trường hợp có thể, bởi vì nó có thể được áp dụng cho các khu vực dự án cụ thể, và nó hoạt động cho tất cả mọi người kiểm tra ra dự án. Đọc [Phần 4.30.1, “Các Thiết Lập Tổng Qu”](#) để biết thêm thông tin.



### Bỏ qua các mục được phiên bản

Các tập tin và thư mục được phiên bản không bao giờ có thể được bỏ qua - đó là một tính năng của Subversion. Nếu bạn phiên bản một tập tin một cách vô ý, đọc [Phần B.8, “Bỏ qua các tập tin được đã được phiên bản”](#) để được hướng dẫn làm thế nào để “bỏ phiên bản” nó.

#### 4.13.1. Việc Khớp các Mẫu trong Danh sách Bỏ qua

Các mẫu bỏ qua của Subversion sử dụng cách toàn cục hóa tên tập tin, một kỹ thuật ban đầu được sử dụng trong Unix để xác định các tập tin bằng cách sử dụng ký tự meta như là ký tự đại diện. Các ký tự sau đây có ý nghĩa đặc biệt:

\*

Khớp bất kỳ một chuỗi ký tự, bao gồm cả chuỗi rỗng (không có ký tự).

?

Khớp bất kỳ ký tự đơn.

[...]

Khớp các ký tự bất kỳ trong dấu ngoặc vuông. Trong dấu ngoặc, một cặp ký tự cách nhau bằng “ - ” khớp bất kỳ ký tự nào theo cách từ vựng giữa hai người. Ví dụ `[AGM-p]` khớp với bất kỳ một trong những `A` , `G` , `m` , `n` , `o` hoặc `p` .

Khớp mẫu có phân biệt chữ hoa, có thể gây ra vấn đề trên Windows. Bạn có thể bắt buộc không phân biệt chữ hoa theo cách khó khăn bằng cách ghép đôi ký tự, ví dụ như bỏ qua `*.tmp` bất kể chữ hoa, bạn có thể sử dụng một mẫu như `*.[Tt][Mm][Pp]` .

Nếu bạn muốn có một định nghĩa chính thức cho globbing, bạn có thể tìm thấy nó trong các chi tiết kỹ thuật IEEE cho ngôn ngữ lệnh shell *Khớp các mẫu ký hiệu* [[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)] .

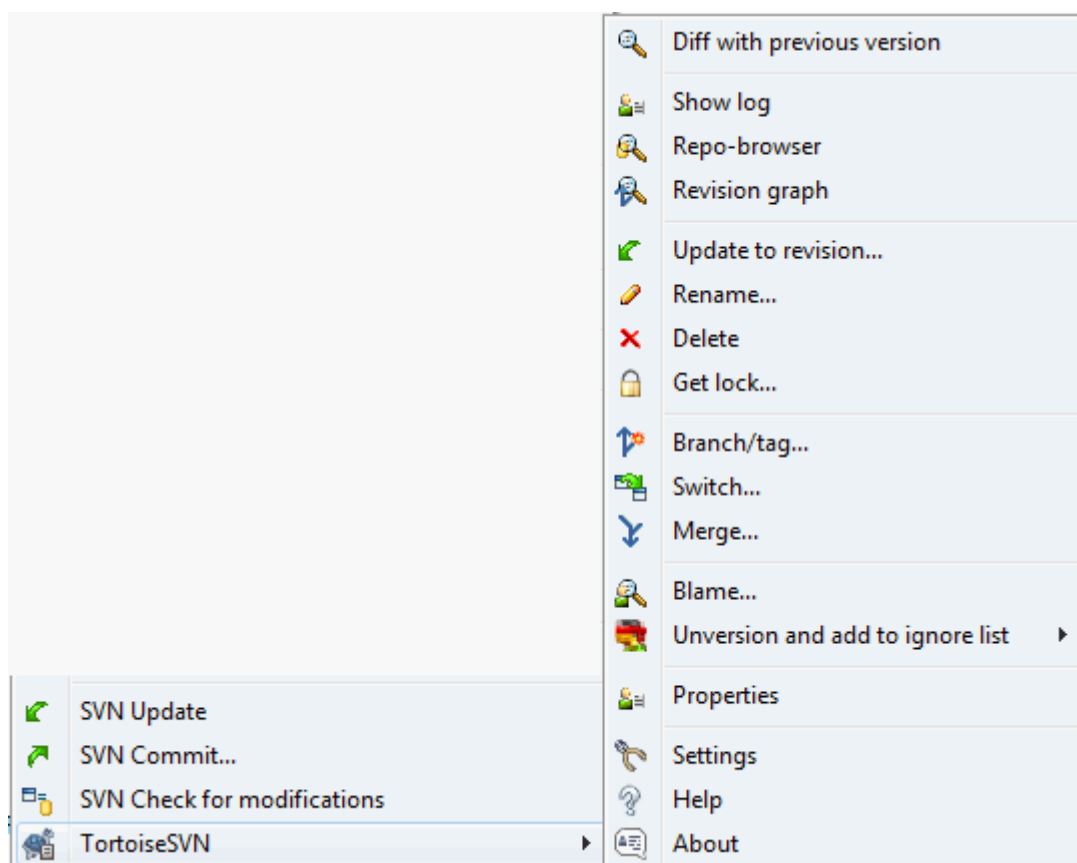


### Không có đường dẫn nào trong Danh sách bỏ qua toàn cục

Bạn không nên bao gồm thông tin đường dẫn trong mẫu của bạn. Cách khớp mẫu có thể dùng để sử dụng trên các tên tập tin suông và tên thư mục. Nếu bạn muốn bỏ qua tất cả thư mục `CVS` , chỉ cần thêm `CVS` vào danh sách bỏ qua. Không cần phải chỉ định `CVS * / CVS` như bạn đã làm trong các phiên bản trước đó. Nếu bạn muốn bỏ qua tất cả thư mục `tmp` khi chúng tồn tại trong một thư mục `prog` nhưng không phải trong một thư mục `doc` bạn nên sử dụng thuộc tính `svn:ignore` thay vào đó. Không có cách nào đáng tin cậy để đạt được điều này bằng cách sử dụng các mẫu bỏ qua toàn cục.

## 4.14. Xóa, Di chuyển và đổi tên

Subversion cho phép đổi tên và di chuyển các tập tin và thư mục. Vì vậy, có các mục trình đơn để xóa và đổi tên trong trình đơn phụ TortoiseSVN.



Hình 4.32. Trình đơn ngữ cảnh của Explorer cho các tập tin được phiên bản

#### 4.14.1. Xóa các tập tin và thư mục

Sử dụng TortoiseSVN → Xóa để loại bỏ các tập tin và thư mục khỏi Subversion.

Khi bạn TortoiseSVN → Xóa một tập tin hoặc thư mục, nó được loại bỏ khỏi bản làm việc của bạn ngay lập tức cũng như được đánh dấu xóa khỏi kho trong lần cam kết tiếp theo. Thư mục mẹ của mục đó hiển thị một biểu tượng lớp phủ “được sửa đổi”. Cho tới khi bạn cam kết các thay đổi, bạn có thể lấy lại tập tin bằng cách dùng TortoiseSVN → Hoàn Nguyên trên thư mục mẹ.

Nếu bạn muốn xóa một mục khỏi kho lưu trữ, nhưng giữ nó tại địa phương như là một tập tin / thư mục không được phiên bản, sử dụng Trình đơn ngữ cảnh mở rộng → Xóa (giữ lại địa phương) . Bạn phải giữ phím **Shift** trong khi nhấp chuột phải vào mục trong cửa sổ danh sách thám hiểm (cửa sổ bên phải) để thấy điều này trong trình đơn ngữ cảnh mở rộng.

Nếu một mục được xóa thông qua explorer thay vì dùng trình đơn ngữ cảnh TortoiseSVN, hộp thoại cam kết hiển thị các mục này như đã mất và để bạn loại bỏ chúng khỏi kiểm soát phiên bản trước khi cam kết. Tuy nhiên, nếu bạn cập nhật bản làm việc, Subversion sẽ phát hiện mục bị mất và thay thế nó với phiên bản mới nhất từ kho.

Nếu bạn cần xóa một tập tin được đánh phiên bản, luôn luôn sử dụng TortoiseSVN → Xóa để Subversion không phải đoán điều bạn thật sự muốn làm.



#### Lấy trở lại một tập tin đã xóa hoặc thư mục

Nếu bạn đã xóa một tập tin hoặc một thư mục và đã cam kết tác vụ xóa vào kho, sau đó là một TortoiseSVN → Hoàn nguyên bình thường không có thể mang lại cho nó trở lại nữa. Tuy nhiên, tập tin hoặc thư mục không phải là mất tất cả. Nếu bạn biết việc sửa đổi các tập tin hoặc thư mục đã bị xóa (nếu bạn không sử dụng hộp thoại đăng nhập để tìm hiểu) mở trình duyệt lưu trữ và chuyển

sang sửa đổi đó. Sau đó chọn tập tin hoặc thư mục mà bạn đã xóa, nhấp chuột phải và chọn Trình đơn ngữ cảnh → Sao chép vào ... như là mục tiêu cho hoạt động sao chép, chọn đường dẫn đến bản sao làm việc của bạn.

#### 4.14.2. Di chuyển tập tin và thư mục

Nếu bạn muốn làm một đổi tên tại chỗ đơn giản của một tập tin hoặc thư mục, sử dụng Trình đơn ngữ cảnh → Đổi tên ... Nhập tên mới cho mục đó và bạn xem như xong.

Nếu bạn muốn di chuyển các tập tin vòng vòng bên trong bản sao làm việc của bạn, có lẽ đến một thư mục con khác, sử dụng xử lý chuột phải kéo-và-thả:

1. lựa chọn các tập tin hoặc thư mục mà bạn muốn di chuyển
2. kéo phải chúng đến vị trí mới bên trong các bản sao làm việc
3. thả nút chuột phải
4. trong trình đơn bật lên chọn Trình Đơn Ngữ Cảnh → SVN Di chuyển các tập tin được phiên bản tới đây



#### Cam kết các thư mục mẹ

Kể từ khi việc đổi tên và di chuyển được thực hiện như là một lệnh xóa theo sau là một lệnh thêm bạn phải cam kết các thư mục mẹ của tập tin đổi tên / di chuyển để cho phần bị xóa của việc đổi tên / di chuyển sẽ hiển thị trong hộp thoại cam kết. Nếu bạn không cam kết phần bị loại bỏ của việc đổi tên / di chuyển, nó sẽ ở lại trong kho và khi đồng nghiệp của bạn cập nhật, tập tin cũ sẽ không được gỡ bỏ, tức là họ sẽ có *cả hai* bản sao cũ và mới.

Bạn *phải* cam kết đổi tên thư mục trước khi thay đổi bất kỳ tập tin bên trong thư mục, nếu không bản sao làm việc của bạn có thể bị rối tung.

Một cách khác để di chuyển hoặc sao chép các tập tin là sử dụng lệnh Windows sao chép / cắt. Chọn các tập tin mà bạn muốn sao chép, nhấp chuột phải và chọn Trình Đơn Ngữ Cảnh → Sao chép từ trình đơn ngữ cảnh thám hiểm. Sau đó duyệt đến thư mục đích, nhấp chuột phải và chọn TortoiseSVN → Dán . Đối với các tập tin di chuyển, chọn Trình Đơn Ngữ Cảnh → Cắt thay vì Trình Đơn Ngữ Cảnh → Sao chép .

Bạn cũng có thể sử dụng trình duyệt kho để di chuyển các mục vòng vòng. Đọc [Phần 4.24, “Trình duyệt Kho”](#) để tìm hiểu thêm.



#### Không SVN Di chuyển các thành phần ngoài

Bạn *không* nên sử dụng các lệnh TortoiseSVN Di chuyển hoặc Đổi tên trên một thư mục đã được tạo ra bằng cách sử dụng `svn:externals` . Hành động này sẽ làm các mục bên ngoài bị xóa từ kho cha mẹ của nó, có thể khó chịu nhiều người khác. Nếu bạn cần phải di chuyển các thành phần bên ngoài thư mục bạn nên sử dụng một di chuyển shell bình thường, sau đó điều chỉnh thuộc tính `svn:externals` của thư mục cha mẹ nguồn và đích.

#### 4.14.3. Đối phó với các cuộc xung đột kiểu chữ tên tập tin

Nếu kho lưu trữ đã có hai tập tin có cùng tên nhưng khác nhau chỉ trong kiểu chữ (ví dụ: `Test.txt` và `test.txt` ), Bạn sẽ không thể cập nhật hoặc kiểm tra các thư mục cha trên một máy khách Windows. Trong khi Subversion hỗ trợ các trường hợp nhạy cảm kiểu chữ tên tập tin, Windows thì lại không.

Điều này đôi khi sẽ xảy ra khi hai người cam kết, từ các bản sao làm việc riêng biệt, các tập tin ngẫu nhiên có cùng tên, nhưng với khác biệt về kiểu chữ. Nó cũng có thể xảy ra khi tập tin được cam kết từ một hệ thống với một hệ thống tập tin phân biệt kiểu chữ hoa, như Linux.

Trong trường hợp đó, bạn phải quyết định một trong số chúng bạn muốn giữ và xóa (hoặc đổi tên) cái kia từ kho lưu trữ.



### Ngăn chặn hai tập tin có cùng tên

Có một kịch bản móc máy chủ có tại: <http://svn.collab.net/repos/svn/trunk/contrib/hook-scripts/> [<http://svn.collab.net/repos/svn/trunk/contrib/hook-scripts/>] mà sẽ ngăn chặn kiểm tra vào mà gây ra xung đột kiểu chữ.

#### 4.14.4. Sửa chữa đổi tên tập tin

Đôi khi IDE thân thiện của bạn sẽ đổi tên tập tin cho bạn như là một phần của bài tập cấu trúc lại, và tất nhiên nó không báo cho Subversion. Nếu bạn cố gắng cam kết thay đổi của bạn, Subversion sẽ xem tên tập tin cũ như đã bị mất tích và tên mới như là một tập tin không phiên bản. Bạn chỉ có thể kiểm tra tên tập tin mới để có được nó được thêm vào, nhưng sau đó bạn sẽ mất truy tìm lịch sử, bởi Subversion không biết các tập tin có liên quan.

Một cách tốt hơn là để thông báo cho Subversion rằng sự thay đổi này thực sự là một đổi tên, và bạn có thể làm điều này trong hộp thoại **Cam kết** và **Kiểm tra cho sửa đổi**. Đơn giản chỉ cần chọn cả hai tên (mất tích) và tên mới (chưa phiên bản) và sử dụng **Trình Đơn Ngữ Cảnh** → **Sửa chữa Di chuyển** để ghép hai tập tin như đổi tên một.

#### 4.14.5. Xóa các tập tin chưa phiên bản

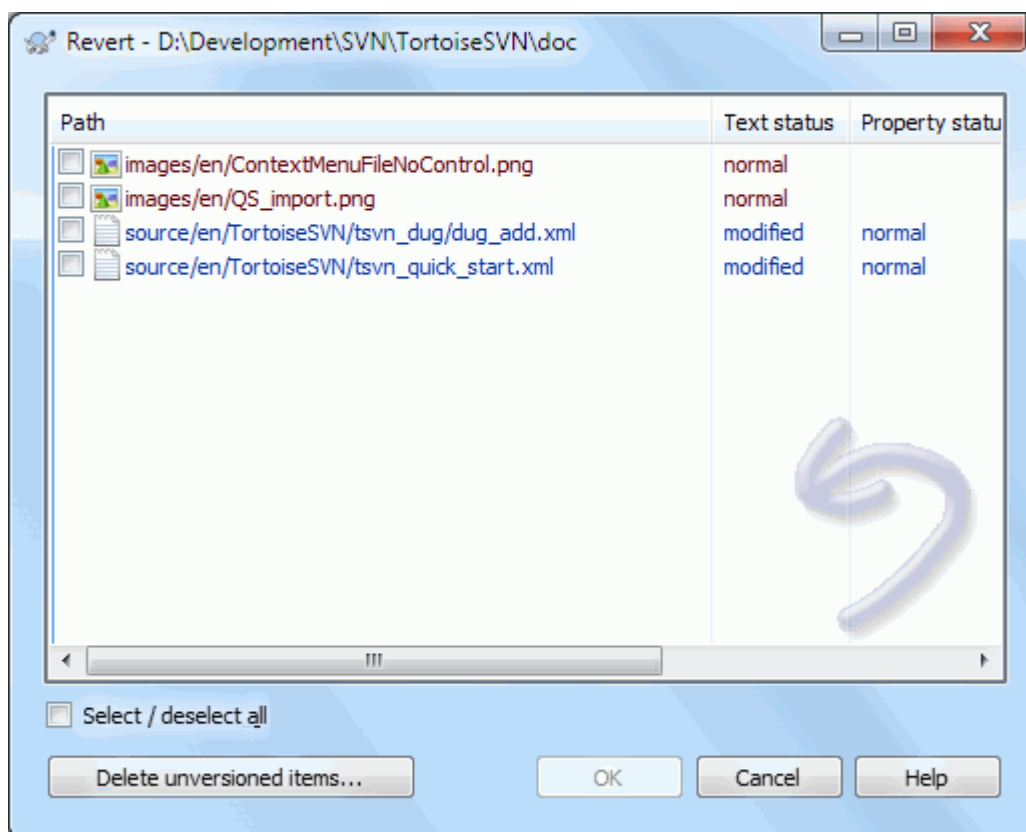
Thông thường bạn đặt danh sách bỏ qua của bạn theo cách mà tất cả các tập tin được tạo ra được bỏ qua trong Subversion. Nhưng điều gì sẽ xảy ra nếu bạn muốn để xóa tất cả những mục được bỏ qua để sản xuất một bản xây dựng sạch? Thông thường, bạn sẽ thiết lập rằng trong makefile của bạn, nhưng nếu bạn gỡ rối các makefile, hoặc thay đổi hệ thống xây dựng nó rất hữu ích để có một cách để dọn dẹp các sản.

TortoiseSVN chỉ cung cấp một tùy chọn sử dụng **Trình Đơn Ngữ Cảnh mở rộng** → **Xóa mục chưa phiên bản ...**. Bạn phải giữ **Shift** trong khi nhấp chuột phải vào một thư mục trong cửa sổ danh sách thám hiểm (cửa sổ bên phải) để thấy điều này trong trình đơn ngữ cảnh mở rộng. Điều này sẽ tạo ra một hộp thoại trong đó liệt kê tất cả các tập tin chưa phiên bản bất cứ nơi nào trong bản sao làm việc của bạn. Sau đó, bạn có thể chọn hoặc bỏ chọn các mục được gỡ bỏ.

Khi các hạng mục bị xóa, thùng rác được sử dụng, vì vậy nếu bạn phạm sai lầm ở đây và xóa một tập tin mà cần phải có được phiên bản, bạn vẫn có thể phục hồi nó.

#### 4.15. Hoàn tác các thay đổi

Nếu bạn muốn phục hồi lại tất cả thay đổi bạn đã thực hiện trong một tập tin kể từ khi cập nhật cuối cùng bạn cần phải chọn tập tin, nhấp chuột phải để bật lên trình đơn ngữ cảnh và sau đó chọn lệnh **TortoiseSVN** → **Hoàn nguyên**. Một hộp thoại sẽ bật lên hiển thị cho bạn các tập tin mà bạn đã thay đổi và có thể trở lại. Chọn những cái muốn hoàn nguyên và bấm vào **OK**.



**Hình 4.33. Hộp thoại hoàn nguyên**

Nếu bạn muốn phục hồi lại một xóa hoặc đổi tên, bạn cần để sử dụng Hoàn nguyên trên thư mục mẹ bởi vì mục bị xóa sẽ không tồn tại để bạn có thể nhấp chuột phải vào.

Nếu bạn muốn hoàn tác việc bổ sung một mục, điều này xuất hiện trong trình đơn ngữ cảnh TortoiseSVN → Hoàn tác việc thêm vào ... . Đây thực sự là một hoàn nguyên, nhưng tên đã được thay đổi để làm cho nó rõ ràng hơn.

Các cột trong hộp thoại này có thể được tùy chỉnh trong cùng một cách như các cột trong hộp thoại Kiểm tra để sửa đổi. Đọc [Phần 4.7.4, “Trạng thái địa phương và từ xa”](#) để biết thêm chi tiết.

Bởi vì hoàn nguyên đôi khi được sử dụng để làm sạch một bản sao làm việc, có thêm một nút cho phép bạn để xóa các mục chưa phiên bản. Khi bạn nhấp vào nút này hộp thoại khác xuất hiện liệt kê tất cả các mục chưa phiên bản, mà sau đó bạn có thể chọn để xóa.



### Hoàn tác các thay đổi đã được cam kết

Hoàn nguyên sẽ chỉ hoàn tác các thay đổi địa phương của bạn. Nó *không* hoàn tác bất kỳ thay đổi nào đã được cam kết. Nếu bạn muốn phục hồi lại tất cả các thay đổi đã được cam kết trong một phiên bản nhất định, đọc [Phần 4.9, “Hộp thoại Trình chỉnh sửa”](#) để biết thêm thông tin.



### Hoàn nguyên là chậm

Khi bạn hoàn nguyên các thay đổi bạn có thể thấy rằng hoạt động mất rất nhiều thời gian hơn bạn mong đợi. Điều này là do các phiên bản sửa đổi của tập tin được gửi vào thùng rác, vì vậy bạn có thể lấy những thay đổi của bạn nếu bạn hoàn nguyên do nhầm lẫn. Tuy nhiên, nếu thùng rác của bạn bị đầy, Windows sẽ mất một thời gian dài để tìm một nơi để đặt các tập tin. Giải pháp là đơn giản: làm trống thùng rác hoặc tắt hộp Sử dụng thùng rác khi hoàn nguyên trong thiết lập TortoiseSVN.

## 4.16. Dọn sạch

Nếu một lệnh Subversion không có thể hoàn thành thành công, có lẽ do vấn đề máy chủ, bản sao làm việc của bạn có thể bị bỏ lại trong một tình trạng không nhất quán. Trong trường hợp đó bạn cần phải sử dụng TortoiseSVN → Dọn sạch thư mục. Đó là một ý tưởng tốt để làm điều này ở cấp cao nhất của bản sao làm việc.

Trong hộp thoại dọn dẹp, cũng có tùy chọn hữu ích khác để lấy bản sao làm việc vào một trạng thái `sạch`.

Làm sạch tình trạng bản sao làm việc

Như đã nêu ở trên, tùy chọn này cố gắng để có được một bản sao làm việc không nhất quán vào một trạng thái khả thi và có thể sử dụng. Điều này không ảnh hưởng đến bất kỳ dữ liệu nào mà bạn có nhưng chỉ trạng thái nội bộ của cơ sở dữ liệu bản sao làm việc. Điều này thực tế là lệnh `Dọn dẹp` mà bạn biết từ các trình khách TortoiseSVN cũ hoặc các trình khách SVN khác.

Làm mới các lớp phủ shell

Đôi khi các lớp phủ shell, đặc biệt là trên màn hình cây ở phía bên trái của explorer không hiển thị trạng thái hiện tại, hoặc trạng thái được lưu trữ tạm không nhận ra những thay đổi. Trong trường hợp này, bạn có thể sử dụng lệnh này để ép buộc một hành động làm mới.

Bao gồm các thành phần ngoài

Nếu điều này được kiểm tra, thì tất cả các hành động cũng được thực hiện cho tất cả các tập tin và thư mục với thuộc tính `svn:externals`.

Xóa các tập tin và thư mục không phiên bản, Xóa các tập tin và thư mục bị bỏ qua

Đây là một cách nhanh chóng và dễ dàng để loại bỏ tất cả các tập tin được tạo ra trong bản sao làm việc của bạn. Tất cả các tập tin và thư mục mà không có phiên bản được chuyển vào thùng rác.

Lưu ý: bạn cũng có thể làm tương tự từ hộp thoại TortoiseSVN → Hoàn nguyên. Ở đó bạn cũng có được một danh sách của tất cả các tập tin không phiên bản và thư mục để lựa chọn để loại bỏ.

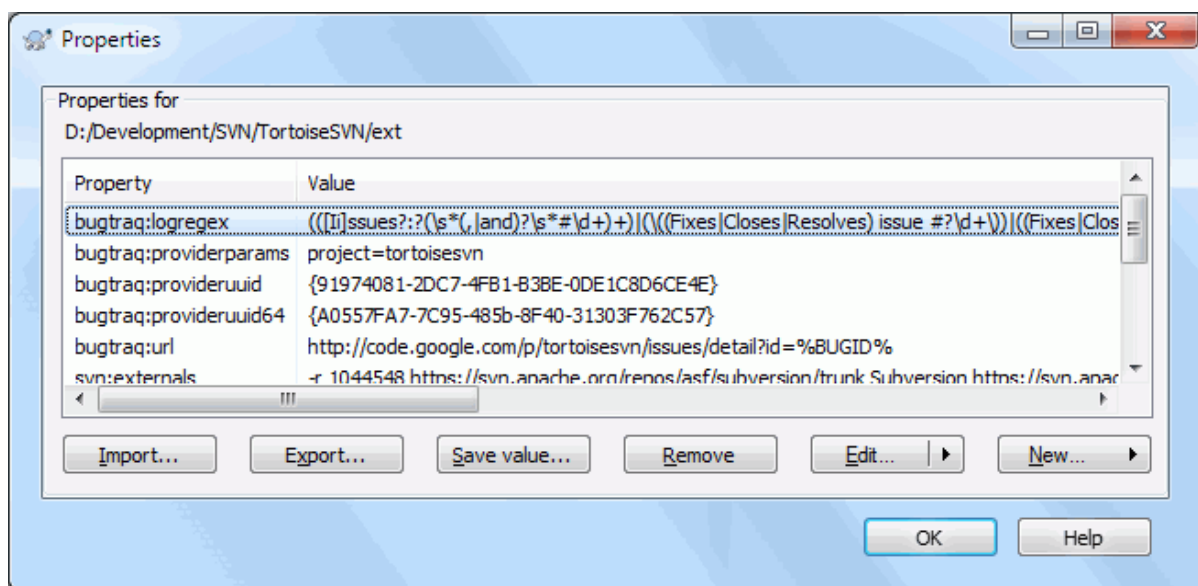
Hoàn nguyên tất cả các thay đổi một cách đệ quy

Lệnh này sẽ hoàn nguyên tất cả các sửa đổi địa phương mà chưa được cam kết.

Lưu ý: tốt hơn là sử dụng lệnh TortoiseSVN → Hoàn nguyên thay vào đó, bởi vì bạn có thể xem trước và chọn các tập tin mà bạn muốn hoàn nguyên.

## 4.17. Thiết lập Dự án

### 4.17.1. Thuộc tính Subversion



**Hình 4.34. Trang thuộc tính Subversion**

Bạn có thể đọc và thiết lập các thuộc tính Subversion từ hộp thoại thuộc tính Windows, nhưng cũng từ TortoiseSVN → thuộc tính và trong danh sách tình trạng TortoiseSVN, từ Trình đơn ngữ cảnh → thuộc tính .

Bạn có thể thêm thuộc tính của riêng bạn, hoặc một số thuộc tính với ý nghĩa đặc biệt trong Subversion. Những cái này bắt đầu với `svn:` . `svn:externals` là một thuộc tính như vậy, xem làm thế nào để xử lý thành phần bên ngoài [Phần 4.18, “Mục bên ngoài”](#) .

#### 4.17.1.1. `svn:keywords`

Subversion hỗ trợ mở rộng từ khóa giống như CVS mà có thể được sử dụng để nhúng tên tập tin và thông tin sửa đổi trong bản thân tập tin. Từ khóa được hỗ trợ là:

##### `$Date$`

Ngày cam kết cuối cùng được biết. Điều này được dựa trên thông tin thu được khi bạn cập nhật bản sao làm việc của bạn. Nó *không* kiểm tra các kho lưu trữ để tìm những thay đổi gần đây.

##### `$Revision$`

Bản sửa đổi cam của kết cuối cùng được biết.

##### `$Author$`

Tác giả đã thực hiện các cam kết cuối cùng được biết.

##### `$HeadURL$`

URL đầy đủ của tập tin này trong kho lưu trữ.

##### `$Id$`

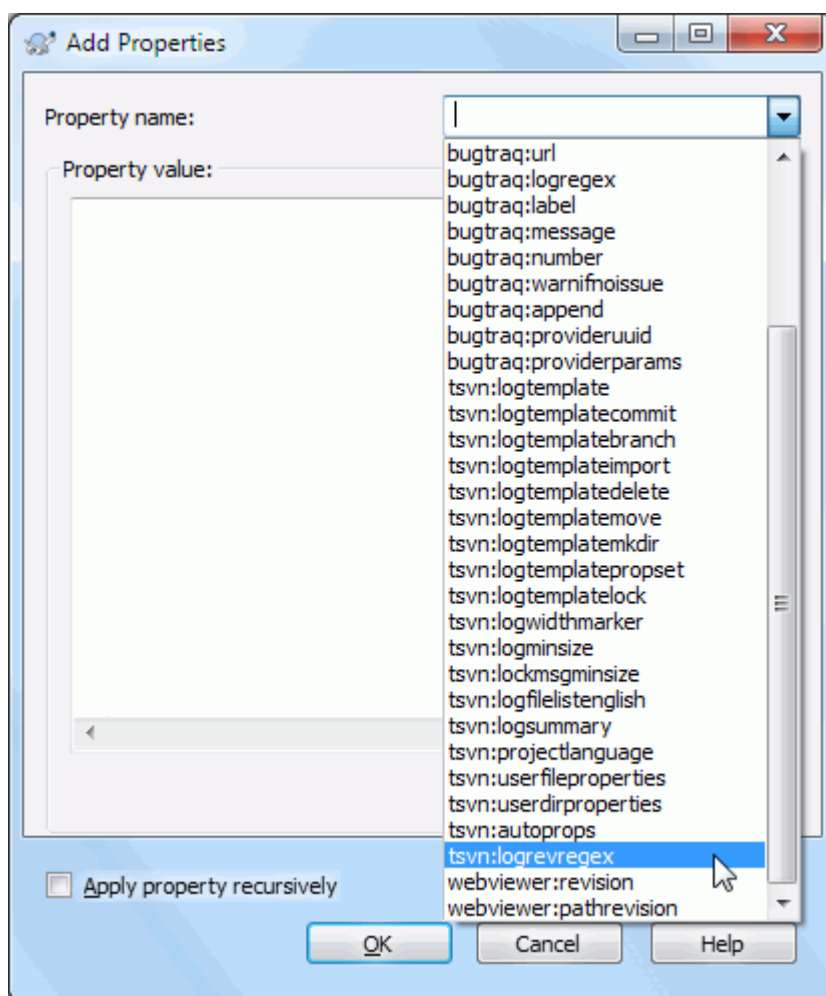
Một sự kết hợp nên của bốn từ khoá trước.

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].



### 4.17.1.2. Thêm và Chỉnh sửa các Thuộc tính



**Hình 4.35. Thêm thuộc tính**

Để thêm một thuộc tính mới, trước tiên nhập vào *Mới ...* . Chọn tên thuộc tính yêu cầu từ trình đơn, và sau đó điền vào các thông tin cần thiết trong hộp thoại thuộc tính cụ thể. Những hộp thoại thuộc tính cụ thể được mô tả chi tiết hơn trong [Phần 4.17.3, “Bộ Biên tập Thuộc tính”](#) .

Để thêm một thuộc tính mà không có hộp thoại riêng của mình, chọn Nâng cao từ trình đơn *Mới ...* . Sau đó chọn một thuộc tính hiện có trong hộp kết hợp hoặc nhập một tên thuộc tính tùy chỉnh.

Nếu bạn muốn áp dụng một thuộc tính cho nhiều mục cùng một lúc, chọn các tập tin / thư mục trong thám hiểm, sau đó chọn *Trình đơn ngữ cảnh* → *thuộc tính* .

Nếu bạn muốn áp dụng các thuộc tính *cho mỗi* tập tin và thư mục trong hệ thống phân cấp bên dưới thư mục hiện hành, kiểm tra hộp kiểm *Đệ quy* .

Nếu bạn muốn chỉnh sửa một thuộc tính hiện có, lựa chọn thuộc tính từ danh sách các thuộc tính hiện có, sau đó nhấn vào *Biên tập ...* .

Nếu bạn muốn loại bỏ một thuộc tính hiện có, lựa chọn thuộc tính từ danh sách các thuộc tính hiện có, sau đó nhấn vào *Loại bỏ* .

Các thuộc tính `svn:externals` có thể được sử dụng để kéo các dự án khác từ cùng kho lưu trữ hoặc kho lưu trữ hoàn toàn khác nhau. Để biết thêm thông tin, đọc [Phần 4.18, “Mục bên ngoài”](#) .



## Biên tập các thuộc tính tại bản sửa đổi HEAD

Bởi vì thuộc tính là được phiên bản, bạn không thể chỉnh sửa các thuộc tính của phiên bản trước. Nếu bạn nhìn vào thuộc tính từ hộp thoại đăng nhập, hoặc từ một phiên bản không phải HEAD trong trình duyệt kho lưu trữ, bạn sẽ thấy một danh sách các thuộc tính và giá trị, nhưng không có điều khiển biên tập.

### 4.17.1.3. Xuất khẩu và nhập khẩu Thuộc tính

Thường thì bạn sẽ thấy mình áp dụng cùng một thuộc tính nhiều lần, ví dụ `Bugtraq: logregex`. Để đơn giản hóa quá trình sao chép thuộc tính từ một dự án khác, bạn có thể sử dụng tính năng xuất khẩu / nhập khẩu.

Từ các tập tin hoặc thư mục nơi mà các thuộc tính đã được thiết lập, sử dụng **TortoiseSVN** → **thuộc tính**, Chọn các thuộc tính bạn muốn xuất khẩu, kích vào **Xuất khẩu ...**. Bạn sẽ được nhắc nhở cho một tên tập tin nơi tên thuộc tính và giá trị sẽ được lưu.

Từ thư mục nơi bạn muốn áp dụng những thuộc tính, sử dụng **TortoiseSVN** → **thuộc tính** và nhấp vào **Nhập khẩu ...**. Bạn sẽ được nhắc nhở cho một tên tập tin để nhập khẩu từ đó, do đó, chuyển hướng đến nơi bạn đã lưu các tập tin xuất khẩu trước đó và chọn nó. Các thuộc tính sẽ được thêm vào thư mục không theo cách đệ quy.

Nếu bạn muốn thêm các thuộc tính vào cây theo cách đệ quy, theo các bước trên, sau đó trong hộp thoại thuộc tính lần lượt lựa chọn từng thuộc tính, nhấp vào **Chỉnh sửa ...**, Kiểm tra hộp **Áp dụng thuộc tính đệ quy** và nhấp vào **OK**.

Các định dạng tập tin nhập khẩu là nhị phân và độc quyền của TortoiseSVN. Mục đích duy nhất của nó là để chuyển giao thuộc tính bằng cách sử dụng nhập khẩu và xuất khẩu, do đó không cần phải chỉnh sửa những tập tin này.

### 4.17.1.4. Thuộc tính Nhị phân

TortoiseSVN có thể xử lý các giá trị thuộc tính nhị phân sử dụng các tập tin. Để đọc một thuộc tính giá trị nhị phân, Lưu ... vào một tập tin. Để đặt một giá trị nhị phân, sử dụng một trình soạn thảo hex hoặc một công cụ thích hợp khác để tạo ra một tập tin với các nội dung mà bạn yêu cầu, sau đó Tải ... từ tập tin đó.

Mặc dù các thuộc tính nhị phân không thường xuyên được sử dụng, chúng có thể hữu ích trong một số ứng dụng. Ví dụ, nếu bạn đang lưu trữ các tập tin đồ họa lớn, hoặc nếu ứng dụng được sử dụng để tải các tập tin là rất lớn, bạn có thể muốn để lưu trữ một hình ảnh thu nhỏ như là một thuộc tính để bạn có thể có được một bản xem trước một cách nhanh chóng.

### 4.17.1.5. Thiết lập thuộc tính tự động

Bạn có thể cấu hình Subversion và TortoiseSVN để thiết lập các thuộc tính tự động trên các tập tin và thư mục khi được được bổ sung vào kho lưu trữ. Có hai cách để làm điều này.

Bạn có thể chỉnh sửa các tập tin cấu hình Subversion để kích hoạt tính năng này trên máy khách của bạn. Trang **Tổng quát** trong hộp thoại thiết lập TortoiseSVN có nút chỉnh sửa để đưa bạn đến đó trực tiếp. Tập tin cấu hình là một tập tin văn bản đơn giản, kiểm soát một số hoạt động của Subversion. Bạn cần thay đổi hai điều: trước hết là trong phần `linh tinhliteral> benable-auto-props = yesliteral>`. Th

Một phương pháp thay thế là để thiết lập thuộc tính `tsvn:autoprops` trên các thư mục, như mô tả trong phần tiếp theo. Phương pháp này chỉ làm việc cho các máy khách TortoiseSVN, nhưng nó không nhận được tuyên truyền cho tất cả các bản sao làm việc lúc cập nhật.

Cho dù bạn chọn phương pháp, bạn nên lưu ý rằng props-tự động chỉ áp dụng cho các tập tin tại thời điểm chúng được thêm vào bản sao làm việc. Props-tự động sẽ không bao giờ thay đổi các thuộc tính của tập tin đã phiên bản.

Nếu bạn muốn được hoàn toàn chắc chắn rằng các tập tin mới có áp dụng các thuộc tính đúng, bạn nên thiết lập một kho lưu trữ trước khi hook tiền cam kết để từ chối cam kết khi các thuộc tính được yêu cầu không được thiết lập.



## Cam kết các thuộc tính

Thuộc tính Subversion được phiên bản. Sau khi bạn thay đổi hoặc thêm một thuộc tính, bạn phải cam kết thay đổi của bạn.



## Xung đột về thuộc tính

Nếu có một cuộc xung đột lúc cam kết thay đổi, bởi vì một người dùng khác đã thay đổi thuộc tính, Subversion tạo ra một tập tin `.prej`. Xóa tập tin sau khi bạn đã giải quyết được xung đột.

### 4.17.2. Thuộc tính dự án TortoiseSVN

TortoiseSVN có một vài thuộc tính đặc biệt của riêng mình, và những cái đó bắt đầu với `tsvn:`.

- `tsvn:logminsize` đặt chiều dài tối thiểu của một thông điệp đăng nhập cho một cam kết. Nếu bạn nhập một thông điệp ngắn hơn so với quy định ở đây, cam kết bị vô hiệu hóa. Tính năng này rất hữu ích để nhắc nhở bạn để cung cấp một thông điệp mô tả thích hợp cho mỗi cam kết. Nếu thuộc tính này được không được thiết lập, hoặc giá trị là số không, các thông điệp tường trình trông được cho phép.

`tsvn:lockmsgminsize` thiết lập chiều dài tối thiểu của một thông điệp khóa. Nếu bạn nhập một thông báo ngắn hơn so với quy định ở đây, khóa bị vô hiệu hóa. Tính năng này rất hữu ích để nhắc nhở bạn cung cấp một thông điệp mô tả thích hợp cho mỗi khóa bạn nhận được. Nếu thuộc tính này không được thiết lập, hoặc giá trị là số không, các thông điệp khóa trông được cho phép.

- `tsvn:logwidthmarker` được sử dụng với các dự án yêu cầu các thông điệp tường trình để được định dạng với một số chiều rộng tối đa (thường là 80 ký tự) trước khi ngắt dòng. Đặt thuộc tính này tới một giá trị khác-không sẽ làm 2 việc trong hộp thoại thông điệp tường trình: đặt một điểm đánh dấu để chỉ ra chiều rộng tối đa, và nó vô hiệu hóa gói gọn chữ trong màn hình, để bạn có thể xem các văn bản bạn quá dài. Lưu ý: tính năng này sẽ chỉ hoạt động được nếu bạn có một phong chữ cố định chiều rộng được lựa chọn cho các thông điệp tường trình.
- `tsvn:logtemplate` được sử dụng với các dự án có những quy định về định dạng thông điệp tường trình. Thuộc tính nắm giữ một chuỗi văn bản nhiều dòng sẽ được chèn vào trong hộp thông báo cam kết khi bạn bắt đầu một cam kết. Sau đó, bạn có thể chỉnh sửa nó để bao gồm các thông tin cần thiết. Lưu ý: nếu bạn cũng sử dụng `tsvn:logminsize`, hãy chắc chắn để thiết lập chiều dài dài hơn mẫu hoặc bạn sẽ mất cơ chế bảo vệ.

Ngoài ra còn có các mẫu hành động cụ thể mà bạn có thể sử dụng thay vì `tsvn:logtemplate`. Các mẫu hành động cụ thể được sử dụng nếu thiết lập, nhưng `tsvn:logtemplate` sẽ được sử dụng nếu không có khuôn mẫu hành động cụ thể được thiết lập.

Các mẫu hành động cụ thể là:

- `tsvn:logtemplatecommit` được sử dụng cho tất cả các cam kết từ một bản sao làm việc.
- `tsvn:logtemplatebranch` được sử dụng khi bạn tạo ra một chi nhánh/thẻ, hoặc khi bạn sao chép các tập tin hoặc thư mục trực tiếp trong trình duyệt kho lưu trữ.
- `tsvn:logtemplateimport` được sử dụng đối với nhập khẩu.
- `tsvn:logtemplatedelete` được sử dụng khi xóa trực tiếp các mục trong trình duyệt kho.
- `tsvn:logtemplatemove` được sử dụng khi đổi tên hoặc di chuyển các mục trong trình duyệt kho.
- `tsvn:logtemplatemkdir` được sử dụng khi tạo các thư mục trong trình duyệt kho lưu trữ.
- `tsvn:logtemplatepropset` được sử dụng khi sửa đổi các thuộc tính trong trình duyệt kho lưu trữ.
- `tsvn:logtemplatelock` được sử dụng khi lấy khóa.

- Subversion cho phép bạn thiết lập “autoprops” sẽ được áp dụng cho các tập tin mới được thêm vào hoặc được nhập khẩu, dựa trên phần mở rộng tập tin. Điều này phụ thuộc vào tất cả các khách hàng đã thiết lập autoprops thích hợp trong tập tin cấu hình Subversion của chúng.
 

```

TSVN: autopropsliteral> c*.sh
= svn:eol-style=native,svn:executable thi.shliteral>.N tsvn:autoprops ,
cTrong h tsvn:logfilelistenglish .TortoiseSVN c tsvn:projectlanguage
thi MSDN: .B 0x . V 0x0409 ho 1033 .Thu tsvn:logsummary
Gi tsvn:logsummary phM \ [SUMMARY\]:\s+(.*) S [SUMMARY] trong
thThu tsvn:logrevregex BinCThe available properties/hook-scripts are
tsvn:startcommithooktsvn:precommithooktsvn:postcommithooktsvn:startupdatehooktsvn:preu
%REPOROOT% nhcontrib/hook-scripts/ client-side/checkyear.js, b%REPOROOT%/
trunk/contrib/hook-scripts/client-side/checkyear.js. BInstead 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.CHKhi
b tsvn:userfileproperties v tsvn:userdirproperties . BboolNbool.
H Chpropertyname=bool;labeltext (YESVALUE;NOVALUE;Checkboxtext) labeltext
lstateNyes, no, maybe, thtrH
nhpropertyname=state;labeltext (DEFVAL;VAL1;TEXT1;VAL2;TEXT2;VAL3;TEXT3;...) Cbool,
vDEFVAL lCho tsinglelineCho csingleline
kiHpropertyname=singleline;labeltext (regex) regex chmultilineCho multiline
kiHpropertyname=multiline;labeltext (regex) regex ch
Ctsvn:userdirproperties: my:boolprop=bool;”

```

TortoiseSVN có thể tích hợp với một số công cụ theo dõi lỗi. Điều này sử dụng các thuộc tính dự án bắt đầu bằng bugtraq: . Đọc [Phần 4.28, “Tích hợp với Hệ thống Theo dõi Lỗi / Trình theo dõi Vấn đề”](#) để biết thêm thông tin.

Nó cũng có thể tích hợp với một số trình duyệt kho lưu trữ dựa trên web, sử dụng các thuộc tính dự án bắt đầu bằng webviewer: . Đọc [Phần 4.29, “Tích hợp với các Trình Xem Kho Lưu Trữ dựa trên Web”](#) để biết thêm thông tin.



## Thiết lập các thuộc tính của dự án trên các thư mục

Những thuộc tính dự án đặc biệt phải được đặt trên *thư mục* cho hệ thống để làm việc. Khi bạn sử dụng một lệnh TortoiseSVN trong đó sử dụng các thuộc tính này, các thuộc tính được đọc từ thư mục mà bạn nhập vào. Nếu thuộc tính không được tìm thấy ở đó, TortoiseSVN sẽ tìm kiếm lên trên thông qua cây thư mục để tìm thấy chúng cho đến khi nó đến một thư mục không phiên bản, hoặc gốc của cây (ví dụ: C: \ ) được tìm thấy. Nếu bạn có thể chắc chắn rằng mỗi người dùng chỉ kiểm tra ra từ ví dụ như trunk/ và không phải là một thư mục con, thì nó là đủ để thiết lập các thuộc tính trên trunk/ . Nếu bạn không thể chắc chắn, bạn nên thiết lập các thuộc tính một cách đệ quy trên mỗi thư mục con. Nếu bạn thiết lập thuộc tính tương tự nhưng sử dụng giá trị khác nhau tại các độ sâu khác nhau trong hệ thống phân cấp dự án của bạn thì bạn sẽ có được kết quả khác nhau tùy thuộc vào nơi bạn nhập vào cấu trúc thư mục.

Đối với thuộc tính *chỉ* của dự án, tức là tsvn: , bugtraq: và webviewer: bạn có thể sử dụng hộp kiểm **Đệ quy** để thiết lập thuộc tính cho tất cả các thư mục con trong hệ thống phân cấp, mà không cần đặt nó trên tất cả các tập tin.

Khi bạn thêm mới các thư mục con vào một bản sao làm việc bằng cách sử dụng TortoiseSVN, bất kỳ thuộc tính dự án có trong thư mục mẹ sẽ tự động được thêm vào thư mục con.



## Các Hạn chế khi Sử dụng Trình duyệt kho

Lấy thuộc tính từ xa là một hoạt động chậm, vì vậy một số tính năng được mô tả ở trên sẽ không làm việc trong trình duyệt kho lưu trữ như chúng làm được trong một bản sao làm việc.

- Khi bạn thêm một thuộc tính bằng cách sử dụng trình duyệt kho, chỉ có thuộc tính tiêu chuẩn `svn:` được cung cấp trong danh sách được xác định trước. Bất kỳ tên thuộc tính khác phải được nhập bằng tay.
- Thuộc tính không có thể được thiết lập hoặc xóa một cách đệ quy bằng cách sử dụng trình duyệt kho.
- Thuộc tính dự án *không* được truyền tự động khi một thư mục con được thêm vào bằng cách sử dụng trình duyệt kho.
- `tsvn:autoprops` sẽ *không* thiết lập các thuộc tính trên các tập tin được thêm vào bằng cách sử dụng trình duyệt kho.



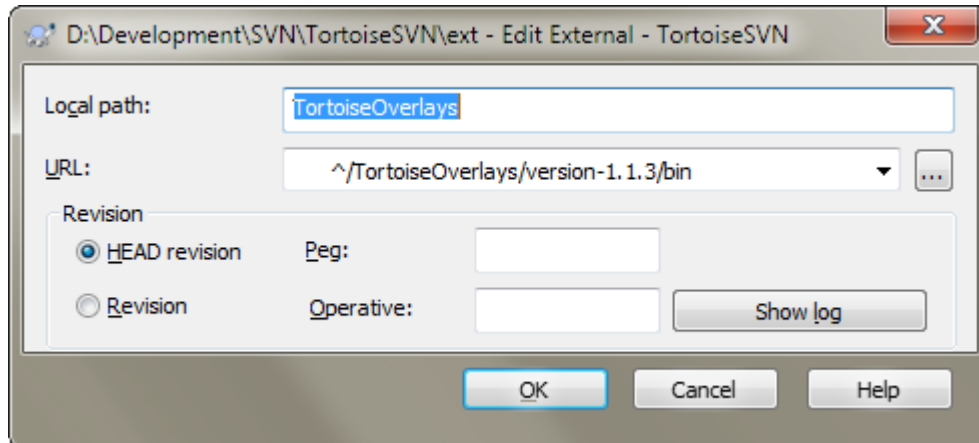
### Cẩn thận

Mặc dù thuộc tính dự án của TortoiseSVN là cực kỳ hữu ích, chúng chỉ làm việc với TortoiseSVN, và một số sẽ chỉ làm việc trong các phiên bản mới hơn của TortoiseSVN. Nếu những người làm việc trên dự án của bạn sử dụng một loạt các máy khách Subversion khác nhau, hoặc có thể có các phiên bản cũ của TortoiseSVN, bạn có thể muốn sử dụng hook kho lưu trữ để thực thi các chính sách dự án. Tài sản dự án chỉ có thể giúp đỡ để thực hiện một chính sách, chúng không thể thực thi nó.

## 4.17.3. Bộ Biên tập Thuộc tính

Một số thuộc tính phải sử dụng các giá trị cụ thể, hoặc được định dạng theo một cách cụ thể để có thể được sử dụng cho tự động hóa. Để giúp có các định dạng chính xác, TortoiseSVN trình bày các hộp thoại chỉnh sửa đối với một số thuộc tính đặc biệt cho thấy các giá trị có thể hoặc phá vỡ thuộc tính thành các thành phần cá nhân.

### 4.17.3.1. Nội dung bên ngoài



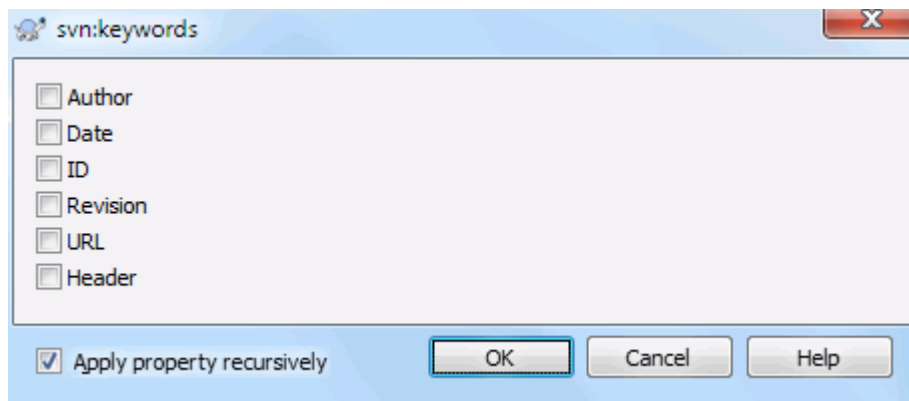
**Hình 4.36. Trang thuộc tính svn: externals**

Các `svn:externals` thuộc tính có thể được sử dụng để kéo các dự án khác từ cùng một kho lưu trữ hoặc từ kho lưu trữ hoàn toàn khác nhau như được mô tả trong [Phần 4.18, “Mục bên ngoài”](#).

Bạn cần phải xác định tên của thư mục con mà các thư mục bên ngoài được kiểm xuất ra, và URL Subversion của các mục bên ngoài. Bạn có thể kiểm xuất một thành phần bên ngoài tại bản CHÍNH của nó, để khi mục trong kho lưu trữ bên ngoài thay đổi, bản sao làm việc của bạn sẽ nhận được những thay đổi đó lúc cập nhật. Tuy nhiên, nếu bạn muốn thành phần bên ngoài tham khảo một điểm cụ thể ổn định thì bạn có thể xác định phiên bản cụ thể để sử dụng. Trong trường hợp này, bạn cũng có thể muốn xác định cùng phiên bản như một phiên bản làm mốc. Nếu mục bên ngoài được đổi tên tại một thời điểm nào đó trong tương lai thì Subversion sẽ không thể cập nhật mục này trong bản sao làm việc của bạn. Bằng việc chỉ định một phiên bản mốc bạn nói với Subversion tìm một mục có tên đó ở sửa đổi mốc hơn là tại bản CHÍNH.

Nút bấm Tìm bản sửa đổi Chính nạp bản sửa đổi CHÍNH của mỗi URL ngoài và hiển thị bản sửa đổi CHÍNH đó trong cột tận cùng bên phải. Sau khi một bản sửa đổi CHÍNH được biết, một cú nhấp chuột phải đơn giản trên một thành phần ngoài cho bạn một lệnh để neo các thành phần ngoài vào bản sửa đổi CHÍNH rõ ràng của chúng. Trong trường hợp bản sửa đổi CHÍNH chưa được biết, lệnh nhấp chuột phải sẽ nạp bản sửa đổi CHÍNH trước.

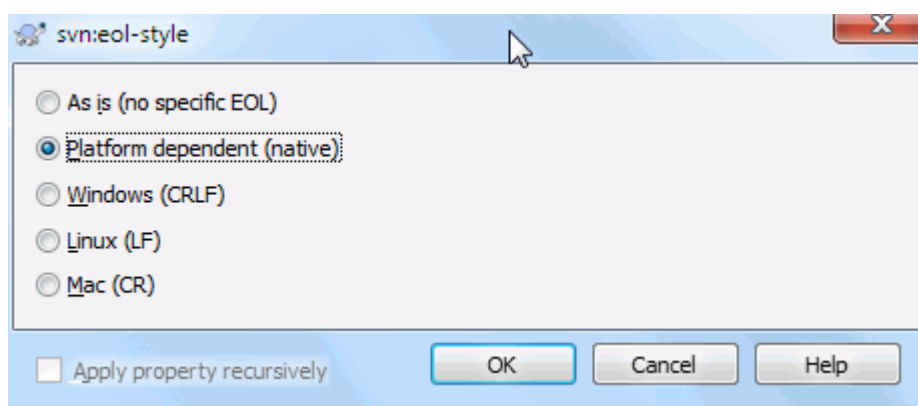
#### 4.17.3.2. Từ khóa SVN



Hình 4.37. Trang thuộc tính svn:keywords

Chọn từ khoá mà bạn muốn được mở rộng trong tập tin của bạn.

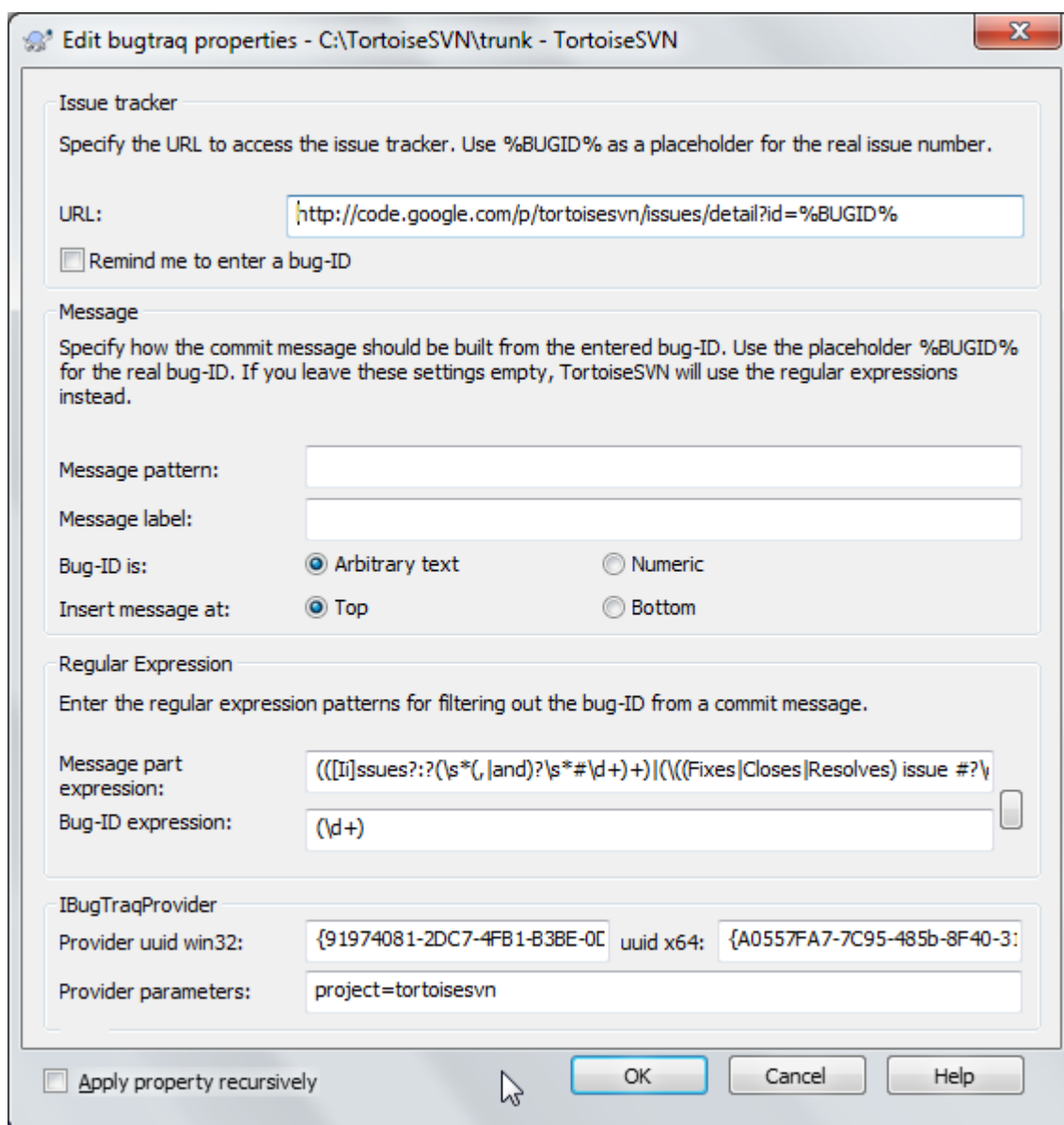
#### 4.17.3.3. Kiểu EOL



Hình 4.38. Trang thuộc tính svn:eol-style

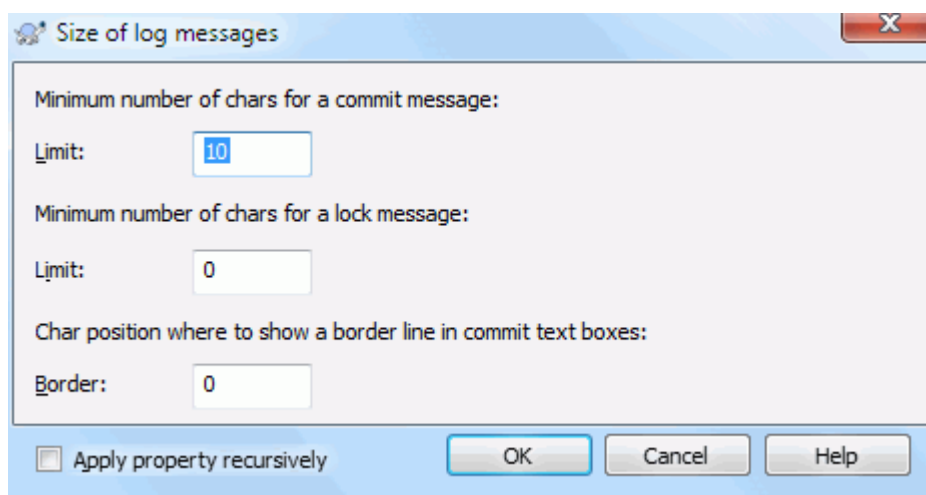
Chọn phong cách-cuối-dòng mà bạn muốn sử dụng và TortoiseSVN sẽ sử dụng đúng giá trị thuộc tính.

#### 4.17.3.4. Tích hợp bộ theo dõi lỗi



Hình 4.39. Trang thuộc tính tsvn:bugtraq

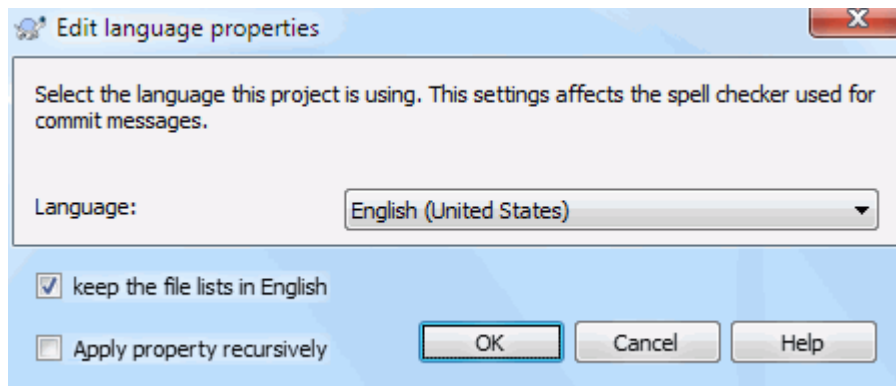
#### 4.17.3.5. Kích thước thông điệp tường trình



**Hình 4.40. Kích thước của trang thuộc tính thông điệp tường trình**

3 thuộc tính này điều khiển định dạng của các thông điệp tường trình. 2 cái đầu tiên vô hiệu hóa OK trong hộp thoại cam kết hoặc khóa cho đến khi thông điệp đáp ứng được chiều dài tối thiểu. Vị trí biên hiện một dấu hiệu tại chiều rộng cột như là một hướng dẫn cho những dự án có giới hạn chiều rộng trên những thông điệp tường trình của chúng. Đặt giá trị bằng không sẽ xóa thuộc tính này.

#### 4.17.3.6. Ngôn ngữ dự án

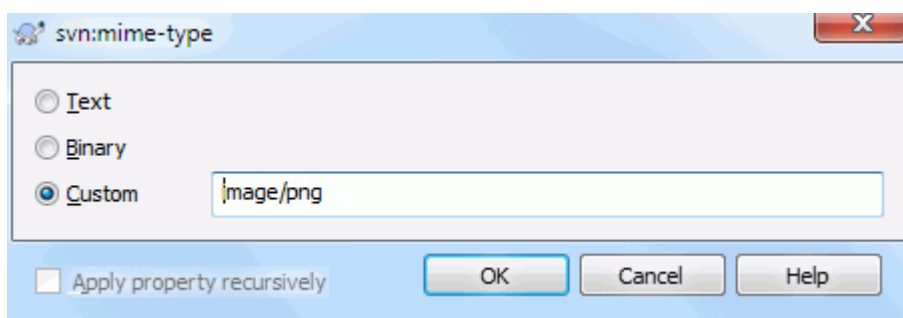


**Hình 4.41. Trang thuộc tính ngôn ngữ**

Chọn ngôn ngữ để sử dụng cho kiểm tra chính tả cho thông điệp tường trình trong hộp thoại cam kết. Hộp kiểm danh sách tập tin có hiệu lực khi bạn nhấp chuột phải trong cửa sổ thông điệp tường trình và chọn **Dán danh sách tập tin**. Theo mặc định, trạng thái Subversion sẽ được hiển thị bằng ngôn ngữ địa phương của bạn. Khi hộp này được kiểm, trạng thái luôn luôn được đưa ra bằng tiếng Anh, đối với các dự án có yêu cầu thông điệp ghi chỉ tiếng Anh.

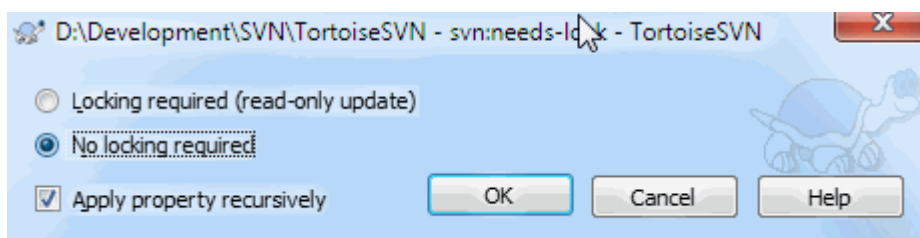


#### 4.17.3.7. MIME-type



Hình 4.42. trang thuộc tính svn:mime-type

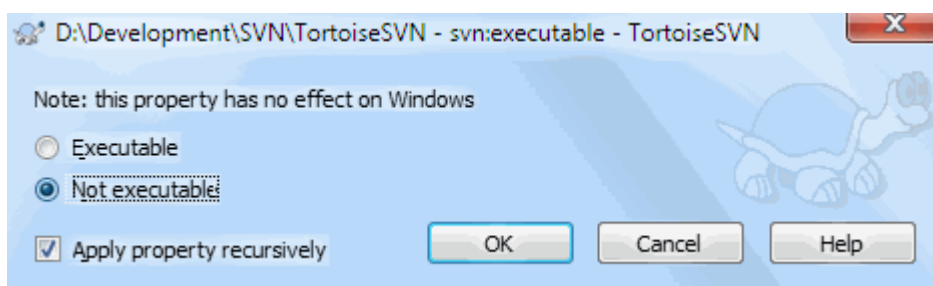
#### 4.17.3.8. svn:needs-lock



Hình 4.43. trang thuộc tính svn:needs-lock

Thuộc tính này chỉ đơn giản là kiểm soát xem một tập tin sẽ được kiểm tra ra là chỉ đọc nếu không có khóa được giữ cho nó trong bản sao làm việc.

#### 4.17.3.9. svn:executable



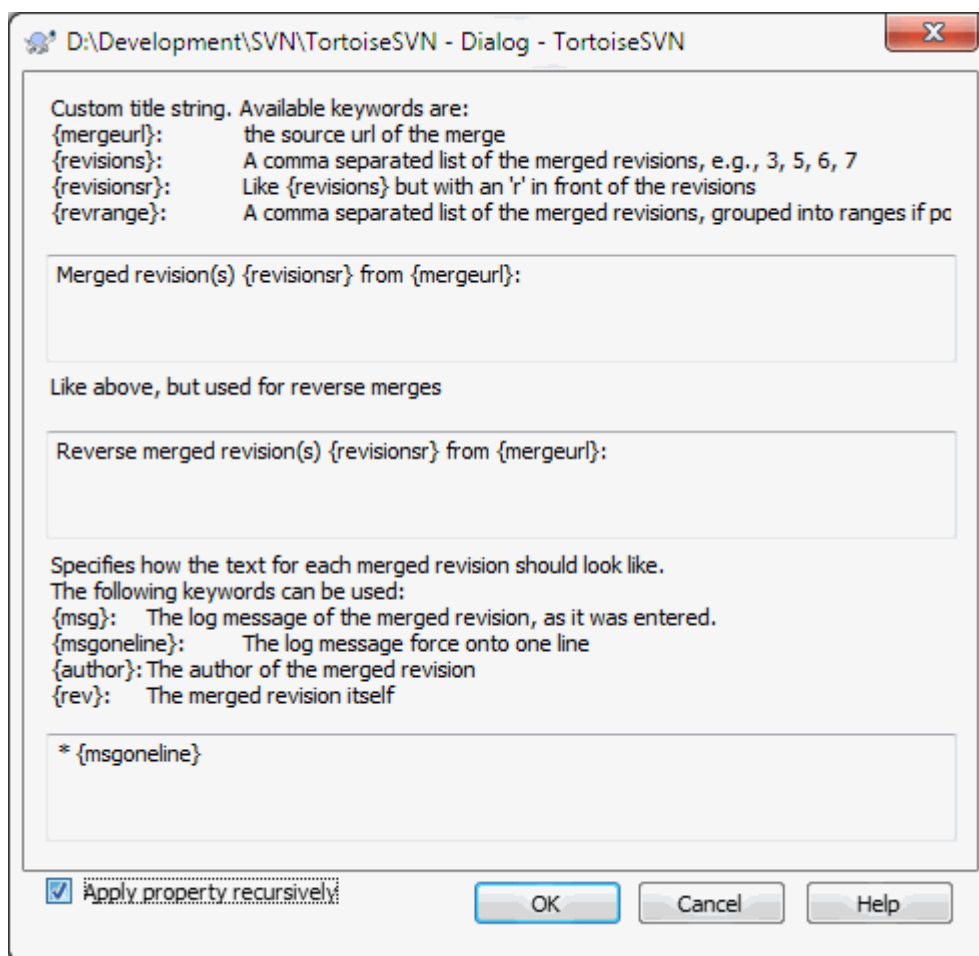
Hình 4.44. trang thuộc tính svn:executable

Thuộc tính này điều khiển xem một tập tin sẽ được thực thi khi kiểm tra ra trên một hệ thống Unix / Linux. Nó không có hiệu lực khi kiểm tra ra trên Windows.

#### 4.17.3.10. Các mẫu thông điệp hợp nhất

Khi các bản sửa đổi được hợp nhất vào bản làm việc, TortoiseSVN tạo ra một thông điệp tường trình từ các bản chỉnh sửa được hợp nhất. Những bản đó được cung cấp từ nút bấm Các Thông Điệp Gần Đây trong hộp thoại cam kết.

Bạn có thể tùy chỉnh thông điệp được sinh ra với các thuộc tính sau:



**Hình 4.45. Các mẫu thông điệp tường trình cho hộp thoại thuộc tính**

tsvn:mergelogtemplatetitle, tsvn:mergelogtemplatereversetitle

Thuộc tính này chỉ định phần đầu tiên của thông điệp tường trình được sinh ra. Các từ khóa sau có thể được dùng:

{revisions}

Một danh sách các bản chỉnh sửa được hợp nhất được phân cách bởi dấu phẩy, ví dụ: 3, 5, 6, 7

{revisionsr}

Giống như {revisions}, nhưng với mỗi bản sửa đổi được gắn phía trước với chữ r, ví dụ, r3, r5, r6, r7

{revrange}

Một danh sách các bản chỉnh sửa được hợp nhất được phân cách bởi dấu phẩy, được nhóm thành các phạm vi nếu có thể, ví dụ: 3, 5-7

{mergeurl}

URL nguồn của kết hợp, ví dụ, nơi từ đó các bản sửa đổi được kết hợp.

Giá trị mặc định cho chuỗi là Các bản sửa đổi được hợp nhất {revrange} từ {mergeurl}: với đánh dấu dòng mới ở cuối.

tsvn:mergelogtemplatemsg

Thuộc tính này chỉ định cách văn bản cho mỗi bản hợp nhất sẽ nhìn thế nào. Các từ khóa sau có thể được dùng:

{msg}

Thông điệp tường trình của bản sửa đổi được hợp nhất, như nó được nhập vào.

{msgonline}

Giống như {msg}, nhưng tất cả các dòng mới được thay thế bởi khoảng trắng, để cho toàn bộ các thông điệp tường trình được hiện ra trên một dòng.

{author}

Tác giả của bản hợp nhất

{rev}

Bản thân bản chỉnh sửa được hợp nhất.

{bugids}

ID lỗi của bản hợp nhất, nếu có.



### Quan trọng

Việc này chỉ làm được nếu các bản hợp nhất đã nằm trong bộ nhớ cache tường trình. Nếu bạn vô hiệu hóa cache tường trình hoặc không hiện tường trình trước đó, thông điệp sinh ra sẽ không chứa thông tin về các bản hợp nhất.

## 4.18. Mục bên ngoài

Đôi khi nó rất hữu ích để xây dựng một bản sao làm việc được tạo ra bởi một số kiểm xuất khác nhau. Ví dụ, bạn có thể muốn các tập tin hoặc thư mục con khác nhau đến từ các địa điểm khác nhau trong một kho lưu trữ, hoặc có lẽ từ các kho khác nhau hoàn toàn. Nếu bạn muốn mọi người sử dụng có cùng một cách bố trí, bạn có thể xác định thuộc tính `svn:externals` để kéo vào nguồn tài nguyên được chỉ định tại các địa điểm nơi chúng là cần thiết.

### 4.18.1. Thư mục bên ngoài

Hãy nói rằng bạn hãy kiểm xuất một bản sao làm việc của `/project1` sang `D:\dev\project1`. Chọn thư mục `D:\dev\project1`, nhấp chuột phải và chọn **Cửa sổ trình đơn** → **Thuộc tính** từ trình đơn ngữ cảnh. Hộp thoại Thuộc Tính xuất hiện. Sau đó vào tab Subversion. Ở đó, bạn có thể thiết lập các thuộc tính. Nhấp vào **Thuộc tính ...**. Trong hộp thoại thuộc tính, hoặc nhấp đúp chuột vào `svn:externals` nếu nó đã tồn tại, hoặc nhấp vào nút **Mới ...** và chọn `externals` từ trình đơn. Để thêm vào một thành phần bên ngoài mới, nhấp vào **Mới ...** và sau đó điền vào các thông tin cần thiết trong hộp thoại hiển thị.



### Cẩn thận

URL phải được thoát ra đúng đắn nếu không chúng sẽ không làm việc, ví dụ như bạn phải thay thế mỗi khoảng trống bằng `%20`.

Nếu bạn muốn đường dẫn địa phương bao gồm các khoảng trống hoặc ký tự đặc biệt khác, bạn có thể đặt nó bên trong dấu ngoặc kép, hoặc bạn có thể sử dụng ký tự `\` (Dấu gạch chéo ngược) như ký tự thoát kiểu phong cách Unix shell trước bất kỳ ký tự đặc biệt nào. Trong quá trình này cũng có nghĩa là bạn phải sử dụng `/` (Dấu gạch chéo) như một dấu phân cách đường dẫn. Lưu ý rằng hành vi này mới có trong Subversion 1.6 và sẽ không làm việc với các máy khách cũ.



### Sử dụng số sửa đổi rõ ràng

Bạn nên xem xét mạnh mẽ việc sử dụng số sửa đổi rõ ràng trong tất cả các định nghĩa bên ngoài của bạn, như mô tả ở trên. Làm như vậy có nghĩa là bạn có thể quyết định khi nào để kéo xuống một bản chụp khác nhau của thông tin bên ngoài, và chính xác là bản chụp nào để kéo. Bên cạnh các khía cạnh chung để không bị ngạc nhiên bởi những thay đổi đến các kho của bên thứ ba mà bạn có thể không có bất kỳ kiểm soát nào, sử dụng số sửa đổi rõ ràng cũng có nghĩa là khi bạn lùi ngày bản sao làm việc của bạn với một phiên bản trước đó, các định nghĩa bên ngoài của bạn cũng sẽ trở lại theo cách mà chúng được nhìn vào trong phiên bản trước đó, lần lượt có nghĩa là các bản sao làm việc

bên ngoài sẽ được cập nhật để phù hợp với họ cách *chúng* được nhìn lại khi kho của bạn tại sửa đổi trước đó. Đối với các dự án phần mềm, điều này có thể là sự khác biệt giữa một xây dựng thành công và thất bại của một bản chụp cũ của cơ sở mã phức tạp của bạn.

Hộp thoại biên tập cho thuộc tính `svn:externals` cho phép bạn chọn các thành phần ngoài và tự động đặt chúng một cách rõ ràng vào bản chính sửa CHÍNH.

Nếu dự án bên ngoài là trong cùng một kho lưu trữ, bất kỳ thay đổi bạn thực hiện sẽ được bao gồm trong danh sách cam kết khi bạn cam kết dự án chính của bạn.

Nếu dự án bên ngoài nằm trong một kho lưu trữ khác, bất kỳ thay đổi bạn thực hiện đến các dự án bên ngoài sẽ được thông báo khi bạn cam kết dự án chính, nhưng bạn phải cam kết những thay đổi bên ngoài một cách riêng biệt.

Nếu bạn sử dụng các URL tuyệt đối `svn:externals` trong các định nghĩa và bạn có di chuyển bản sao làm việc của bạn (ví dụ, nếu URL của kho của bạn thay đổi), sau đó các thành phần bên ngoài của bạn sẽ không thay đổi và có thể không làm việc nữa.

Để tránh các vấn đề như vậy, máy khách Subversion phiên bản 1.5 và mới hơn hỗ trợ các URL bên ngoài tương đối. Bốn phương pháp khác nhau để chỉ định một URL tương đối được hỗ trợ. Trong các ví dụ sau đây, giả sử chúng ta có hai kho: một ở `http://example.com/svn/repos-1` và một tại `http://example.com/svn/repos-2`. Chúng ta đã kiểm xuất `http://example.com/svn/repos-1/project/trunk` vào `C:\Working` và thuộc tính `svn:externals` được thiết lập trên thân.

Tương đối so với thư mục mẹ

Các URL luôn luôn bắt đầu với chuỗi `../` ví dụ:

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

Điều này sẽ trích xuất `http://example.com/svn/repos-1/widgets/foo` vào `C:\Working\common\foo-widget`.

Lưu ý rằng URL là tương đối với địa chỉ URL của thư mục với thuộc tính `svn:externals`, không phải với thư mục nơi thành phần bên ngoài được ghi vào đĩa.

Tương đối so với gốc kho

Các URL này luôn luôn bắt đầu với chuỗi `^/` ví dụ:

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

Điều này sẽ trích `http://example.com/svn/repos-1/widgets/foo` vào `C:\Working\common\foo-widget`.

Bạn có thể dễ dàng tham khảo các kho khác với cùng một `SVNParentPath` (Một thư mục thông thường chứa nhiều kho). Ví dụ:

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

Điều này sẽ trích `http://example.com/svn/repos-2/hammers/claw` vào `C:\Working\common\claw-hammer`.

Tương đối với giản đồ

Các URL bắt đầu với chuỗi `//` chỉ sao chép phần giản đồ của URL. Điều này rất hữu ích khi cùng một tên máy phải truy cập với các giản đồ khác nhau tùy thuộc vào vị trí mạng; ví dụ như máy khách trong mạng nội bộ sử dụng `http://` trong khi các máy khách bên ngoài sử dụng `svn+ssh://`. Ví dụ:

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

Điều này sẽ trích `http://example.com/svn/repos-1/widgets/foo` hoặc `svn+ssh://example.com/svn/repos-1/widgets/foo` tùy thuộc vào phương pháp được sử dụng để kiểm xuất `C:\Working`.

Tương đối so với tên chủ của máy chủ

Các URL bắt đầu với chuỗi `/` sao chép gián đồ và một phần tên chủ của URL, ví dụ:

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

Điều này sẽ trích `http://example.com/svn/repos-1/widgets/foo` vào `C:\Working\common\foo-widget`. Nhưng nếu bạn kiểm xuất bản sao của bạn làm việc từ một máy chủ khác tại `svn+ssh://another.mirror.net/svn/repos-1/project1/trunk` thì các tham chiếu bên ngoài sẽ trích xuất `svn+ssh://another.mirror.net/svn/repos-1/widgets/foo`.

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

Nếu bạn cần thêm thông tin về làm cách nào TortoiseSVN xử lý các thuộc tính đọc [Phần 4.17, “Thiết lập Dự án”](#).

Để tìm hiểu về các phương pháp khác nhau để truy cập các tiểu dự án chung hãy đọc [Phần B.6, “Bao gồm các tiểu dự án chung”](#).

#### 4.18.2. Các tập tin bên ngoài

Kể từ Subversion 1.6, bạn có thể thêm các thành phần bên ngoài trong một tập tin vào bản sao làm việc của bạn bằng cách sử dụng các cú pháp tương tự như đối với các thư mục. Tuy nhiên, có một số hạn chế.

- Đường dẫn đến tập tin bên ngoài phải đặt các tập tin trong một thư mục được phiên bản đang tồn tại. Nói chung, nó có ý nghĩa nhất để đặt các tập tin trực tiếp trong thư mục có `svn:externals` được thiết lập, nhưng nó có thể được ở trong một thư mục con được phiên bản nếu cần thiết. Ngược lại, thư mục bên ngoài sẽ tự động tạo ra bất kỳ thư mục trung gian không phiên bản theo yêu cầu.
- URL cho một tập tin bên ngoài phải ở trong cùng một kho lưu trữ như địa chỉ URL tập tin bên ngoài sẽ được chèn vào; tập tin thành phần ngoài liên kho lưu trữ không được hỗ trợ.

Một tập tin bên ngoài cư xử giống như bất kỳ tập tin được phiên bản khác trong nhiều khía cạnh, nhưng chúng không thể được di chuyển hoặc xóa bằng cách sử dụng các lệnh bình thường, thay vào đó thuộc tính `svn:externals` phải được sửa đổi.

#### 4.19. Phn Nhánh / Tạo Thẻ

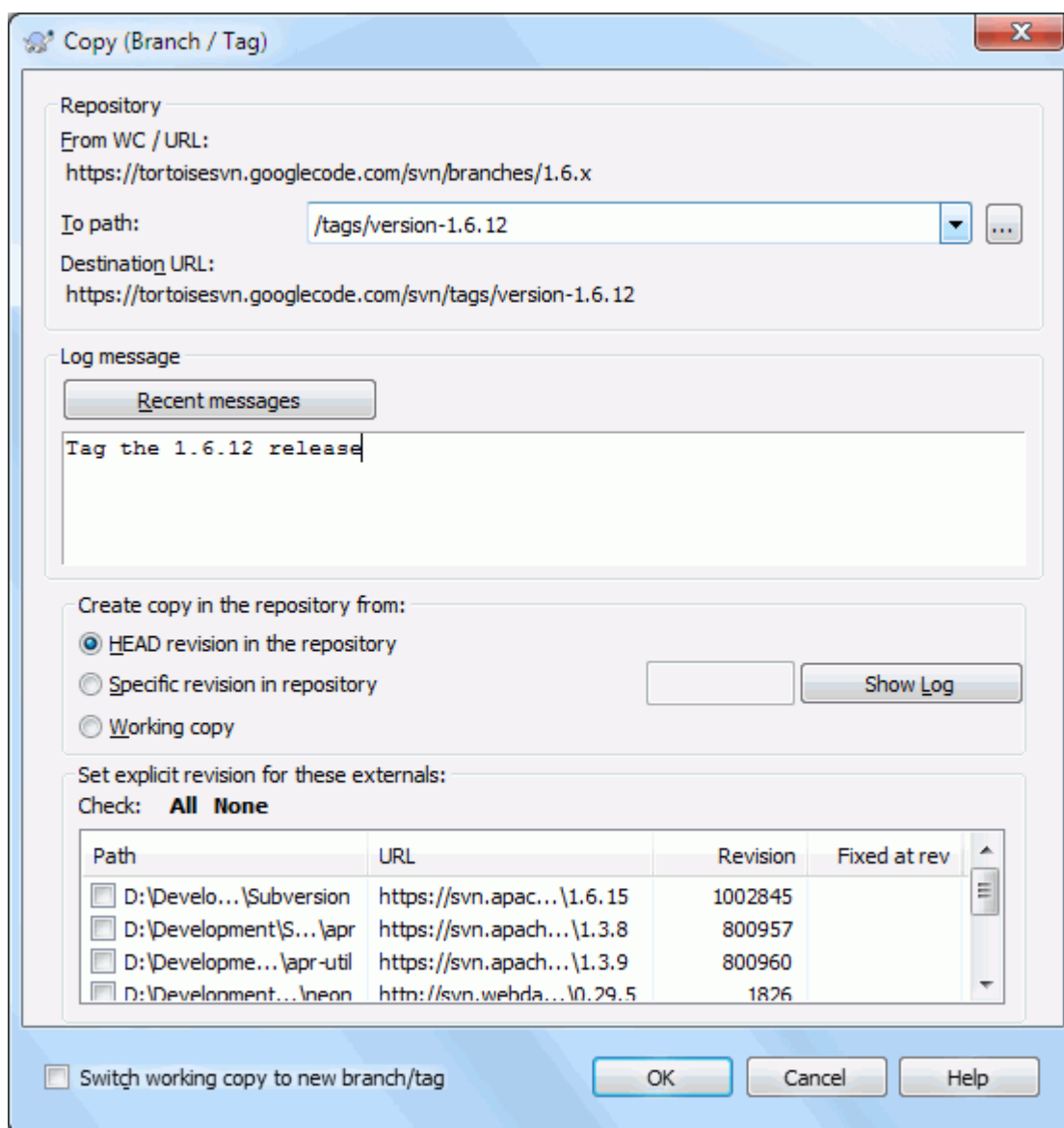
Một trong các tính năng của hệ thống kiểm soát phiên bản là khả năng để cô lập những thay đổi trên một dòng phát triển riêng biệt. Dòng này được biết đến như một *chi nhánh*. Các chi nhánh thường được sử dụng để thử các chức năng mới mà không làm ảnh hưởng đến dòng chính của phát triển với các lỗi trình biên dịch và lỗi chương trình. Ngay sau khi các tính năng mới là đủ ổn định thì nhánh phát triển sẽ được *sáp nhập* trở lại vào các chi nhánh chính (thân cây).

Một tính năng khác của hệ thống kiểm soát phiên bản là khả năng để đánh dấu các phiên bản đặc biệt (ví dụ như một phiên bản phát hành), vì vậy bạn bất cứ lúc nào có thể tái tạo một b xây dựng hoặc môi trường nhất định. Quá trình này được gọi là *gắn thẻ*.

Subversion không có lệnh đặc biệt để phân nhánh hoặc gắn thẻ, nhưng sử dụng cái gọi là “bản sao giá rẻ” thay vào đó. Bản sao giá rẻ tương tự như liên kết cứng trong Unix, có nghĩa rằng thay vì làm một bản sao đầy đủ trong kho, một liên kết nội bộ được tạo ra, chỉ vào một cây / sửa đổi cụ thể. Kết quả là các chi nhánh và thẻ được tạo ra rất nhanh chóng, và hầu như không chiếm thêm không gian trong kho.

### 4.19.1. Tạo một Chi nhánh hoặc Thẻ

Nếu bạn đã nhập khẩu dự án của bạn với cấu trúc thư mục được đề nghị, việc tạo ra một chi nhánh hoặc phiên bản thẻ là rất đơn giản:



**Hình 4.46. Hộp tho Chi nhánh / Thẻ**

Chọn các thư mục trong bản sao làm việc của bạn mà bạn muốn sao chép vào một chi nhánh hoặc thẻ, sau đó chọn lệnh TortoiseSVN → Chi nhánh/Thẻ... .

URL đích mặc định cho chi nhánh mới sẽ là URL nguồn mà trên đó bản sao làm việc của bạn được dựa trên. Bạn sẽ cần phải chỉnh sửa URL đến đường dẫn mới cho chi nhánh / thẻ của bạn. Ví thế thay vì

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

bây giờ bạn có thể sử dụng một cái gì đó như

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

Nếu bạn không thể nhớ quy ước đặt tên bạn sử dụng lần cuối, nhấn vào nút bên phải để mở trình duyệt kho lưu trữ để bạn có thể xem cấu trúc kho lưu trữ hiện có.



## **các thư mục trung gian**

Khi bạn chỉ ra một URL đích, tất cả các thư mục từ đầu đến cuối phải tồn tại sẵn nếu không bạn sẽ gặp thông điệp lỗi. Trong ví dụ phía trên, URL `http://svn.collab.net/repos/ProjectName/tags/` phải tồn tại để tạo thẻ `Release_1.10`.

Tuy nhiên nếu bạn muốn tạo một nhánh/thẻ đến một URL có các thư mục trung gian chưa tồn tại bạn có thể chọn tùy chọn **Tạo các thư mục trung gian ở đây** hộp thoại. Nếu tùy chọn đó được kích hoạt, tất cả thư mục trung gian sẽ tự động được tạo.

Lưu ý rằng tùy chọn này được vô hiệu hóa theo mặc định để tránh lỗi đánh máy. Ví dụ, nếu bạn đánh vào URL là `http://svn.collab.net/repos/ProjectName/Tags/Release_1.10` thay vì `http://svn.collab.net/repos/ProjectName/tags/Release_1.10`, bạn sẽ nhận được một lỗi với tùy chọn bị vô hiệu hóa, nhưng với một tùy chọn có hiệu lực thì một thư mục `Tags` sẽ tự động được tạo ra, và bạn sẽ có một thư mục `Tags` và một thư mục `tags`.

Bây giờ bạn phải chọn nguồn của bản sao. Ở đây bạn có ba lựa chọn:

### **bản sửa đổi CHÍNH trong kho**

Các chi nhánh mới được sao chép trực tiếp trong kho từ bản sửa đổi CHÍNH. Không có dữ liệu cần được chuyển từ bản sao làm việc của bạn, và chi nhánh được tạo ra rất nhanh chóng.

### **Bản sửa đổi cụ thể trong kho**

Chi nhánh mới được sao chép trực tiếp trong kho nhưng bạn có thể chọn một phiên bản cũ hơn. Điều này rất hữu ích nếu bạn quên làm thẻ khi bạn phát hành dự án của bạn vào tuần trước. Nếu bạn không thể nhớ số phiên bản, kích vào nút bên phải để hiển thị tường trình sửa đổi, và chọn số sửa đổi từ đó. Một lần nữa không có dữ liệu được chuyển từ bản sao làm việc của bạn, và chi nhánh được tạo ra rất nhanh chóng.

### **Bản sao làm việc**

Chi nhánh mới là một bản sao giống hệt bản sao làm việc của bạn tại địa phương. Nếu bạn đã cập nhật một số tập tin đến một phiên bản cũ trong WC của bạn, hoặc nếu bạn đã thực hiện thay đổi cục bộ, đó là chính xác những gì đi vào bản sao. Đương nhiên loại thẻ phức tạp này có thể liên quan đến việc chuyển dữ liệu từ WC trở lại kho lưu trữ của bạn nếu nó không tồn tại ở đ.

Nếu bạn muốn bản sao làm việc của bạn được chuyển sang chi nhánh mới được tạo ra tự động, sử dụng hộp kiểm **Chuyển bản sao làm việc cho chi nhánh / thẻ mới**. Nhưng nếu bạn làm điều đó, trước tiên hãy chắc chắn rằng bản sao làm việc của bạn không có chứa sửa đổi. Nếu có, những sửa đổi sẽ được sáp nhập vào các chi nhánh WC khi bạn chuyển đổi.

Nếu bản sao làm việc của bạn có bao gồm các dự án khác với thuộc tính `svn:externals`, những thành phần bên ngoài sẽ được liệt kê ở dưới cùng của hộp thoại chi nhánh / thẻ. Đối với mỗi thành phần bên ngoài, đường dẫn đích, các URL nguồn và sửa đổi được hiển thị. Các sửa đổi của phần bên ngoài được xác định từ bản sao làm việc, có nghĩa là nó cho thấy những bản sửa đổi mà phần bên ngoài thực sự chỉ đến.

Nếu bạn muốn chắc chắn rằng thẻ mới luôn luôn trong trạng thái thống nhất, kiểm tra tất cả các phần bên ngoài để có các bản chỉnh sửa của chúng cố định vào sửa đổi của bản sao làm việc hiện tại của chúng. Nếu bạn không kiểm tra phần bên ngoài và những phần bên ngoài đó chỉ tới một phiên bản CHÍNH mà có thể thay đổi trong tương lai, kiểm xuất thẻ mới sẽ kiểm xuất sửa đổi CHÍNH của phần bên ngoài và thẻ của bạn có thể không biên dịch nữa. Vì vậy, nó luôn luôn là một ý tưởng tốt để đặt các phần bên ngoài tới một bản sửa đổi rõ ràng khi tạo ra một thẻ.

Nếu các phần bên ngoài được đặt đến một bản chỉnh sửa rõ ràng khi tạo ra một chi nhánh hoặc thẻ, TortoiseSVN tự động thay đổi thuộc tính `svn:externals`. Khi chi nhánh / thẻ được tạo ra từ bản CHÍNH hay một bản sửa đổi cụ thể trong kho, TortoiseSVN đầu tiên tạo ra các chi nhánh / thẻ, sau đó điều chỉnh các thuộc tính. Điều

này sẽ tạo ra thêm các cam kết cho mỗi thuộc tính. Khi chi nhánh / thẻ được tạo ra từ bản sao làm việc, các thuộc tính được sửa đổi trước tiên, sau đó các chi nhánh / thẻ được tạo ra và sau đó các thuộc tính được thay đổi trở lại giá trị ban đầu của chúng.

Nhấn **OK** để cam kết bản sao mới vào kho lưu trữ. Đừng quên cung cấp một thông điệp tường trình. Lưu ý rằng bản sao được tạo ra *bên trong các kho lưu trữ* .

Lưu ý rằng trừ khi bạn chọn để chuyển đổi bản sao làm việc của bạn sang chi nhánh mới được tạo ra, việc tạo ra một Chi nhánh hoặc Thẻ *không* ảnh hưởng đến bản sao làm việc của bạn. Ngay cả khi bạn tạo ra các chi nhánh từ WC của bạn, những thay đổi được cam kết vào chi nhánh mới, chứ không phải thân cây, vì vậy WC của bạn vẫn có thể được đánh dấu là sửa đổi đối với thân cây.

#### 4.19.2. Các cách khác để tạo ra một chi nhánh hoặc thẻ

Bạn cũng có thể tạo ra một chi nhánh hoặc thẻ mà không có một bản sao làm việc. Để làm được điều đó, hãy mở trình duyệt kho. Bạn có thể kéo các thư mục đến một vị trí mới. Bạn phải giữ phím **Ctrl** trong khi bạn kéo lên để tạo một bản sao, nếu không thư mục sẽ được di chuyển, chứ không được sao chép.

Bạn cũng có thể kéo một thư mục với nút chuột phải. Một khi bạn thả chuột bạn có thể lựa chọn từ trình đơn ngữ cảnh là bạn muốn thư mục được di chuyển hoặc sao chép. Tất nhiên để tạo ra một chi nhánh hoặc thẻ bạn phải sao chép thư mục, chứ không di chuyển nó.

Tuy nhiên, có một cách khác là từ hộp thoại tường trình. Bạn có thể hiển thị hộp thoại tường trình ví dụ như thân cây, chọn một bản sửa đổi (hoặc là bản sửa đổi CHÍNH ở đầu hoặc một bản sửa đổi trước đó), nhấp chuột phải và chọn **tạo chi nhánh / thẻ từ sửa đổi ...** .

#### 4.19.3. Kiểm xu hoặc Chuyển mạch ...

... Đó (không thực sự) là câu hỏi. Trong khi một kiểm xuất tải xuống mọi thứ từ chi nhánh mong muốn trong kho lưu trữ vào thư mục làm việc của bạn, **TortoiseSVN** → **Chuyển đổi ...** chỉ chuyển dữ liệu được thay đổi vào bản sao làm việc của bạn. Tốt cho tải mạng, tốt cho sự kiên nhẫn của bạn. :-)

Để có thể làm việc với chi nhánh hay thẻ mới được tạo ra bạn có nhiều cách để xử lý nó. Bạn có thể:

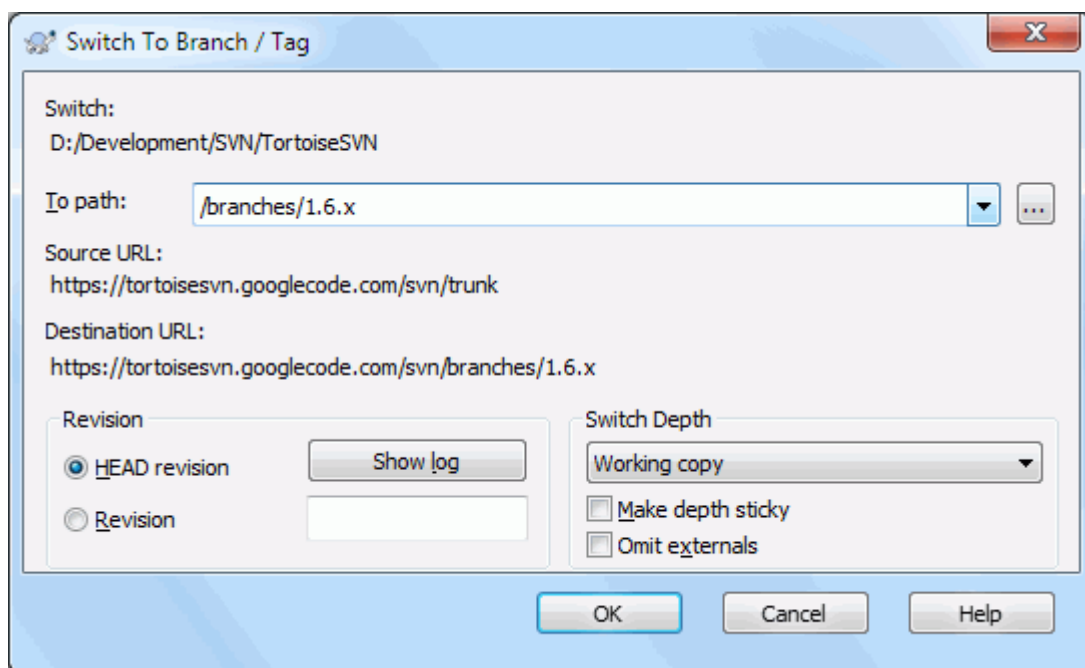
- **TortoiseSVN** → **Kiểm xuất** để làm một kiểm xuất tươi trong một thư mục trống. Bạn có thể kiểm tra bất kỳ vị trí nào trên đĩa địa phương của bạn và bạn có thể tạo nhiều bản sao làm việc từ kho lưu trữ của bạn như bạn muốn.
- Chuyển bản sao làm việc hiện tại của bạn sang bản sao mới được tạo ra trong kho. Một lần nữa chọn thư mục cấp cao nhất của dự án của bạn và sử dụng **TortoiseSVN** → **Chuyển đổi ...** từ trình đơn ngữ cảnh.

Trong hộp thoại tiếp theo, nhập địa chỉ URL của chi nhánh mà bạn vừa tạo ra. Chọn nút radio **Bản sửa đổi Chính** và nhấp chuột vào **OK** . Bản sao làm việc của bạn được chuyển sang chi nhánh / thẻ mới.

Chuyển đổi công việc giống như Cập Nhật ở chỗ nó không bao giờ loại bỏ các thay đổi cục bộ của bạn. Bất kỳ thay đổi bạn đã thực hiện trong bản sao làm việc của bạn chưa được cam kết sẽ được sáp nhập khi bạn làm Chuyển Đổi. Nếu bạn không muốn điều này xảy ra thì bạn phải cam kết những thay đổi trước khi chuyển đổi, hoặc trở lại bản sao của bạn làm việc với một phiên bản đã cam kết (thường là CHÍNH).

- Nếu bạn muốn làm việc trên thân cây và chi nhánh, nhưng không muốn chi phí của một việc kiểm xuất mới, bạn có thể sử dụng Windows Explorer để tạo một bản sao của thân cây của bạn kiểm xuất ra trong một thư mục khác, sau đó **TortoiseSVN** → **Chuyển đổi ...** bản sao đó sang chi nhánh mới của bạn.





**Hình 4.47. Hộp thoại Chuyển ôi**

Mặc dù Subversion chính nó làm cho không có sự phân biệt giữa các thẻ và các nhánh, cách mà chúng được sử dụng thường khác nhau một chút.

- Các thẻ thường được sử dụng để tạo ra một ảnh chụp tĩnh của dự án tại một giai đoạn cụ thể. Bởi vì chúng thường không được sử dụng cho phát triển - đó là những gì các chi nhánh, là lý do chúng tôi đề nghị cấu trúc kho lưu trữ `/trunk/branches/tag` ngay lúc đầu. Làm việc trên một phiên bản thẻ *không phải là một ý tưởng tốt*, Nhưng bởi vì các tập tin địa phương của bạn không được bảo vệ chống ghi nên không có gì để ngăn chặn bạn làm điều này lúc nhầm lẫn. Tuy nhiên, nếu bạn cố gắng cam kết một đường dẫn trong kho, trong đó có `/Tags/`, TortoiseSVN sẽ cảnh báo bạn.
- Có thể rằng bạn cần phải thực hiện thay đổi thêm đến bản phát hành mà bạn đã gắn thẻ. Cách đúng đắn để xử lý việc này là tạo ra một chi nhánh mới từ thẻ đầu tiên và cam kết chi nhánh. Thực hiện các Thay đổi của bạn trên chi nhánh này và sau đó tạo ra một thẻ mới từ chi nhánh mới này, ví dụ như `Version_1.0.1`.
- Nếu bạn chỉnh sửa một bản sao làm việc được tạo ra từ một chi nhánh và cam kết, thì tất cả thay đổi đi đến chi nhánh mới và *không* phải là thân cây. Chỉ có những sửa đổi được lưu trữ. Phần còn lại vẫn còn là một bản sao rê.

## 4.20. Hợp nhất

Trường hợp các chi nhánh được sử dụng để duy trì các dòng phát triển riêng biệt, ở giai đoạn nào đó, bạn sẽ muốn kết hợp các thay đổi được thực hiện trên một chi nhánh trở lại vào thân cây, hoặc ngược lại.

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) [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.

Điểm tiếp theo cần lưu ý là hợp nhất *luôn luôn* diễn ra trong một bản sao làm việc. Nếu bạn muốn kết hợp những thay đổi *vào* một chi nhánh, bạn phải có một bản sao làm việc cho chi nhánh đó kiểm xu, và gọi trình hướng dẫn hợp nhất từ bản sao làm việc bằng cách sử dụng TortoiseSVN → Hợp nhất ...

Nói chung, đó là một ý tưởng tốt để thực hiện một hợp nhất thành một bản sao làm việc chưa sửa đổi. Nếu bạn đã thực hiện thay đổi khác trong WC của bạn, cam kết những cái đó trước. Nếu hợp nhất không như bạn mong đợi, bạn có thể muốn phục hồi các thay đổi, và lệnh Hoàn nguyên sẽ loại bỏ *tất cả* thay đổi bao gồm bất kỳ thay đổi bạn đã thực hiện trước khi hợp nhất.

Có ba trường hợp phổ biến cho việc sáp nhập được xử lý theo những cách hơi khác nhau, như mô tả dưới đây. Trang đầu tiên của hướng dẫn hợp nhất yêu cầu bạn chọn phương pháp nào bạn cần.

#### Hợp nhất một phạm vi các sửa đổi

Phương pháp này bao gồm các trường hợp khi bạn đã thực hiện một hoặc nhiều sửa đổi đến một chi nhánh (hoặc thân cây) và bạn muốn chuyển đổi các thay đổi trên đến một chi nhánh khác nhau.

Những gì bạn đang yêu cầu Subversion làm là: “ Tính toán những thay đổi cần thiết để có được [TỪ] phiên bản 1 của chi nhánh A [SANG] sửa đổi 7 chi nhánh A, và áp dụng những thay đổi để sao chép các thay đổi đó sang bản làm việc của tôi (của thân cây hoặc chi nhánh B). ”

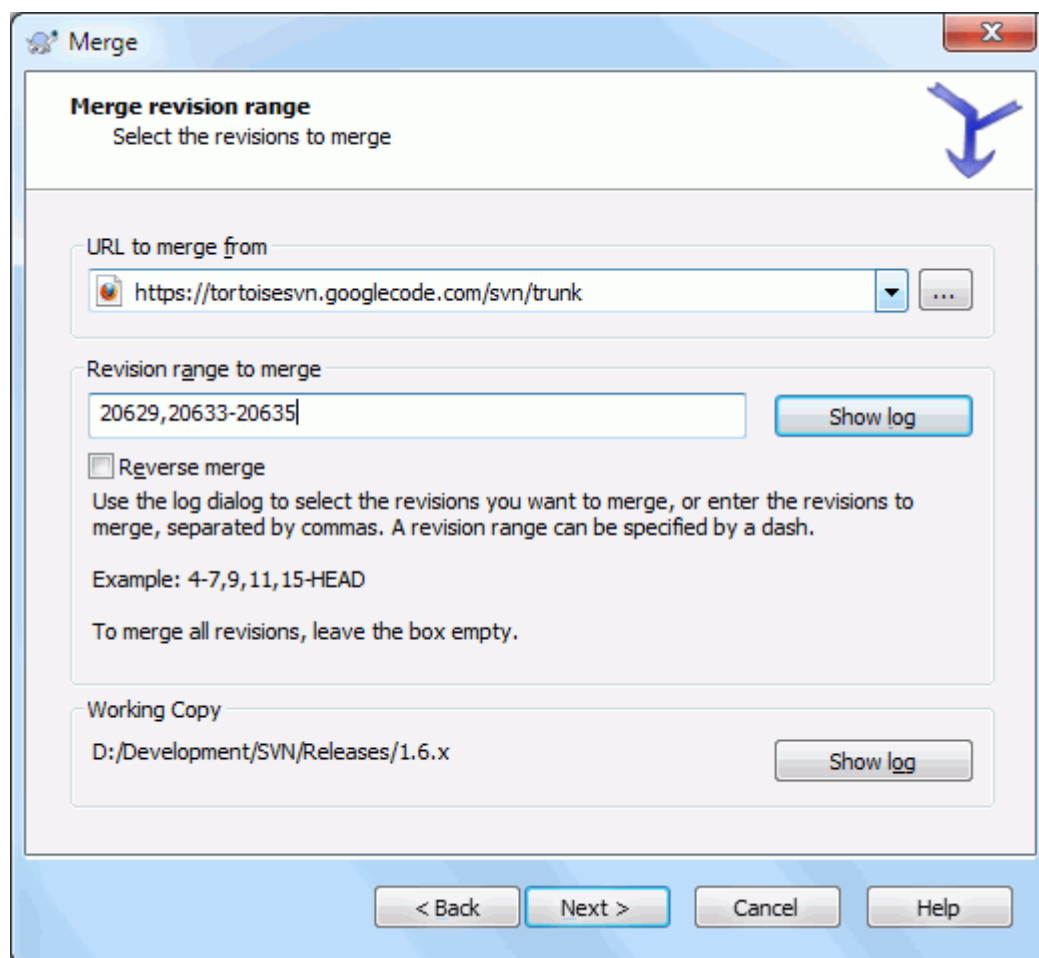
If you leave the revision range empty, Subversion uses the merge-tracking features to calculate the correct revision range to use. This is known as a reintegrate or automatic merge.

#### Hợp nhất hai cây khác nhau

Đây là trường hợp tổng quát hơn của phương pháp tái tích hợp. Những gì bạn đang yêu cầu Subversion làm là: “ Tính toán những thay đổi cần thiết để có được [TỪ] bản sửa chính của thân cây [ĐẾN] bản sửa đổi chính của chi nhánh, và áp dụng những thay đổi đó tới bản sao làm việc của tôi (của thân cây). ” Kết quả là thân cây giờ đây trông giống hệt như chi nhánh.

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.

### 4.20.1. Hợp nhất một Phạm vi Các sửa đổi



Hình 4.48. Tình huống hướng dẫn Hợp nh - Chọn Phạm vi Sửa đổi

Trong mục **Từ**: nhập vào URL thư mục đầy đủ của chi nhánh hoặc thẻ có chứa các thay đổi mà bạn đưa đến bản sao làm việc của bạn. Bạn cũng có thể nhấp chuột vào ... để duyệt qua các kho lưu trữ và tìm những chi nhánh mong muốn. Nếu bạn đã sập nhập từ chi nhánh này trước đó, thì chỉ cần sử dụng danh sách thả xuống cho thấy một lịch sử của các URL được sử dụng trước đó.

Nếu bạn đang hợp nhất từ một nhánh được đổi tên hoặc bị xóa thì bạn sẽ phải đi ngược đến bản chỉnh sửa mà nhánh đó vẫn tồn tại. Trong trường hợp này bạn cũng sẽ phải chỉ định bản chỉnh sửa như một bản mốc trong phạm vi các bản chỉnh sửa đang được hợp nhất (xem phía dưới), nếu không việc hợp nhất sẽ thất bại khi nó không thể tìm được đường dẫn đó tại bản CHÍNH.

Trong trường **Phạm vi sửa đổi để hợp nhất** nhập vào danh sách các phiên bản bạn muốn kết hợp. Đây có thể là một bản sửa đổi duy nhất, một danh sách các bản sửa đổi cụ thể cách nhau bằng dấu phẩy, hoặc một loạt các bản sửa đổi được phân cách bởi một dấu gạch ngang, hoặc bất kỳ sự kết hợp của những cái đó.

Nếu bạn cần phải chỉ định một bản sửa đổi peg cho hợp nhất, bổ sung sửa đổi peg vào cuối của các phiên bản, ví dụ như 5-7, 10@3 . Trong ví dụ trên, các phiên bản 5,6,7 và 10 sẽ được sập nhập, với 3 là bản sửa đổi peg.



## Quan trọng

Có một sự khác biệt quan trọng trong cách một phạm vi sửa đổi được chỉ ra với TortoiseSVN so với các dòng lệnh máy khách. Cách dễ nhất để hình dung nó là nghĩ đến một hàng rào với đôn và tấm hàng rào.

Với các dòng lệnh máy khách bạn chỉ ra các thay đổi để hợp nhất sử dụng hai bản sửa đổi “ đôn hàng rào ” mà chỉ ra các điểm *trước khi* và *sau khi* .

Với TortoiseSVN, bạn chỉ định tập hợp thay đổi để hợp nhất sử dụng “ hàng rào tấm ” . Lý do cho việc này trở nên rõ ràng khi bạn sử dụng hộp thoại tương trình để xác định các sửa đổi để sập nhập, khi mỗi b sửa đổi xuất hiện như một tập hợp thay đổi.

Nếu bạn đang hợp nhất các bản chỉnh sửa theo các mẫu lớn, phương pháp chỉ ra trong sách Subversion sẽ đòi hỏi bạn hợp nhất 100-200 lần này và 200-300 vào lần tới. Với TortoiseSVN bạn sẽ hợp nhất 100-200 vào lần này và 201-300 vào lần tới.

Sự khác biệt này đã tạo ra rất nhiều sự nóng bỏng trên các danh sách gửi thư. Chúng tôi thừa nhận rằng có một sự khác biệt từ các dòng lệnh máy khách, nhưng chúng tôi tin rằng đối với phần lớn người sử dụng giao diện đồ họa để sẽ dàng hơn để hiểu được phương pháp mà chúng tôi đã thực hiện.

Cách dễ nhất để chọn phạm vi các bản sửa đổi bạn cần là bấm vào **Hiện Tường Trình** , vì điều này sẽ liệt kê những thay đổi gần đây với lời bình tường trình của họ. Nếu bạn muốn kết hợp các thay đổi từ một phiên bản duy nhất, chỉ cần chọn sửa đổi đó. Nếu bạn muốn kết hợp thay đổi từ một số sửa đổi, thì chọn phạm vi đó (bằng cách sử dụng phím **Shift** -modifier). Nhấp vào **OK** và danh sách các số sửa đổi để hợp nhất sẽ được điền vào cho bạn.

Nếu bạn muốn kết hợp thay đổi trở lại *ra khỏi* bản sao làm việc của bạn, để quay lại một sự thay đổi đã được cam kết, lựa chọn các phiên bản để trở lại và chắc chắn rằng hộp kiểm **Đảo ngược hợp nhất** được ch.

Nếu bạn đã sập nhập một số thay đổi từ chi nhánh này, hy vọng là bạn sẽ để lại lưu ý về việc sửa đổi cuối cùng được sập nhập trong thông điệp tường trình khi bạn cam kết thay đổi. Trong trường hợp đó, bạn có thể sử dụng **Hiện Tường Trình** cho Bản Sao Làm Việc để theo dõi thông điệp tường trình. Nhớ rằng chúng ta đang nghĩ đến các bản sửa đổi như các tập hợp thay đổi, bạn Nên sử dụng bản sửa đổi sau khi các điểm cuối của việc hợp nhất cuối cùng như là điểm bắt đầu cho việc hợp nhất này. Ví dụ, nếu bạn đã hợp nhất các phiên bản 37-39 lần cuối cùng, thì các điểm bắt đầu cho hợp nhất này nên là sửa đổi 40.

Nếu bạn đang sử dụng tính năng theo dõi hợp nhất của Subversion, bạn không cần phải nhớ các bản sửa đổi đã được sập nhập - Subversion sẽ ghi lại điều đó cho bạn. Nếu bạn bỏ trống phạm vi sửa đổi, tất cả các phiên bản chưa được sập nhập sẽ được bao gồm. Đọc **Phần 4.20.5, “Theo dõi hợp nh”** để tìm hiểu thêm.

Khi theo dõi hợp nhất được sử dụng, hộp thoại tương trình sẽ hiển thị các bản sửa đổi hợp nhất trước đây, và các bản sửa đổi có ngày tháng trước điểm tổ tiên chung, tức là trước khi chi nhánh đã được sao chép, như đã được

làm xám. Hộp kiểm **Ấn sửa đổi không hợp nhất** được cho phép bạn lọc ra những bản sửa đổi hoàn toàn để bạn chỉ nhìn thấy các bản sửa đổi *có thể* được hợp nhất.

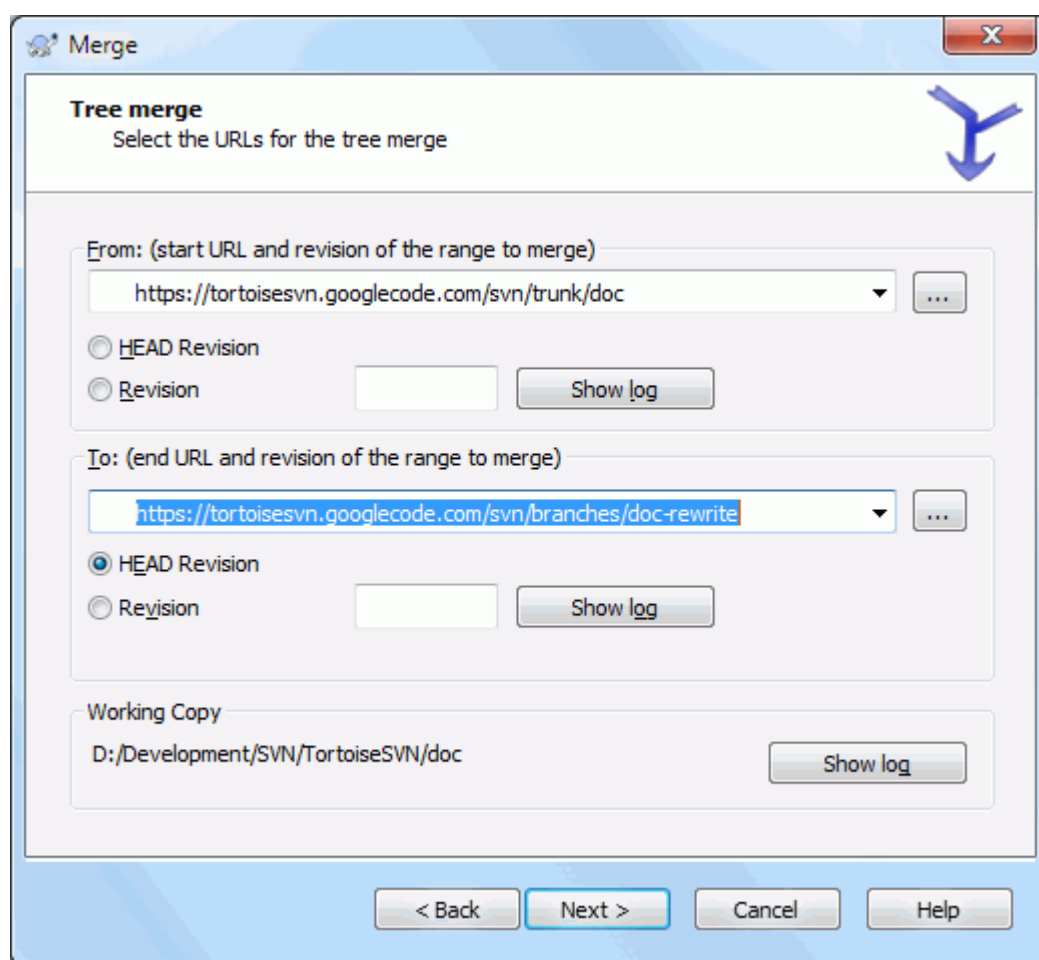
Nếu người khác có lẽ đang cam kết các thay đổi thì hãy cẩn thận về việc sử dụng bản sửa đổi CHÍNH. Nó có thể không đề cập đến bản sửa đổi bạn nghĩ nếu một người nào khác đã thực hiện một cam kết sau lần cập nhật cuối cùng của bạn.

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.

Có một số điều kiện áp dụng đối với một hợp nhất tái tích hợp. Thứ nhất, các máy chủ phải hỗ trợ theo dõi kết hợp. Bản sao làm việc phải có chiều sâu vô hạn (không có kiểm xuất thừa thớt), và nó không được có bất kỳ thay đổi địa phương, hạng mục bị chuyển hay hạng mục đã được cập nhật vào các bản sửa đổi khác không phải bản CHÍNH. Mọi thay đổi vào thân cây được thực hiện trong quá trình phát triển chi nhánh phải được sáp nhập qua chi nhánh (hoặc đánh dấu là đã được sáp nhập). Phạm vi các bản sửa đổi để hợp nhất sẽ được tính toán tự động.

Nhấp vào **Tiếp theo** và đi đến **Phần 4.20.3, “Tùy Chọn Hợp Nhất”**.

## 4.20.2. Việc Sáp Nhập Hai Cây Khác Nhau



**Hình 4.49. Trình hướng dẫn sáp nh - Sáp nhập cây**

Nếu bạn đang sử dụng phương pháp này để hợp nhất một chi nhánh tính năng quay trở lại với thân cây, bạn cần phải chạy trình hướng dẫn hợp nhất từ bên trong một bản sao hoạt động của thân cây.

Trong trường **Từ:** nhập vào URL thư mục đầy đủ của *thân cây*. Điều này nghe có vẻ sai, nhưng hãy nhớ rằng thân cây là điểm bắt đầu mà bạn muốn thêm vào những thay đổi chi nhánh. Bạn cũng có thể nhấp chuột vào ... để duyệt qua kho lưu trữ.

Trong trường **Đến**: nhập vào URL thư mục đầy đủ của chi nhánh tính năng.

Trong cả hai trường **Từ Bản Sửa Đổi** và **Đến Bản Sửa Đổi**, nhập số phiên bản cuối cùng mà hai cây đã được đồng bộ. Nếu bạn chắc chắn không ai khác đã thực hiện cam kết, bạn có thể sử dụng bản sửa đổi **CHÍNH** trong cả hai trường hợp. Nếu có một khả năng mà một người nào đó có thể thực hiện một cam kết kể từ khi đồng bộ hóa, sử dụng số phiên bản cụ thể để tránh mất các cam kết gần đây.

Bạn cũng có thể sử dụng **Hiện Trường Tr** để lựa chọn sửa đổi.

### 4.20.3. Tùy Chọn Hợp Nhất

Trang này của trình hướng dẫn cho phép bạn chỉ định các tùy chọn nâng cao, trước khi bắt đầu quá trình hợp nhất. Hầu hết thời gian bạn có thể chỉ sử dụng các thiết lập mặc định.

Bạn có thể chỉ định chiều sâu để sử dụng cho hợp nhất, tức là hợp nhất nên đi sâu như thế nào xuống bản sao làm việc của bạn. Các điều khoản độ sâu sử dụng được mô tả trong **Phần 4.3.1, “Độ sâu Kiểm xu”**. Độ sâu mặc định là **Bản sao làm việc**, trong đó sử dụng các thiết lập độ sâu hiện tại, và hầu như luôn luôn là những gì bạn muốn.

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>].

Bạn có thể chỉ định cách mà dòng kết thúc và thay đổi khoảng trắng được xử lý. Các tùy chọn này được mô tả trong **Phần 4.10.2, “Tùy chọn cho Hết-Dòng và Khoảng Trắng”**. Các hành vi mặc định là để điều trị tất cả các sự khác biệt khoảng trắng và cuối dòng như thay đổi thực sự cần được sáp nhập.

Hộp kiểm được đánh dấu **Buộc thực hiện hợp nhất** được sử dụng để tránh một cuộc xung đột cây khi một việc xóa ảnh hưởng đến một tập tin được sửa đổi tại địa phương hay hoàn toàn không được đánh phiên bản. Nếu tập tin này bị xóa thì không có cách nào để khôi phục lại nó, đó là lý do tại sao tùy chọn này không được chọn theo mặc định.

Nếu bạn đang sử dụng theo dõi kết hợp và bạn muốn đánh dấu một phiên bản đã được sáp nhập, mà không thực sự làm việc hợp nhất ở đây, kiểm tra hộp kiểm **Chỉ ghi lại các hợp nhất**. Có hai lý do có thể bạn có thể muốn làm điều này. Nó có thể là hợp nhất như vậy là quá phức tạp đối với các thuật toán hợp nhất, vì vậy bạn viết mã thay đổi bằng tay, sau đó đánh dấu sự thay đổi như sáp nhập để các thuật toán theo dõi hợp nhất nhận thức được nó. Hoặc bạn có thể muốn ngăn chặn một phiên bản nào đó khỏi bị sáp nhập. Đánh dấu nó như đã được sáp nhập sẽ ngăn chặn việc hợp nhất xảy ra với các chương trình khách biệt theo dõi kết hợp.

Bây giờ tất cả mọi thứ được thiết lập, tất cả những gì bạn phải làm là nhấp vào nút **Hợp nhất**. Nếu bạn muốn xem trước kết quả **Kiểm Tra Hợp Nhất** mô phỏng các hoạt động hợp nhất, nhưng hoàn toàn *không* sửa đổi các bản sao làm việc. Nó cho bạn thấy một danh sách các tập tin mà sẽ được thay đổi bằng một hợp nhất thực sự, và ghi chú các tập tin mà mâu thuẫn *có thể* xảy ra. Do việc theo dõi kết hợp làm cho quá trình hợp nhất phức tạp hơn rất nhiều, không có cách nào bảo đảm có thể tìm hiểu trước xem việc hợp nhất sẽ hoàn thành mà không có xung đột, do đó, các tập tin đánh dấu là mâu thuẫn trong một kiểm tra hợp nhất về thực tế có thể hợp nhất mà không có bất kỳ vấn đề.

Hộp thoại tiến triển hợp nhất cho thấy mỗi giai đoạn của hợp nhất, với phạm vi sửa đổi có liên quan. Điều này có thể chỉ ra một hay nhiều sửa đổi hơn bạn mong đợi. Ví dụ, nếu bạn yêu cầu kết hợp sửa đổi 123 hộp thoại tiến triển sẽ báo cáo “Đang hợp nhất các bản sửa đổi 122 tới 123”. Để hiểu được điều này, bạn cần phải nhớ rằng Hợp Nhất liên quan chặt chẽ với Khác Biệt. Quá trình hợp nhất hoạt động bằng cách tạo ra một danh sách các sự khác biệt giữa hai điểm trong kho, và áp dụng những sự khác biệt vào bản sao làm việc của bạn. Hộp thoại tiến triển chỉ đơn giản là hiển thị các điểm bắt đầu và kết thúc cho các khác bi.

#### 4.20.4. Rà soát các Kết quả Hợp nhất

Hợp nhất bây giờ là hoàn tất. Đó là một ý tưởng tốt để có một cái nhìn vào việc hợp nhất và xem nếu nó có như mong đợi không. Hợp nhất thường là khá phức tạp. Xung đột thường xảy ra nếu chi nhánh đã trôi dạt xa thân cây.



##### Mẹo

Khi các bản sửa đổi được hợp nhất vào bản làm việc, TortoiseSVN tạo ra một thông điệp tường trình từ các bản chỉnh sửa được hợp nhất. Những bản đó được cung cấp từ nút bấm Các Thông Điệp Gần Đây trong hộp thoại cam kết.

Để tùy chỉnh thông điệp được sinh ra, đặt các thuộc tính dự án trên bản làm việc của bạn. Xem [Phần 4.17.3.10, “Các mẫu thông điệp hợp nhất”](#)

Đối với các máy khách và máy chủ Subversion trước 1.5, không có thông tin sáp nhập nào được lưu trữ và sửa đổi được sáp nhập phải được theo dõi bằng tay. Khi bạn đã thử nghiệm những thay đổi và đi đến cam kết sửa đổi này, thông điệp tường trình cam kết của bạn nên *luôn luôn* bao gồm các số sửa đổi đã được chuyển trong hợp nhất. Nếu bạn muốn áp dụng một hợp nhất tại một thời gian sau đó bạn sẽ cần phải biết những gì bạn đã sáp nhập, bởi bạn không muốn chuyển đổi một thay đổi nhiều hơn một lần. Để biết thêm thông tin về điều này, hãy tham khảo [Thực tiễn Tốt nhất cho sáp nhập](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] trong cuốn sách Subversion.

Nếu máy chủ của bạn và tất cả các máy khách đang chạy Subversion 1.5 hoặc cao hơn, hỗ trợ theo dõi kết hợp sẽ ghi lại các bản sửa đổi được sáp nhập và tránh việc một bản sửa đổi sáp nhập nhiều hơn một lần. Điều này làm cho cuộc sống của bạn đơn giản hơn nhiều khi bạn chỉ đơn giản là có thể hợp nhất toàn bộ phạm vi sửa đổi mỗi lần và biết rằng chỉ phiên bản mới sẽ thực sự được sáp nhập.

Quản lý chi nhánh là quan trọng. Nếu bạn muốn giữ cho chi nhánh này được cập nhật với thân cây, bạn nên chắc chắn là kết hợp thường xuyên để chi nhánh và thân cây không trôi quá xa nhau. Tất nhiên, bạn vẫn nên tránh lặp đi lặp lại kết hợp những thay đổi, như được giải thích ở trên.



##### Mẹo

Nếu bạn đã sáp nhập một chi nhánh tính năng trở lại vào thân cây, thân cây giờ đây có chứa tất cả các mã tính năng mới, và chi nhánh trở nên lỗi thời. Bạn có thể xóa nó từ kho nếu cần thiết.



##### Quan trọng

Subversion không thể hợp nhất một tập tin với một thư mục và ngược lại - chỉ thư mục vào các thư mục và tập tin vào các tập tin. Nếu bạn nhấp chuột vào một tập tin và mở hộp thoại hợp nhất, thì bạn phải cung cấp một đường dẫn đến một tập tin trong hộp thoại đó. Nếu bạn chọn một thư mục và đưa lên hộp thoại, sau đó bạn phải chỉ định một URL thư mục để hợp nhất.

#### 4.20.5. Theo dõi hợp nh

Subversion 1.5 giới thiệu các hỗ trợ cho theo dõi hợp nhất. Khi bạn hợp nhất các thay đổi từ một cây khác, những con số sửa đổi sáp nhập được lưu trữ và thông tin này có thể được sử dụng cho nhiều mục đích khác nhau.

- Bạn có thể tránh nguy hiểm của việc hợp nhất cùng một sửa đổi hai lần (lặp đi lặp lại vấn đề kết hợp). Một khi sửa đổi được đánh dấu như đã được hợp nhất, các sáp nhập trong tương lai trong đó bao gồm sửa đổi trong phạm vi sẽ bỏ qua nó.
- Khi bạn hợp nhất một chi nhánh trở lại vào thân cây, hộp thoại tường trình có thể cho bạn thấy việc cam kết chi nhánh như một phần của tường trình thân cây, cho phép theo dõi tốt hơn những thay đổi.
- Khi bạn hiển thị hộp thoại tường trình từ bên trong hộp thoại hợp nhất, các sửa đổi đã được sáp nhập được thể hiện trong màu xám.

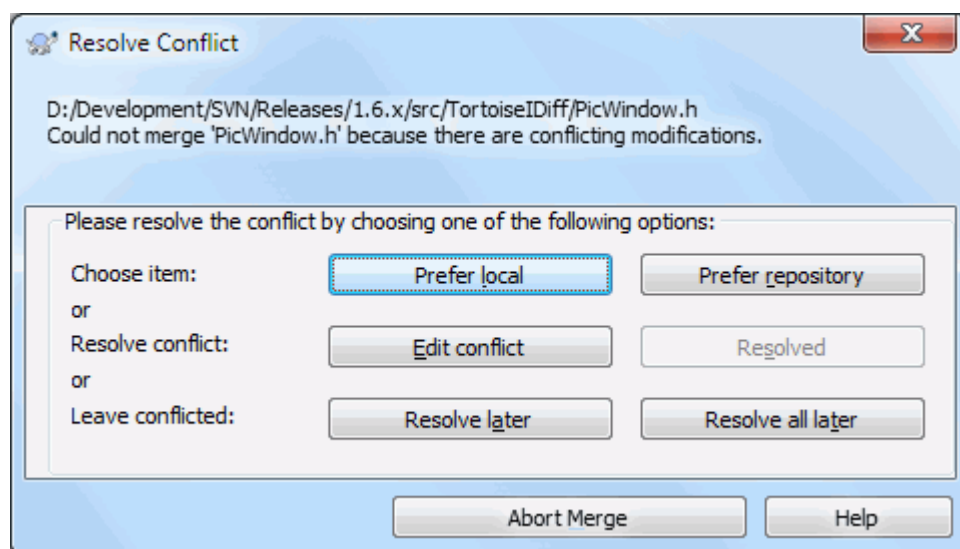
- Khi hiển thị thông tin đồ lỗi cho một tập tin, bạn có thể chọn hiển thị các tác giả ban đầu của các phiên bản sáp nhập, chứ không phải là người đã thực hiện hợp nhất.
- Bạn có thể đánh dấu các phiên bản như *đừng hợp nhất* bằng cách bao gồm chúng trong danh sách các phiên bản sáp nhập mà không thực sự làm việc hợp nhất.

Thông tin theo dõi hợp nhất được lưu trữ trong thuộc tính `svn:mergeinfo` của máy khách khi thực hiện một hợp nhất. Khi hợp nhất được cam kết máy chủ lưu thông tin đó trong một cơ sở dữ liệu, và khi bạn yêu cầu thông tin hợp nhất, trình trình hoặc đồ lỗi, máy chủ có thể trả lời một cách thích hợp. Để cho hệ thống làm việc đúng cách, bạn phải đảm bảo rằng các máy chủ, kho lưu trữ và tất cả các máy khách được nâng cấp. Các máy khách cũ hơn sẽ không lưu trữ thuộc tính `svn:mergeinfo` và máy chủ cũ hơn sẽ không cung cấp các thông tin theo yêu cầu của máy khách mới.

Tìm hiểu thêm về theo dõi hợp nhất từ Subversion [Tài liệu về theo dõi việc hợp nhất](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].

#### 4.20.6. Xử Lý Xung Đột trong Hợp Nhất

Hợp nhất không phải luôn luôn diễn ra suôn sẻ. Đôi khi có xung đột, và nếu bạn hợp nhất nhiều phạm vi, bạn thường muốn giải quyết xung đột trước khi việc hợp nhất phạm vi tiếp theo bắt đầu. TortoiseSVN giúp bạn thông qua quá trình này bằng cách hiển thị hộp thoại *gọi lại khi hợp nhất xung đột*



**Hình 4.50. Hộp Thoại Gọi Lại Xung Đột Hợp Nhất**

Có khả năng rằng một số những thay đổi sẽ hợp nhất trơn tru, trong khi các thay đổi địa phương khác xung đột với những thay đổi đã cam kết vào kho lưu trữ. Tất cả các thay đổi có thể được sáp nhập được sáp nhập. Hộp Thoại Gọi Lại khi Xung đột cho bạn ba cách khác nhau để xử lý các dòng có xung đột.

1. Nếu hợp nhất của bạn bao gồm các tập tin nhị phân, kết hợp các cuộc xung đột đó là không thể. Bạn phải lựa chọn một tập tin hoàn chỉnh. Sử dụng *Ưu tiên địa phương* để chọn phiên bản địa phương như trong bản sao làm việc của bạn trước khi hợp nhất, hoặc *Ưu tiên kho* để chọn các tập tin đến từ các nguồn hợp nhất trong kho.

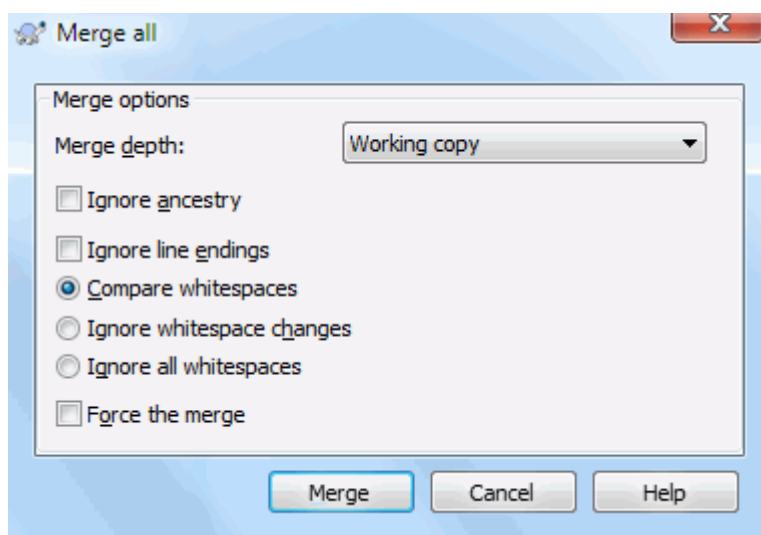
Nếu bạn là sáp nhập các tập tin văn bản thì hai nút đầu tiên này cho phép bạn hợp nhất các dòng không mâu thuẫn như bình thường và luôn luôn ưu tiên một phiên bản nơi có xung đột. Chọn *Ưu tiên địa phương* sẽ lựa chọn phiên bản địa phương của bạn trong mọi xung đột, tức là nó sẽ ưu tiên những gì đã có trước khi hợp nhất so với thay đổi đến từ các nguồn hợp nhất. Tương tự, *Ưu tiên kho* sẽ lựa chọn những thay đổi kho lưu trữ trong mọi xung đột, tức là nó sẽ ưu tiên những thay đổi đến từ nguồn kết hợp sơ với những gì đã có trong bản sao làm việc của bạn. Điều này nghe có vẻ dễ dàng, nhưng những xung đột thường trải trên nhiều dòng hơn bạn nghĩ và bạn có thể nhận được kết quả bất ngờ.

- Thông thường bạn sẽ muốn nhìn vào những xung đột và tự mình giải quyết chúng. Trong trường hợp đó, chọn **Biên tập Xung đột** sẽ chạy công cụ hợp nhất của bạn. Khi bạn đã hài lòng với kết quả, nhấp chuột vào **Giải quyết**.
- Tùy chọn cuối cùng là trì hoãn giải quyết và tiếp tục với việc sáp nhập. Bạn có thể chọn để làm điều đó cho các tập tin đang xung đột hiện tại, hoặc cho tất cả các file trong phần còn lại của hợp nhất. Tuy nhiên, nếu có các thay đổi xa hơn trong tập tin đó, việc hoàn thành hợp nhất là không thể.

Nếu bạn không muốn sử dụng tương tác gọi lại này, có một hộp kiểm trong hộp thoại tiến triển hợp nhất **Hợp nhất không tương tác**. Nếu đây là thiết lập cho một hợp nhất và hợp nhất sẽ cho kết quả trong một xung đột, tập tin được đánh dấu xung đột và hợp nhất sẽ đi tiếp. Bạn sẽ phải giải quyết những xung đột sau khi toàn bộ hợp nhất được hoàn thành. Nếu nó không được thiết lập, thì trước khi một tập tin được đánh dấu là mâu thuẫn bạn có cơ hội để giải quyết xung đột *trong lúc* hợp nhất. Điều này có lợi thế rằng nếu một tập tin có nhiều hợp nhất (nhiều bản sửa đổi áp dụng một thay đổi đến tập tin đó), các hợp nhất tiếp theo có thể thành công tùy thuộc vào những dòng bị ảnh hưởng. Nhưng tất nhiên, bạn không thể chỗ khác để lấy cả phê trong khi hợp nhất đang chạy.)

#### 4.20.7. Hợp nhất một Chi nhánh Đầy đủ

Nếu bạn muốn hợp nhất tất cả các thay đổi từ một chi nhánh tính năng vào thân cây, thì bạn có thể sử dụng TortoiseSVN → **Tái hòa nhập hợp nhất...** từ trình đơn ngữ cảnh mở rộng (giữ phím **Shift** trong khi bạn kích chuột phải vào tập tin).



**Hình 4.51. Hộp thoại tái tích hợp sáp nh**

Hộp thoại này rất dễ dàng. Tất cả những gì bạn phải làm là thiết lập các tùy chọn cho hợp nhất, theo mô tả trong [Phần 4.20.3, “Tùy Chọn Hợp Nhất”](#). Phần còn lại được thực hiện bởi TortoiseSVN một cách tự động bằng cách sử dụng theo dõi hợp nh.

#### 4.20.8. Bảo trì các Chi nhánh Tính năng

Khi bạn phát triển một tính năng mới trên một nhánh riêng biệt, nó là một ý tưởng tốt để làm ra một chính sách tái tích hợp khi tính năng được hoàn tất. Nếu công việc khác đang xảy đồng thời ra trong *thân cây* bạn có thể thấy sự khác biệt đáng kể theo thời gian, và hợp nhất trở lại biến thành một cơn ác mộng.

Nếu tính năng là tương đối đơn giản và phát triển sẽ không mất nhiều thời gian thì bạn có thể áp dụng một phương pháp đơn giản, để giữ cho chi nhánh hoàn toàn riêng biệt cho đến khi tính năng này hoàn tất, sau đó hợp nhất các thay đổi chi nhánh trở lại vào thân cây. Trong trình hướng dẫn hợp nhất việc này sẽ đơn giản là **Hợp nhất một loạt các sửa đổi**, với phạm vi sửa đổi là khoảng sửa đổi của chi nhánh.



Nếu tính năng này sẽ mất nhiều thời gian và bạn cần vào tài khoản cho những thay đổi trong thân cây, thì bạn cần phải giữ cho chi nhánh đồng bộ. Điều này đơn giản chỉ có nghĩa là định kỳ, bạn hợp nhất các thay đổi thân cây vào chi nhánh, do đó chi nhánh có chứa tất cả các thay đổi thân *cộng thêm* tính năng mới. Quá trình đồng bộ sử dụng **Hợp nhất một loạt các sửa đổi**. Khi tính năng hoàn tất, sau đó bạn có thể kết hợp nó trở lại thân cây bằng cách sử dụng **Tái tích hợp một chi nhánh** hoặc **Hợp nhất hai cây khác nhau**.

## 4.21. Khóa

Subversion nói chung hoạt động tốt nhất mà không cần khóa, sử dụng phương pháp “Sao Chép-Sửa Đổi-Hợp Nhất” mô tả trước đây trong **Phần 2.2.3, “Giải pháp Sao chép-Sửa đổi-Hợp nhất”**. Tuy nhiên có một số ít trường hợp khi bạn có thể cần phải thực hiện một số hình thức của chính sách khóa.

- Bạn đang sử dụng các tập tin “không hợp nhất được”, ví dụ, các tập tin đồ họa. Nếu hai người thay đổi cùng một tập tin, sáp nhập là không thể, vì vậy một trong các bạn sẽ mất các thay đổi của họ.
- Công ty của bạn đã luôn luôn sử dụng một hệ thống kiểm soát khóa sửa đổi trong quá khứ và đã có một quyết định quản lý mà “khóa là tốt nhất”.

Trước hết bạn cần đảm bảo rằng máy chủ Subversion của bạn được nâng cấp lên phiên bản 1.2. Phiên bản trước đó hoàn toàn không hỗ trợ khóa. Nếu bạn đang sử dụng truy cập `file://`, thì tất nhiên máy khách của bạn cần được cập nhật.



### Ba ý nghĩa của “Khóa”

Trong đoạn này, và hầu như bất kỳ đâu trong cuốn sách này, các từ “khóa” và “mở khóa” mô tả cách thức loại lẫn nhau giữa những người dùng để tránh đụng độ các cam kết. Thật không may, có hai loại “khóa” mà Subversion, và do đó trong cuốn sách này, thỉnh thoảng cần được quan tâm.

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 **Phần 4.16, “Dọn sạch”**.

Và thứ ba, các tập tin và thư mục có thể được khóa nếu chúng đang được sử dụng bởi các tiến trình khác, ví dụ nếu bạn có một tài liệu word mở trong Word, tập tin đó bị khóa và không thể truy cập được bởi TortoiseSVN.

Bạn thường có thể quên những loại khóa này cho tới khi có chuyện gì đó sai xảy ra đòi hỏi bạn phải để ý đến chúng. Trong cuốn sách này, “khóa” có nghĩa loại đầu tiên trừ khi điều trái ngược là rõ ràng từ ngữ cảnh hoặc được tuyên bố rõ ràng.

### 4.21.1. Khóa làm việc thế nào trong Subversion

Theo mặc định, không có gì là bị khóa và bất cứ ai có quyền truy cập cam kết có thể cam kết những thay đổi ở bất kỳ tập tin vào bất cứ lúc nào. Những người khác sẽ cập nhật bản sao làm việc của họ định kỳ và thay đổi trong kho sẽ được sáp nhập với những thay đổi địa phương.

Nếu bạn *Nhận được một khóa* trên một tập tin, thì chỉ có bạn có thể cam kết tập tin đó. Cam kết của tất cả các người dùng khác sẽ bị chặn cho đến khi bạn phát hành khóa. Một tập tin bị khóa không thể được sửa đổi trong bất kỳ cách nào trong kho, vì vậy nó không thể bị xóa hay đổi tên hoặc, ngoại trừ b chủ sở hữu khóa.



### Quan trọng

Khóa là không được giao cho một người dùng cụ thể, nhưng cho một người dùng cụ thể và một bản sao làm việc. Có một khóa trong một bản sao làm việc cũng ngăn cản cùng người sử dụng đó cam kết các tập tin bị khóa từ một bản sao làm việc khác.

Ví dụ, hãy tưởng tượng rằng người sử dụng Jon có một bản sao làm việc trên máy tính văn phòng của ông. Ở đó, ông bắt đầu làm việc trên một hình ảnh, và do đó có được một khóa trên tập tin đó. Khi ông rời khỏi văn phòng của ông, ông chưa xong với tập tin đó, vì vậy ông ta không giải phóng khóa. Trở lại nhà Jon cũng có một bản sao làm việc và quyết định làm việc nhiều hơn một chút về dự án. Tuy nhiên, ông không có thể sửa đổi hoặc cam kết cùng tập tin hình ảnh đó, bởi vì các khóa cho tập tin đó đang nằm trong bản sao làm việc trong văn phòng.

Tuy nhiên, những người dùng khác sẽ không nhất thiết phải biết rằng bạn đã lấy khóa ra. Trừ khi họ kiểm tra tình trạng khóa thường xuyên, việc đầu tiên mà họ sẽ biết về nó là khi cam kết của họ thất bại, mà trong nhiều trường hợp không phải là rất hữu ích. Để làm cho nó dễ dàng hơn để quản lý ổ khóa, có một thuộc tính Subversion mới `svn:needs-lock`. Khi thuộc tính này được thiết lập (với bất kỳ giá trị nào) trên một tập tin, bất cứ khi nào tập tin được kiểm xuất hoặc cập nhật, các bản sao địa phương được đặt chỉ đọc *trừ khi* bản sao làm việc đó nắm giữ khóa cho tập tin đó. Điều này hoạt động như một lời cảnh báo rằng bạn không nên chỉnh sửa tập tin đó trừ khi bạn lấy khóa trước. Các tập tin này được phiên bản và được đánh dấu chỉ đọc bằng một lớp phủ đặc biệt trong TortoiseSVN để cho biết rằng bạn cần phải có được khóa trước khi biên t.

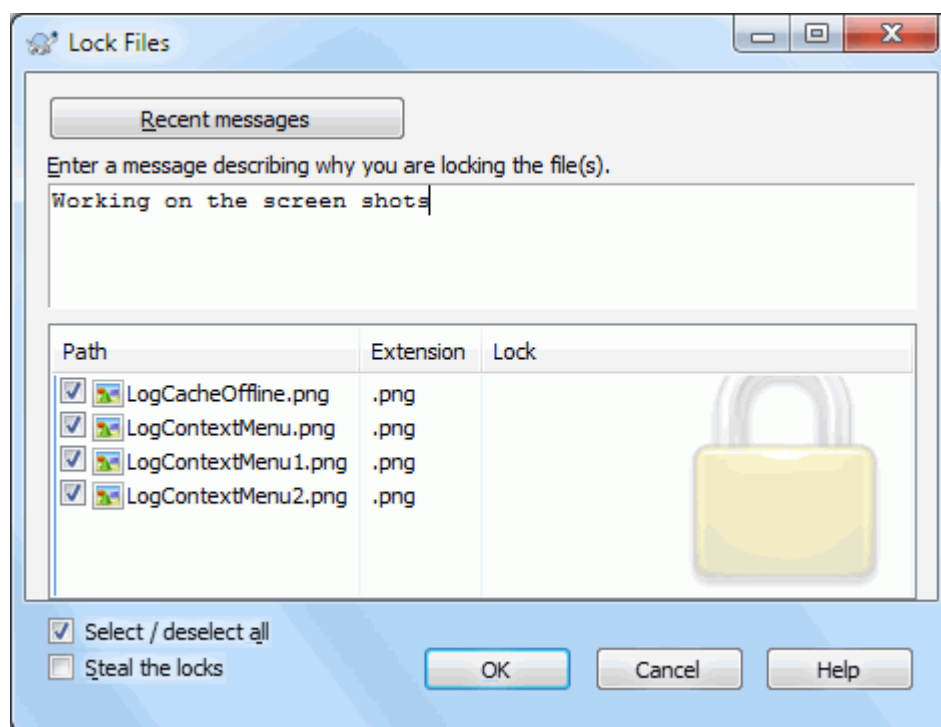
Các khóa được ghi lại bằng vị trí bản sao làm việc cũng như chủ sở hữu. Nếu bạn có nhiều bản sao làm việc (ở nhà, tại nơi làm việc) thì bạn chỉ có thể giữ một khóa ở *một* trong những bản sao làm việc đ.

Nếu một trong những đồng nghiệp của bạn có được một khóa và sau đó đi vào kỳ nghỉ mà không phát hành nó, bạn sẽ làm gì? Subversion cung cấp một phương tiện để các ép buộc ổ khóa. Phát hành một khóa giữ bởi người khác được gọi là *Phá vỡ* khóa, và ép buộc có được một khóa mà người khác đã giữ được gọi là *Trộm cắp* khóa. Đương nhiên đây không phải là điều bạn nên coi nhẹ nếu bạn muốn vẫn là bạn bè với các đồng nghiệp.

Các khóa được ghi lại trong kho, và một mã thông báo khóa được tạo ra trong bản sao của bạn làm việc tại địa phương. Nếu có sự khác biệt, ví dụ nếu một người nào khác đã phá vỡ khóa, mã thông báo khóa địa phương trở thành không hợp lệ. Kho luôn luôn là tham khảo cuối cùng.

## 4.21.2. Lấy một khóa

Chọn tập tin trong bản sao làm việc của bạn mà bạn muốn lấy khóa, sau đó chọn lệnh TortoiseSVN → Nhận Khóa ... .



Hình 4.52. Hộp thoại khóa

Một hộp thoại xuất hiện, cho phép bạn nhập một bình luận, để những người khác có thể thấy lý do tại sao bạn đã khóa tập tin. Bình luận là tùy chọn và chỉ được sử dụng với các kho dựa trên Svnserve. Nếu (và *chỉ* nếu) bạn cần phải ấn cấp khóa từ một người nào khác, chọn hộp kiểm **Ấn cấp khóa**, sau đó nhấn vào **OK**.

Bạn có thể bật thuộc tính dự án `tsvn:logtemplatelock` để cung cấp một mẫu thông điệp cho người dùng để điền vào như thông điệp khóa. Tham khảo [Phần 4.17, “Thiết lập Dự án”](#) để được hướng dẫn làm thế nào để b các thuộc tính.

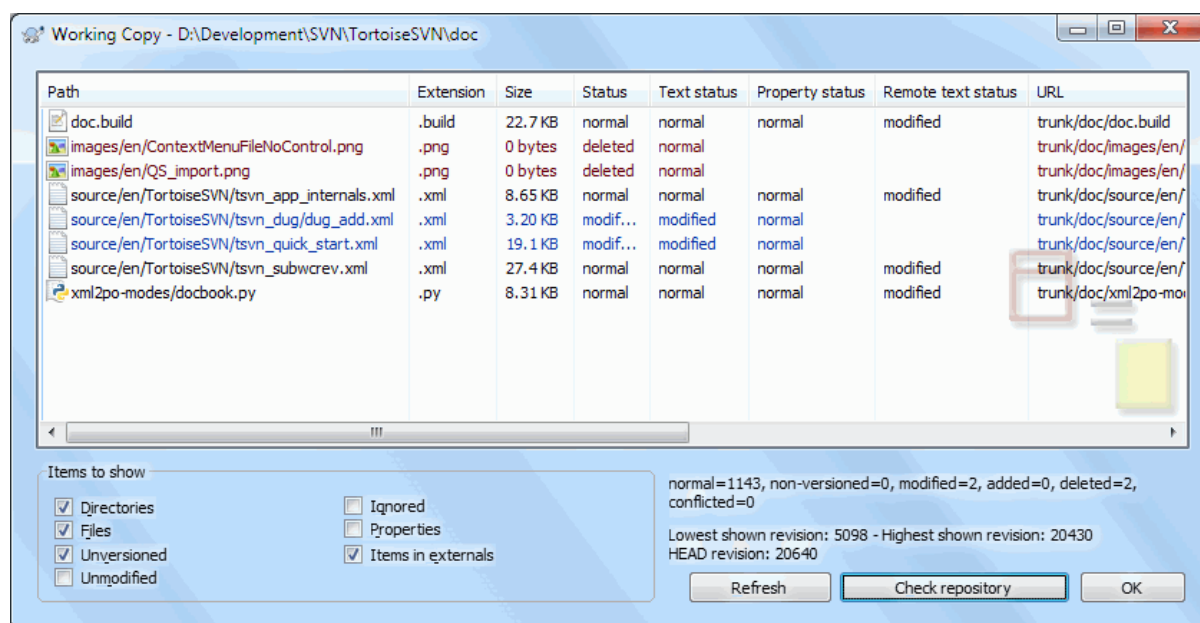
Nếu bạn chọn một thư mục và sau đó sử dụng **TortoiseSVN** → **Nhận Khóa ...** hộp thoại khóa sẽ mở ra với *mỗi* tập tin trong *mỗi* thư mục con được chọn để khóa. Nếu bạn thực sự muốn khóa toàn bộ một hệ thống phân cấp, đó là cách để làm điều đó, nhưng bạn có thể trở nên rất không được ưa thích với các đồng nghiệp nếu bạn khóa họ ra khỏi toàn bộ dự án. Sử dụng cẩn th ...

### 4.21.3. Phát hành một khóa

Để chắc chắn rằng bạn không quên phát hành một khóa bạn không còn cần nữa, tập tin bị khóa được hiển thị trong hộp thoại cam kết và được lựa chọn theo mặc định. Nếu bạn tiếp tục với cam kết, các khóa bạn giữ trên các tập tin được lựa chọn được loại bỏ, ngay cả nếu các tập tin không được sửa đổi. Nếu bạn không muốn phát hành một khóa trên các tập tin nào đó, bạn có thể bỏ chọn chúng (nếu chúng không bị sửa đổi). Nếu bạn muốn giữ khóa trên một tập tin bạn đã sửa đổi, bạn phải kích hoạt hộp kiểm **Giữ khóa** trước khi bạn cam kết các thay đổi của bạn.

Để phát hành một khóa theo cách thủ công, chọn các tập tin trong bản sao làm việc của bạn mà bạn muốn phát hành các khóa, sau đó chọn lệnh **TortoiseSVN** → **Phát hành Khóa** Không có gì hơn nữa để nhập vì thế TortoiseSVN sẽ liên lạc với các kho lưu trữ và phát hành các ổ khóa. Bạn cũng có thể sử dụng lệnh này vào một thư mục để phát hành tất cả các ổ khóa theo cách đệ quy.

### 4.21.4. Kiểm tra trạng thái Khóa



**Hình 4.53. Hộp thoại Kiểm tra Sửa đổi**

Để xem những khóa nào mà bạn và những người khác giữ, bạn có thể sử dụng **TortoiseSVN** → **Kiểm tra sửa đổi ...**. Các mã khóa tại địa phương hiển thị ngay lập tức. Để kiểm tra cho các ổ khóa được giữ bởi những người khác (và để xem nếu có ổ khóa nào của bạn bị phá vỡ hoặc bị đánh cắp), bạn cần phải bấm vào **Kiểm tra Kho**.

Từ trình đơn ngữ cảnh ở đây, bạn cũng có thể nhận được và phát hành ổ khóa, cũng như phá vỡ và ấn cấp các khóa được tổ chức bởi những người khác.



## Tránh Phá vỡ và Trộm cắp Khóa

Nếu bạn phá vỡ hoặc ăn cắp khóa của người khác mà không nói với họ, bạn có khả năng có thể gây ra mất việc làm. Nếu bạn đang làm việc với các loại tập tin không hợp nhất được và bạn ăn cắp khóa của người khác, một khi bạn phát hành các khóa họ được tự do để kiểm nhập những thay đổi của họ và ghi đè lên của bạn. Subversion không mất dữ liệu, nhưng bạn có bị mất bảo vệ làm việc nh mà việc khóa đem lại cho bạn.

### 4.21.5. Làm các Tập tin không bị Khóa Chỉ đọc

Như đã đề cập ở trên, cách hiệu quả nhất để sử dụng khóa là bật thuộc tính `svn:needs-lock` trên các tập tin. Tham khảo [Phần 4.17, “Thiết lập Dự án”](#) để được hướng dẫn làm thế nào để bật các thuộc tính. Tập tin với thuộc tính này được bật sẽ luôn luôn được kiểm tra và cập nhật với cờ chỉ đọc trừ khi bản sao của bạn làm việc nắm giữ một khóa.



Xin nhắc lại, TortoiseSVN sử dụng một lớp phủ đặc biệt để cho biết điều này.

Nếu bạn điều hành một chính sách mà tất cả các tập tin bị khóa thì bạn có thể tìm thấy nó dễ dàng hơn để sử dụng tính năng auto-props của Subversion để thiết lập thuộc tính tự động mỗi khi bạn thêm các tập tin mới. Đọc [Phần 4.17.1.5, “Thiết lập thuộc tính tự động”](#) để biết thêm thông tin.

### 4.21.6. Các kịch bản Hook khóa

Khi bạn tạo một kho lưu trữ mới với Subversion 1.2 hoặc cao hơn, bốn mẫu hook được tạo ra ở thư mục `hooks` trong kho. Chúng được gọi là trước và sau khi nhận được một khóa, và trước và sau khi phát hành một khóa.

Đó là một ý tưởng tốt để cài đặt một kịch bản hook `sau-khóa` và `sau-mở khóa` trên máy chủ mà sẽ gửi email cho thấy các tập tin đã bị khóa. Với một kịch bản tại chỗ, tất cả người dùng của bạn có thể được thông báo nếu ai đó khóa / mở khóa một tập tin. Bạn có thể tìm thấy một kịch bản mẫu ví dụ tại `hooks/post-lock.tmp1` trong thư mục kho lưu trữ của bạn.

Bạn cũng có thể sử dụng móc để không cho phép phá vỡ hoặc ăn cắp ổ khóa, hoặc có thể hạn chế nó đến một quản trị viên được chỉ định. Hoặc có thể bạn muốn gửi email cho chủ sở hữu khi một trong những ổ khóa của họ bị phá vỡ hoặc bị đánh cắp.

Đọc [Phần 3.3, “Kịch bản móc phía máy chủ”](#) để tìm hiểu thêm.

## 4.22. Tạo và áp dụng các bản vá

Đối với các dự án mã nguồn mở như thế này tất cả mọi người đều có truy cập đọc vào kho lưu trữ, và bất cứ ai cũng có thể đóng góp cho dự án. Vì vậy, làm thế nào quản lý những đóng góp đó? Nếu bất kỳ ai cũng có thể cam kết thay đổi, dự án sẽ được vĩnh viễn không ổn định và có lẽ vĩnh viễn bị phá vỡ. Trong trường hợp này thay đổi được quản lý bằng cách nộp một tập tin *bản vá* để nhóm phát triển, những người có quyền truy cập ghi. Họ có thể xem lại các bản vá trước, và sau đó, hoặc gửi vào kho hoặc từ chối nó trở lại cho tác giả.

Các file vá chỉ đơn giản là các tập tin khác biệt-thống nhất hiển thị các khác biệt giữa bản sao làm việc của bạn và bản sửa đổi cơ sở.

### 4.22.1. Tạo một tập tin v

Trước tiên, bạn cần phải thực hiện *và thử nghiệm* các thay đổi của bạn. Sau đó thay vì sử dụng TortoiseSVN → Cam kết ... trên thư mục mẹ, bạn chọn TortoiseSVN → Tạo Bản Vá ...



**Hình 4.54. Hộp thoại Tạo Bản Vá**

bây giờ bạn có thể chọn các tập tin bạn muốn đưa vào bản vá, cũng giống như bạn làm với một cam kết đầy đủ. Việc này sẽ tạo ra một tập tin duy nhất có chứa một bản tóm tắt của tất cả các thay đổi bạn đã thực hiện trong các tập tin được chọn kể từ khi lần cập nhật cuối từ kho.

Các cột trong hộp thoại này có thể được tùy chỉnh trong cùng một cách như các cột trong hộp thoại **Kiểm tra để sửa đổi**. Đọc [Phần 4.7.4, “Trạng thái địa phương và từ xa”](#) để biết thêm chi tiết.

Bằng cách nhấp vào nút **Tùy chọn**, bạn có thể xác định bản vá lỗi được tạo ra như thế nào. Ví dụ, bạn có thể xác định rằng những thay đổi trong các kết thúc dòng hay khoảng trắng không được bao gồm trong các tập tin vá lỗi cuối cùng.

Bạn có thể tạo ra các bản vá lỗi riêng biệt có chứa các thay đổi đối với các tập hợp tập tin khác nhau. Tất nhiên, nếu bạn tạo ra một tập tin bản vá, thực hiện thêm vài thay đổi đến *cùng* các tập tin và sau đó tạo ra một bản vá khác, tập tin vá lỗi thứ hai sẽ bao gồm *cả hai* tập h thay đổi.

Chỉ cần lưu tập tin bằng cách sử dụng một tên tập tin do bạn lựa chọn. Các tập tin vá lỗi có thể có bất kỳ phần mở rộng mà bạn thích, nhưng theo quy ước, họ nên sử dụng phần mở rộng `.patch` hoặc `.diff`. Bây giờ bạn đã sẵn sàng để n tập tin bản vá của bạn.

Bạn cũng có thể lưu các bản vá vào clipboard thay vì vào một tập tin. Bạn có thể muốn làm điều này để bạn có thể dán nó vào một thư điện tử để xem xét bởi những người khác. Hoặc nếu bạn có hai bản sao làm việc trên một máy và bạn muốn chuyển giao thay đổi từ một bản đến bản khác, một bản vá lỗi trong clipboard là một cách thuận tiện để làm điều này.

Nếu bạn thích, bạn có thể tạo một tập tin bản vá từ bên trong hộp thoại **Cam kết** hoặc **Kiểm tra cho sửa đổi**. Chỉ cần chọn các tập tin và sử dụng các mục trình đơn ngữ cảnh để tạo ra một bản vá từ những tập tin. Nếu bạn muốn xem hộp thoại **Tùy chọn**, bạn phải giữ **shift** khi bạn nhấp chuột phải.

#### 4.22.2. Áp dụng một tập tin v

Các tập tin vá lỗi được áp dụng vào bản sao làm việc của bạn. Điều này nên được thực hiện từ cấp độ thư mục tương tự như đã được sử dụng để tạo ra các bản vá. Nếu bạn không chắc chắn việc này là gì, chỉ cần nhìn vào

dòng đầu tiên của các tập tin và lỗi. Ví dụ, nếu tập tin đầu tiên đang được làm việc là `doc/source/english/chapter1.xml` và dòng đầu tiên trong các tập tin và lỗi là `Index: english/chapter1.xml` thì bạn cần phải áp dụng các bản vá vào thư mục `doc/source/`. Tuy nhiên, giả sử là bạn đang ở trong bản sao làm việc đúng, nếu bạn chọn cấp độ thư mục sai, TortoiseSVN sẽ thông báo và đề nghị mức độ chính xác.

Để áp dụng một tập tin bản vá vào bản sao làm việc của bạn, bạn cần phải có ít nhất là truy cập đọc vào kho lưu trữ. Lý do cho điều này là chương trình hợp nhất phải tham khảo các thay đổi trở lại bản sửa đổi mà đã được thực hiện bởi nhà phát triển từ xa.

Từ trình đơn ngữ cảnh cho thư mục đó, nhấp chuột vào TortoiseSVN → Áp dụng bản vá... Việc này sẽ mang đến một hộp thoại mở tập tin cho phép bạn chọn tập tin và lỗi để áp dụng. Mặc định, chỉ có tập tin `.patch` hoặc `.diff` được hiển thị, nhưng bạn có thể lựa chọn “Tất cả các tập tin”. Nếu bạn trước đó đã lưu một bản vá vào clipboard, bạn có thể sử dụng Mở từ clipboard ... trong hộp thoại mở tập tin. Lưu ý rằng tùy chọn này chỉ xuất hiện nếu bạn đã lưu các bản vá vào clipboard bằng cách sử dụng TortoiseSVN → Tạo bản vá... . Sao chép một bản vá vào clipboard từ ứng dụng khác sẽ không làm cho nút bấm này xuất hiện.

Cách khác, nếu các tập tin và lỗi có phần mở rộng `.patch` hoặc `.diff`, bạn có thể nhấp chuột phải vào nó trực tiếp và chọn TortoiseSVN → Áp dụng bản vá ... . Trong trường hợp này, bạn sẽ được nhắc nhở để nhập vào vị trí một bản sao làm việc.

Hai phương pháp này chỉ cung cấp các cách khác nhau để làm cùng một chuyện. Với phương pháp đầu tiên, bạn chọn WC và duyệt đến tập tin và lỗi. Với cách thứ hai bạn chọn tập tin và lỗi và duyệt đến WC.

Một khi bạn đã lựa chọn các tập tin và lỗi và vị trí bản sao làm việc, TortoiseMerge sẽ chạy để hợp nhất các thay đổi từ tập tin và lỗi với bản sao làm việc của bạn. Một cửa sổ nhỏ liệt kê các tập tin đã được thay đổi. Lần lượt kích đúp vào mỗi tập tin, xem xét những thay đổi và lưu các tập tin đã được hợp nhất.

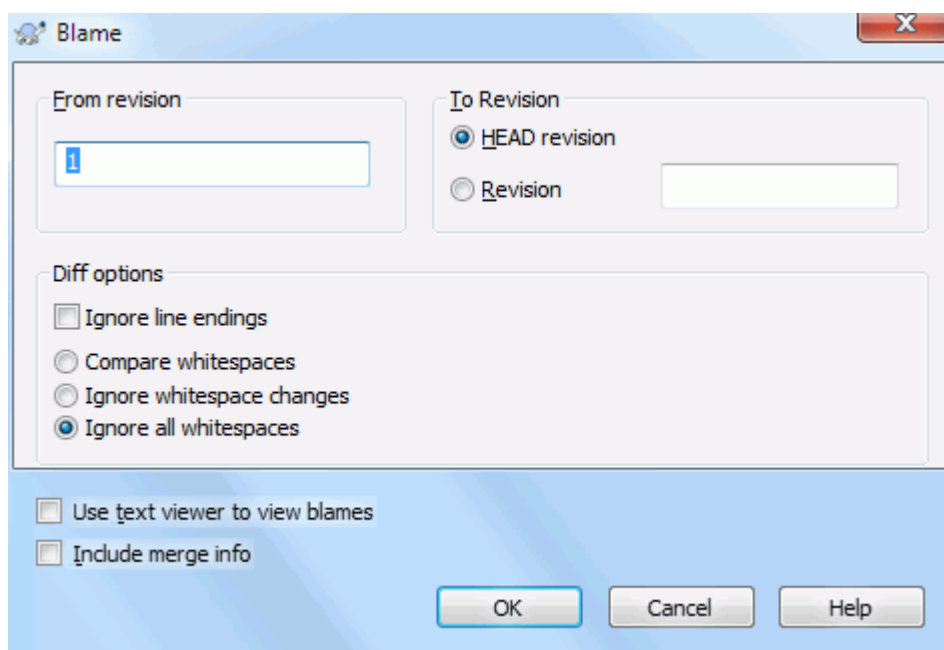
Bản vá của các nhà phát triển từ xa đã được áp dụng vào bản sao làm việc của bạn, vì vậy bạn cần phải cam kết để cho phép tất cả mọi người khác được truy cập vào thay đổi từ kho lưu trữ.

## 4.23. Ai Đã Thay Đổi Dòng Nào?

Đôi khi bạn cần phải biết không chỉ có những dòng đã thay đổi, mà còn là chính xác những người nào đã thay đổi những dòng cụ thể trong một tập tin. Đó là khi lệnh TortoiseSVN → Đổ lỗi cho ... , đôi khi cũng được gọi là lệnh *chú thích* trở nên tiện dụng.

Các lệnh này liệt kê, đối với mỗi dòng trong một tập tin, tác giả và sửa đổi mà dòng đã được thay đổi.

### 4.23.1. Đồ Lỗi vì Tập Tin



**Hình 4.55. Chú Thích / Hộp Thoại Đồ Lỗi**

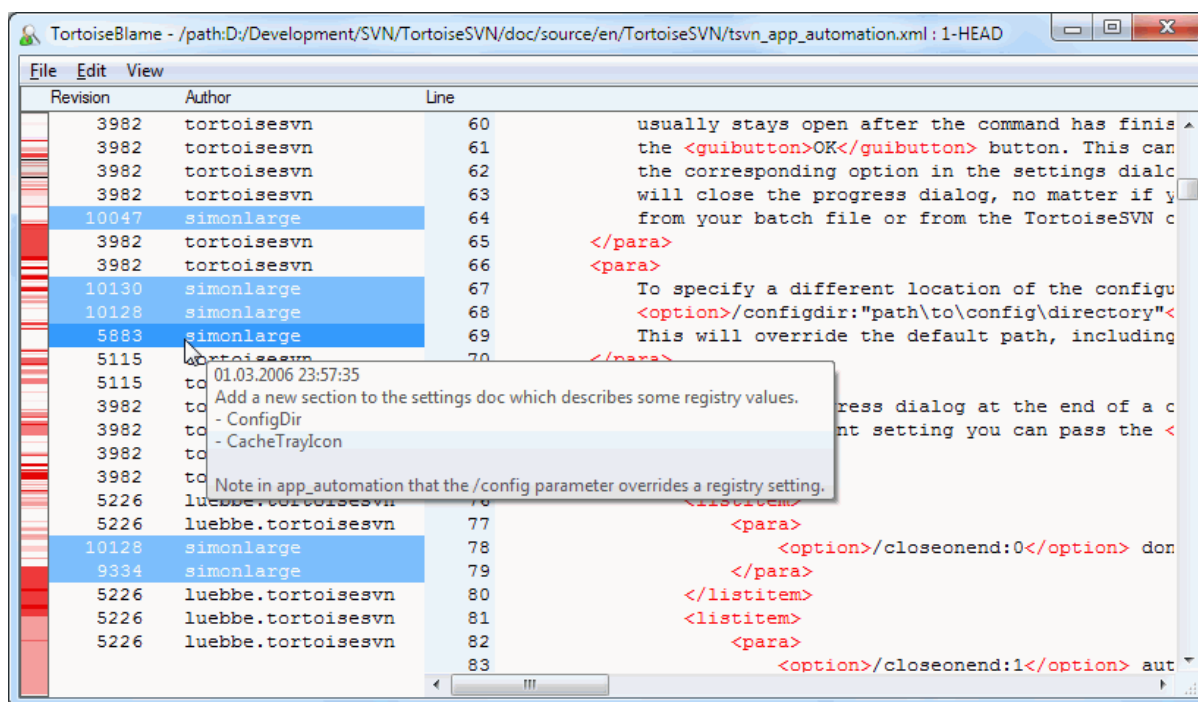
Nếu bạn không quan tâm đến thay đổi từ các phiên bản trước đó, bạn có thể thiết lập bản sửa đổi để từ đó bắt đầu đồ lỗi. Thiết lập nó thành 1, nếu bạn muốn đồ lỗi cho m sửa đổi.

Theo mặc định tập tin lỗi được xem bằng cách sử dụng *TortoiseBlame*, trong đó làm nổi rõ các phiên bản khác nhau để làm cho nó dễ đọc hơn. Nếu bạn muốn in hay chỉnh sửa các tập tin lỗi, chọn *Sử dụng trình xem Văn Bản để xem đồ lỗi*.

Bạn có thể chỉ định cách mà các thay đổi kết thúc dòng và khoảng trắng được xử lý. Các tùy chọn này được mô tả trong [Phần 4.10.2, “Tùy chọn cho Hết-Dòng và Khoảng Trắng”](#). Hành vi mặc định là để xử lý tất cả các sự khác biệt khoảng trắng và cuối dòng như là thay đổi thực sự, nhưng nếu bạn muốn bỏ qua một sự thay đổi thật đầu dòng và tìm ra tác giả ban đầu, bạn có thể chọn một tùy chọn thích hợp ở đây.

Bạn có thể bao gồm thông tin hợp nhất nếu bạn muốn, mặc dù tùy chọn này có thể mất đáng kể nhiều thời gian hơn để lấy từ máy chủ. Khi các dòng được sáp nhập từ một nguồn khác, thông tin đồ lỗi cho thấy bản sửa đổi mà thay đổi đã được thực hiện trong nguồn nguyên thủy cũng như bản sửa đổi khi nó được sáp nhập vào tập tin này.

Một khi bạn nhấn *OK* TortoiseSVN bắt đầu lấy các dữ liệu để tạo ra các tập tin đồ lỗi. Xin lưu ý: Điều này có thể mất vài phút để kết thúc, tùy thuộc vào bao nhiêu tập tin đã thay đổi và tất nhiên vào kết nối mạng của bạn đến kho lưu trữ. Một khi quá trình đồ lỗi đã hoàn tất thì kết quả sẽ được viết vào một tập tin tạm thời và bạn có thể xem các kết quả.



**Hình 4.56. TortoiseBlame**

TortoiseBlame, được bao gồm với TortoiseSVN, làm cho việc đọc tập tin dễ đọc hơn. Khi bạn di chuyển chuột trên một dòng trong cột thông tin đồ lỗi, tất cả các dòng với cùng bản sửa đổi được hiển thị với một nền tối hơn. Các dòng từ các phiên bản khác đã được thay đổi bởi cùng một tác giả được thể hiện với một nền sáng. Màu sắc có thể làm việc không rõ ràng nếu bạn đặt màn hình của bạn ở chế độ 256 màu.

Nếu bạn nhấp chuột trái trên một dòng, tất cả các dòng cùng bản sửa đổi được làm nổi, và các dòng từ các phiên bản khác của cùng tác giả được làm nổi trong một màu sáng hơn. Việc làm nổi bật này là dính, cho phép bạn di chuyển chuột mà không làm mất những nổi bật. Nhấp vào vào sửa đổi đó một lần nữa để tắt nổi bật.

Các ý kiến sửa đổi (thông điệp tường trình) được thể hiện trong một hộp gợi ý bất cứ khi nào chuột di chuyển lên cột thông tin đồ lỗi. Nếu bạn muốn sao chép các thông điệp tường trình cho bản sửa đổi đó, sử dụng trình đơn ngữ cảnh xuất hiện khi bạn kích chuột phải vào cột thông tin đồ lỗi.

Bạn có thể tìm kiếm trong các báo cáo đồ lỗi sử dụng **Biên tập** → **Tìm kiếm...**. Điều này cho phép bạn tìm kiếm các số sửa đổi, tác giả và nội dung của tập tin. Thông điệp tường trình không được bao gồm trong tìm kiếm - bạn nên sử dụng **Hộp thoại Tường trình** để tìm kiếm những cái đó.

Bạn cũng có thể nhảy đến một dòng cụ thể bằng cách sử dụng **Biên tập** → **Đi đến dòng ...**.

Khi chuột nằm trên các cột thông tin đồ lỗi, một trình đơn ngữ cảnh có sẵn giúp so sánh các phiên bản và kiểm tra lịch sử, bằng cách sử dụng số phiên bản của dòng dưới con chuột như là một tài liệu tham khảo. **Trình đơn ngữ cảnh** → **Đồ lỗi cho sửa đổi trước** tạo ra một báo cáo đồ lỗi cho cùng một tập tin, nhưng sử dụng bản sửa đổi trước đó làm giới hạn trên. Điều này cung cấp cho bạn báo cáo đồ lỗi cho tình trạng của tập tin ngay trước dòng bạn đang nhìn vào được thay đổi lần cuối. **Trình đơn ngữ cảnh** → **Hiện thị những thay đổi** chạy trình xem khác biệt của bạn, hiển thị cho bạn những gì thay đổi trong phiên bản tham chiếu. **Trình đơn ngữ cảnh** → **Hiện tường trình** hiển thị hộp thoại đăng nhập sửa đổi bắt đầu với bản sửa đổi được tham chiếu.

Nếu bạn cần một chỉ dấu hình ảnh tốt hơn của những thay đổi lâu đời nhất và mới nhất, chọn **Xem** → **Màu tuổi của dòng**. Điều này sẽ sử dụng một màu sắc với độ chuyển đậm nhạt để hiển thị dòng mới trong các đường màu đỏ và dòng cũ trong màu xanh. Các màu mặc định là khá nhẹ, nhưng bạn có thể thay đổi nó bằng cách sử dụng các thiết lập của TortoiseBlame.



Nếu bạn đang sử dụng Theo Dõi Hợp Nhất và yêu cầu thông tin hợp nhất khi bắt đầu đồ lỗi, các dòng bị sáp nhập được thể hiện hơi khác nhau. Khi một dòng đã thay đổi do kết quả hợp nhất từ đường dẫn khác, TortoiseBlame sẽ hiển thị bản sửa đổi và tác giả của thay đổi mới nhất trong các tập tin ban đầu chứ không phải là bản sửa đổi nơi việc hợp nhất đã diễn ra. Những dòng này được thể hiện bằng cách hiển thị các sửa đổi và tác giả trong chữ in nghiêng. Bản sửa đổi nơi hợp nhất đã diễn ra được hiển thị một cách riêng biệt trong tooltip khi bạn di chuyển chuột lên các cột thông tin đồ lỗi. Nếu bạn không muốn các dòng sáp nhập được thể hiện trong cách này, bỏ chọn hộp kiểm **Bao gồm thông tin hợp nhất** khi bắt đầu đồ lỗi.

Nếu bạn muốn nhìn thấy các đường dẫn liên quan đến việc hợp nhất, chọn **Xem** → **Các đường dẫn hợp nhất**. Điều này cho thấy đường dẫn nơi dòng đã được thay đổi lần cuối, không bao gồm các thay đổi từ một hợp nhất.

Bản sửa đổi được hiển thị trong thông tin đồ lỗi đại diện cho bản sửa đổi cuối cùng nơi mà nội dung của dòng đó thay đổi. Nếu tập tin được tạo ra bằng cách sao chép một tập tin khác, thì cho đến khi bạn thay đổi một dòng, đồ lỗi cho sửa đổi của nó sẽ hiển thị thay đổi mới nhất trong tập tin nguồn gốc, chứ không phải bản sửa đổi nơi bản sao đã được thực hiện. Điều này cũng áp dụng cho các đường dẫn được hiển thị với thông tin kết hợp. Đường dẫn hiển thị vị trí kho lưu trữ nơi sự thay đổi cuối cùng đã được thực hiện tới dòng đó.

Các thiết lập cho TortoiseBlame có thể được truy cập bằng cách sử dụng **TortoiseSVN** → **Thiết lập ...** trên tab TortoiseBlame. Tham khảo **Phần 4.30.9, “Thiết Lập TortoiseBlame”**.

### 4.23.2. Đồ lỗi cho khác biệt

Một trong những hạn chế của báo cáo đồ lỗi là nó chỉ cho thấy các tập tin như lúc nó đã ở trong một phiên bản nhất định, và người cuối cùng thay đổi mỗi dòng. Đôi khi bạn muốn biết những thay đổi gì đã được thực hiện, cũng như những người đã làm ra nó. Nếu bạn nhấp chuột phải trên một dòng trong TortoiseBlame bạn có một menu trình đơn ngữ cảnh để hiển thị các thay đổi được thực hiện trong bản chỉnh sửa đó. Nhưng nếu bạn muốn xem các thay đổi và các thông tin đồ lỗi một cách đồng thời thì bạn cần một sự kết hợp của các báo cáo tìm khác biệt và đồ lỗi.

Hộp thoại tương trình sửa đổi bao gồm một số tùy chọn cho phép bạn làm điều này.

**Đồ lỗi cho sửa đổi**

Trong cửa sổ trên cùng, chọn 2 bản sửa đổi, sau đó chọn **Trình đơn ngữ cảnh** → **Đồ lỗi cho các phiên bản**. Điều này sẽ lấy dữ liệu đồ lỗi cho 2 phiên bản, sau đó sử dụng trình xem khác biệt để so sánh hai tập tin đồ lỗi.

**Đồ lỗi cho thay đổi**

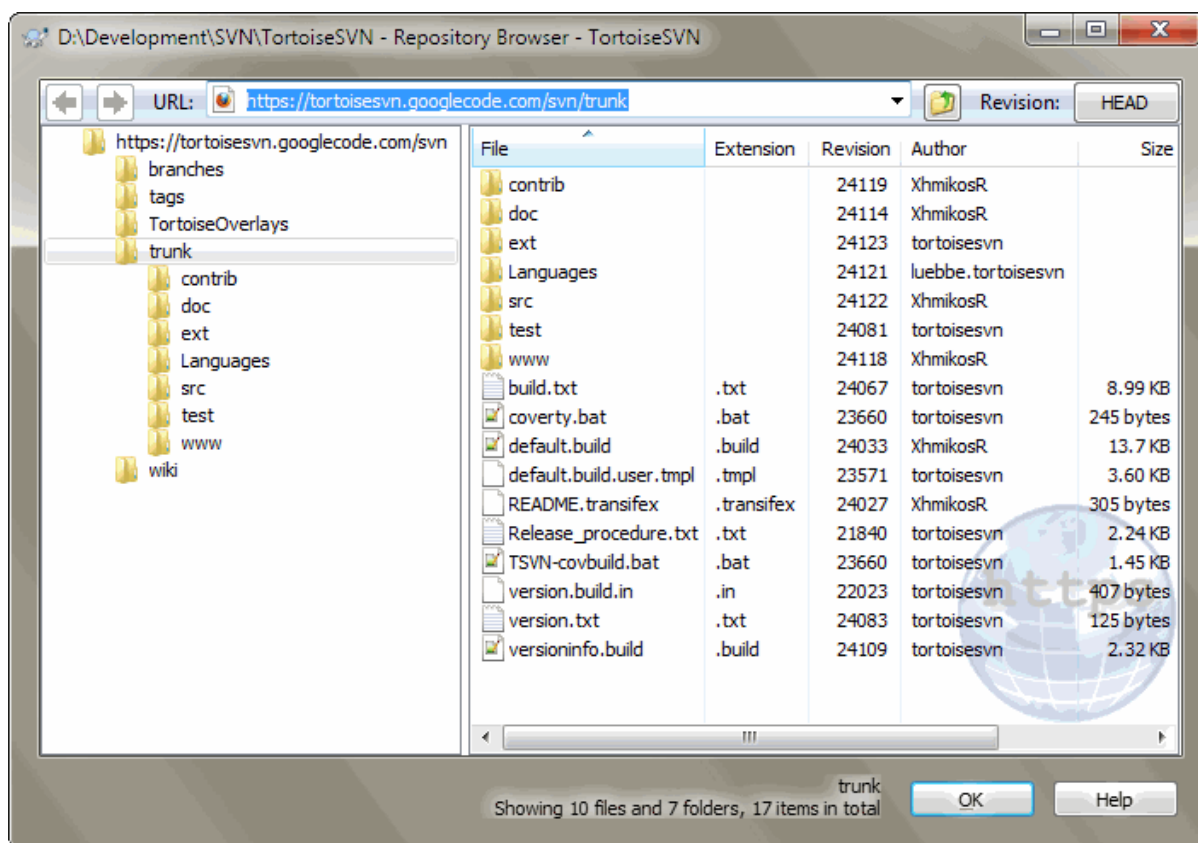
Chọn một sửa đổi trong cửa sổ trên cùng, sau đó chọn một tập tin trong khung bên dưới và chọn **Trình đơn ngữ cảnh** → **Đồ lỗi cho các thay đổi**. Điều này sẽ lấy dữ liệu đồ lỗi cho việc sửa đổi được lựa chọn và sửa đổi trước đó, sau đó sử dụng trình xem khác biệt để so sánh hai đồ lỗi cho các tập tin.

**So sánh và Đồ lỗi cho bản làm việc CƠ SỞ**

Hiển thị tương trình cho một tập tin duy nhất, và trong cửa sổ trên cùng, chọn một phiên bản duy nhất, sau đó chọn **Trình đơn ngữ cảnh** → **So sánh và Đồ lỗi cho bản làm việc CƠ SỞ**. Điều này sẽ lấy dữ liệu đồ lỗi cho bản sửa đổi được lựa chọn, và cho tập tin trong bản làm việc CƠ SỞ, sau đó sử dụng trình xem khác biệt để so sánh hai đồ lỗi cho các tập tin.

## 4.24. Trình duyệt Kho

Đôi khi bạn cần làm việc trực tiếp trên kho lưu trữ, mà không có một bản sao làm việc. Đó là những gì *Trình duyệt Kho lưu trữ* được dùng cho. Cũng giống như explorer và các lớp phủ biểu tượng cho phép bạn xem bản sao làm việc của bạn, do đó, trình duyệt Kho lưu trữ cho phép bạn xem cấu trúc và tình trạng của kho.



**Hình 4.57. Trình duyệt Kho**

Với Trình duyệt Kho lưu trữ, bạn có thể thực hiện các lệnh như sao chép, di chuyển, đổi tên, ... trực tiếp trên kho lưu trữ.

Các trình duyệt kho trông rất giống với Windows explorer, ngoại trừ rằng nó hiển thị các nội dung của kho tại một phiên bản nhất định hơn là các tập tin trên máy tính của bạn. Trong khung bên trái, bạn có thể nhìn thấy một cây thư mục, và trong khung bên phải là nội dung của thư mục được lựa chọn. Ở phía trên của Cửa sổ Trình duyệt Kho lưu trữ, bạn có thể nhập địa chỉ URL của kho và bản sửa đổi mà bạn muốn để duyệt.

Các thư mục bao gồm thuộc tính `svn:externals` cũng được hiển thị trong trình duyệt kho. Những thư mục được hiển thị với một mũi tên nhỏ cho biết rằng chúng không phải là một phần của cấu trúc kho lưu trữ, chỉ là các liên kết.

Cũng giống như Windows Explorer, bạn có thể bấm vào tiêu đề cột trong khung bên phải nếu bạn muốn thiết lập thứ tự sắp xếp. Và cũng như trong explorer có các trình đơn ngữ cảnh có sẵn trong cả hai khung.

Trình đơn ngữ cảnh cho một tập tin cho phép bạn:

- Mở tập tin được lựa chọn, hoặc là với trình xem mặc định cho loại tập tin, hoặc với một chương trình mà bạn chọn.
- Biên tập các tập tin được chọn. Điều này sẽ kiểm xuất một bản sao làm việc tạm thời và bắt đầu trình soạn thảo mặc định cho loại tập tin đó. Khi bạn đóng chương trình soạn thảo, nếu thay đổi đã được lưu thì một hộp thoại cam kết xuất hiện, cho phép bạn nhập vào một bình luận và cam kết thay đổi.
- Hiện trường trình sửa đổi cho tập tin đó, hoặc hiển thị một biểu đồ của tất cả các bản sửa đổi để bạn có thể nhìn thấy từ nơi mà các tập tin đến.
- Đồ lỗi cho các tập tin, để xem ai đã thay đổi dòng nào và khi nào.

- Kiểm xuất một tập tin duy nhất. Điều này tạo ra một bản sao làm việc “thưa thớt” mà trong đó chỉ chứa tập tin này.
- Xóa hoặc đổi tên tập tin.
- Lưu một bản sao chưa phiên bản của tập tin vào ổ cứng của bạn.
- Sao chép địa chỉ URL được hiển thị trong thanh địa chỉ vào clipboard.
- Tạo một bản sao của tập tin, hoặc một phần khác của kho, hoặc một bản sao làm việc bắt nguồn từ cùng một kho lưu trữ.
- Xem / Biên t các thuộc tính của tập tin.
- Tạo một phím tắt để bạn có thể nhanh chóng bắt đầu trình duyệt kho một lần nữa, mở trực tiếp tại địa điểm này.

Trình đơn ngữ cảnh cho một thư mục cho phép bạn:

- Hiện tường trình sửa đổi cho thư mục đó, hoặc hiển thị một biểu đồ của tất cả các bản sửa đổi để bạn có thể nhìn thấy từ nơi mà các thư mục đến.
- Xuất khẩu thư mục vào một bản sao không phiên bản địa phương trên ổ cứng của bạn.
- Kiểm xuất thư mục để sản xuất một bản sao làm việc địa phương trên ổ cứng của bạn.
- Tạo một thư mục mới trong kho.
- Thêm tập tin hoặc thư mục chưa phiên bản trực tiếp vào kho lưu trữ. Đây về cơ bản là tác vụ Nhập Khẩu của Subversion.
- Xóa hoặc đổi tên thư mục.
- Tạo một bản sao của thư mục, hoặc một phần khác của kho, hoặc một bản sao làm việc bắt nguồn từ cùng một kho lưu trữ. Điều này cũng có thể được sử dụng để tạo ra một chi nhánh / thẻ mà không cần phải có một bản sao làm việc được kiểm xuất.
- Xem / Biên t các thuộc tính của thư mục.
- Đánh dấu thư mục để so sánh. Một thư mục được đánh dấu sẽ được hiển thị in đậm.
- So sánh thư mục với một thư mục được đánh dấu trước đó, hoặc như là một khác biệt thống nhất, hoặc là một danh sách các tập tin thay đổi mà sau đó có thể được tìm khác biệt trực quan bằng cách sử dụng công cụ khác biệt mặc định. Điều này có thể đặc biệt hữu ích để so sánh hai thẻ, hoặc thân và chi nhánh để xem những gì đã thay đổi.

Nếu bạn chọn hai thư mục trong khung bên phải, bạn có thể xem sự khác biệt hoặc như là một khác biệt-thống nhất, hoặc như một danh sách các tập tin có thể được tìm khác biệt trực quan bằng cách sử dụng công cụ khác biệt mặc định.

Nếu bạn chọn nhiều thư mục ở khung bên phải, bạn có thể kiểm xu tất cả chúng cùng một lúc vào một thư mục mẹ chung.

Nếu bạn chọn 2 thẻ được sao chép từ cùng một gốc (thường `/trunk/`), Bạn có thể sử dụng Trình đơn Ngữ cảnh → Hiện Tường Trình... để xem danh sách các bản sửa đổi giữa hai điểm thẻ.

Các mục bên ngoài (được tham chiếu sử dụng `svn:externals` cũng được hiển thị trong trình duyệt kho lưu trữ, và bạn thậm chí có thể đi sâu vào nội dung thư mục. Các mục bên ngoài được đánh dấu bằng một mũi tên màu đỏ trên mục.

Bạn có thể sử dụng **F5** để làm mới màn hình xem như bình thường. Điều này sẽ làm mới tất cả mọi thứ đang hiển thị. Nếu bạn muốn nạp trước hoặc làm mới thông tin cho các nút chưa được mở, sử dụng **Ctrl-F5**. Sau đó, mở rộng bất kỳ nút sẽ xảy ra ngay lập tức mà không có một sự chậm trễ mạng trong khi thông tin được n.

Bạn cũng có thể sử dụng trình duyệt kho lưu trữ cho các hoạt động kéo-và-thả. Nếu bạn kéo một thư mục từ explorer vào trong trình duyệt-kho, nó sẽ được nhập khẩu vào kho lưu trữ. Lưu ý rằng nếu bạn kéo nhiều mục, chúng sẽ được nhập khẩu trong các cam kết riêng biệt.

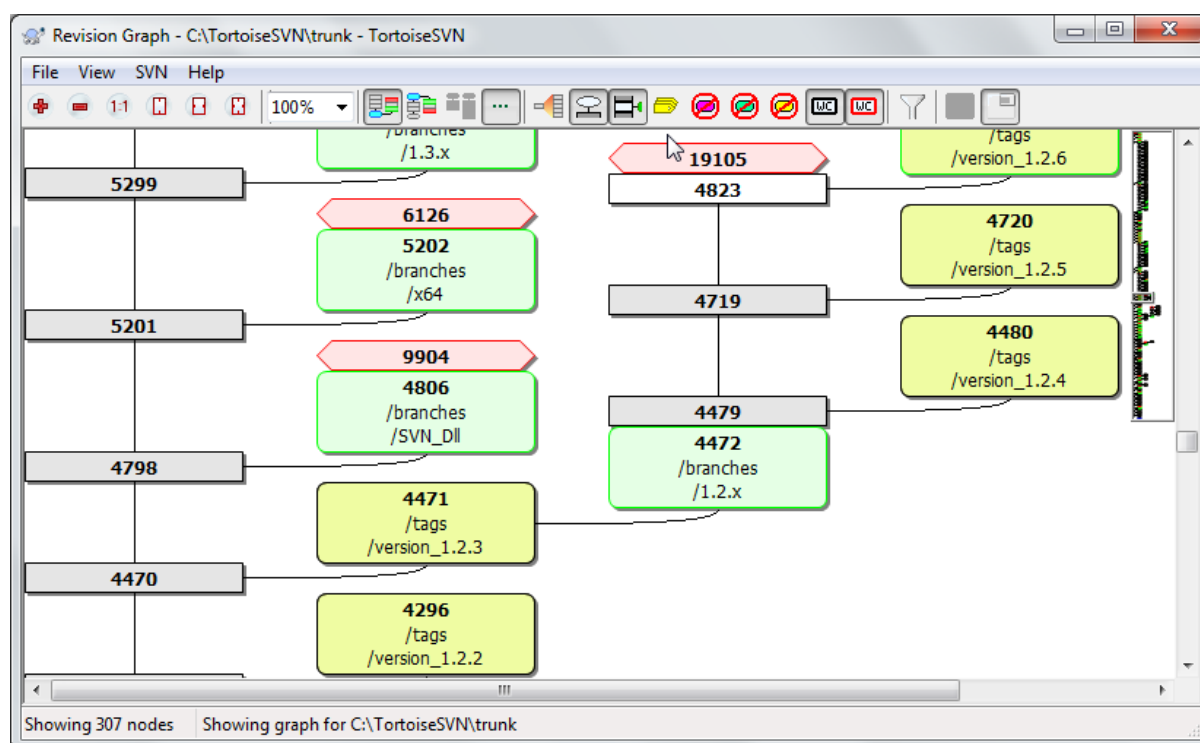
Nếu bạn muốn di chuyển một mục trong kho, chỉ việc kéo trái nó đến vị trí mới. Nếu bạn muốn tạo một bản sao thay vì di chuyển các mục, thay vào đó dùng **Ctrl-trái** kéo. Khi sao chép, con trỏ có một dấu hiệu “thêm” trên nó, giống như trong Explorer.

Nếu bạn muốn sao chép / di chuyển một tập tin hoặc thư mục vào một vị trí khác và cũng cung cấp cho nó một cái tên mới cùng một lúc, bạn có thể kéo phải hoặc **Ctrl-kéo phải** hạng mục thay vì sử dụng kéo trái. Trong trường hợp đó, một hộp thoại đổi tên được hiển thị, nơi bạn có thể nhập một tên mới cho tập tin hoặc thư mục.

Bất cứ khi nào bạn thực hiện thay đổi trong kho lưu trữ bằng cách sử dụng một trong những phương pháp này, bạn sẽ được trình bày với một mục tin thông điệp tường trình. Nếu bạn kéo một cái gì đó do nhầm lẫn, đây cũng là cơ hội để bạn hủy bỏ hành động.

Đôi khi, khi bạn cố gắng để mở một đường dẫn, bạn sẽ nhận được một thông báo lỗi trong vị trí của các mục chi tiết. Điều này có thể xảy ra nếu bạn chỉ định một URL không hợp lệ, hoặc nếu bạn không có quyền truy cập, hoặc nếu có một số vấn đề máy chủ khác. Nếu bạn cần sao chép thông báo này để bao gồm nó trong một email, chỉ cần nhấp chuột phải vào nó và sử dụng Trình Đơn Ngữ Cảnh → Sao chép thông báo lỗi vào clipboard, hoặc đơn giản là sử dụng **Ctrl+C**.

## 4.25. Các đồ thị sửa đổi



**Hình 4.58. Một Đồ thị Sửa đổi**

Đôi khi bạn cần phải biết nơi đặt các chi nhánh và các thẻ được lấy từ thân cây, và cách lý tưởng để xem loại thông tin này là như một biểu đồ hoặc cấu trúc cây. Đó là khi bạn cần sử dụng TortoiseSVN → Đồ thị Sửa đổi ...

Lệnh này phân tích lịch sử bản sửa đổi và cố gắng để tạo ra một cây hiển thị các điểm mà tại đó các bản sao đã được thực hiện, và l các chi nhánh / thẻ đã bị xóa.



## Quan trọng

Để tạo ra đồ thị, TortoiseSVN phải lấy tất cả các thông điệp tường trình từ thư mục kho gốc. Không cần phải nói việc này có thể mất vài phút ngay cả với một kho lưu trữ với một vài ngàn sửa đổi, tùy thuộc vào tốc độ máy chủ, mạng băng thông, vv. Nếu bạn thử việc này với một cái gì đó giống như dự án *Apache* mà hiện nay đã có hơn 500.000 sửa đổi, bạn có thể chờ đợi một thời gian.

Các tin tốt là rằng nếu bạn đang sử dụng bộ nhớ đệm cho tường trình, bạn chỉ phải chịu sự chậm trễ này một lần. Sau đó, dữ liệu tường trình được tổ chức tại địa phương. Bộ nhớ đệm cho tường trình được kích hoạt trong thiết lập của TortoiseSVN.

### 4.25.1. Các Nút Đồ thị Sửa đổi

Mỗi nút đồ thị sửa đổi đại diện cho một sửa đổi trong kho, nơi một cái gì đó thay đổi trong cây mà bạn đang nhìn vào. Các loại khác nhau của nút có thể được phân biệt bởi hình dạng và màu sắc. Các hình dạng là cố định, nhưng màu sắc có thể được thiết lập bằng cách sử dụng TortoiseSVN → Thiết l

Các mục được thêm vào hoặc sao chép

Các mục đã được thêm vào, hoặc được tạo ra bằng cách sao chép một tập tin / thư mục được hiển thị bằng một hình chữ nhật tròn. Màu mặc định là màu xanh lá cây. Các thê và thân cây đang được coi là một trường hợp đặc biệt và sử dụng một màu khác nhau, tùy thuộc vào TortoiseSVN → Thiết Lập .

Các mục đã xóa

Các mục đã xóa, ví dụ như một chi nhánh không còn cần thiết, được thể hiện bằng cách sử dụng một hình bát giác (hình chữ nhật với các góc bị cắt). Màu mặc định là màu đỏ.

Các mục được ổi tên

Các mục ổi tên cũng được hiển thị bằng cách sử dụng một hình bát giác, nhưng màu sắc mặc định là màu xanh dương.

Bản sửa đổi đầu chi nhánh

Đồ thị thường bị giới hạn hiển thị các điểm chi nhánh, nhưng nó thường hữu ích để có thể thấy việc sửa đổi CHÍNH tương ứng cho từng nhánh. Nếu bạn chọn Sửa đổi CHÍNH , Mỗi nút sửa đổi CHÍNH sẽ được hiển thị như là một hình elip. Lưu ý rằng CHÍNH ở đây đề cập đến bản sửa đổi cuối cùng cam kết trên đường dẫn đó, không sửa đổi CHÍNH của kho.

Bản sao sửa đổi làm việc

Nếu bạn gọi đồ thị sửa đổi từ một bản sao làm việc, bạn có thể chọn để hiển thị bản sửa đổi CHÍNH trên đồ thị bằng cách sử dụng Hiển thị sửa đổi WC , mà sẽ đánh dấu nút CƠ SỞ với một phác thảo được in đậm.

Bản sao làm việc được sửa đổi

Nếu bạn gọi đồ thị sửa đổi từ một bản sao làm việc, bạn có thể chọn để hiển thị thêm một nút đại diện cho bản sao làm việc được chỉnh sửa của bạn bằng cách sử dụng Hiển thị sửa đổi WC . Đây là một nút hình elip với một phác thảo đậm màu đỏ theo mặc định.

Mục bình thường

Tất cả các mục khác được hiển thị bằng cách sử dụng một hình chữ nhật đơn giản.

Lưu ý rằng theo mặc định, đồ thị chỉ hiển thị các điểm của các mục đã được thêm vào, sao chép hoặc bị xóa. Hiển thị tất cả các sửa đổi của dự án sẽ tạo ra một đồ thị rất lớn cho các trường hợp không tầm thường. Nếu bạn thực sự muốn xem *tất cả* các bản sửa đổi mà thay đổi được thực hiện, có một tùy chọn để làm điều này trong trình đơn Xem và trên thanh công cụ.

Giao diện mặc định (xếp nhóm) đặt các nút theo cách mà vị trí thẳng đứng của chúng theo thứ tự chỉnh sửa nghiêm ngặt, vì vậy bạn có một dấu hiệu trực quan cho thứ tự mà trong đó mọi thứ đã được thực hiện. Trường hợp hai nút trong cùng một cột thì thứ tự là rất rõ ràng. Khi các nút hai cột liền kề độ chênh lệch là nhỏ hơn nhiều vì không có nhu cầu để ngăn chặn các nút chồng chéo lên nhau, và dẫn đến kết quả là thứ tự là ít rõ ràng hơn một chút. Việc tối ưu hóa như vậy là cần thiết để giữ cho các đồ thị phức tạp ở một kích thước hợp lý. Xin lưu ý rằng

lệnh này sử dụng *biên* của các nút trên cạnh *cũ* như là một tham chiếu, tức là cạnh dưới của nút khi biểu đồ được hiển thị với nút cũ nhất ở phía dưới. Cạnh tham chiếu có ý nghĩa quan trọng bởi vì hình dáng nút không phải tất cả là cùng một chiều cao.

## 4.25.2. Thay đổi cách xem

Bởi vì một đồ thị sửa đổi thường khá phức tạp, có một số tính năng có thể được sử dụng điều chỉnh cách xem theo cách bạn muốn. Đây là có sẵn trong trình đơn **Xem** và thanh công cụ.

### Nhóm các nhánh

Các hành vi mặc định (xếp nhóm) có tất cả các hàng được sắp xếp chặt chẽ theo sửa đổi. Kết quả là, các nhánh có các cam kết thừa thớt chiếm toàn bộ một cột cho một số ít thay đổi và đồ thị trở nên rất rộng.

Chế độ này nhóm các thay đổi theo nhánh, do đó không có xếp thứ tự toàn cục cho các bản sửa đổi: Các sửa đổi liên tục trên một chi nhánh sẽ (thường) được hiển thị trong các dòng liên tiếp. Các nhánh con, tuy nhiên, được bố trí trong một cách mà các chi nhánh sau đó sẽ được hiển thị trong cùng một cột phía trên chi nhánh trước đó để giữ cho đồ thị mỏng. Kết quả là, một hàng nhất định có thể chứa các thay đổi từ các phiên bản khác nhau.

### Cũ nhất ở trên

Thông thường, đồ thị cho thấy sửa đổi lâu đời nhất ở phía dưới, và cây phát triển hướng trở lên. Sử dụng tùy chọn này để thay vào đó phát triển từ trên xuống.

### Dóng hàng cây trên c

Khi một đồ thị được chia thành nhiều cây nhỏ hơn, các cây có thể xuất hiện hoặc trong thứ tự sửa đổi tự nhiên, hoặc được đóng hàng ở dưới cùng của cửa sổ, tùy thuộc vào bạn đang sử dụng tùy chọn **Xếp Nhóm Các Chi nhánh** hay không. Sử dụng tùy chọn này để thay vào đó phát triển tất cả các cây từ trên xuống.

### Giảm các đường cắt ngang

Tùy chọn này thường được kích hoạt và tránh hiển thị đồ thị với rất nhiều dòng chéo nhau gây nhầm lẫn. Tuy nhiên điều này cũng có thể làm cho bố cục các cột xuất hiện trong ít địa điểm lô gic hơn, ví dụ trong một đường chéo chứ không phải là một cột, và đồ thị có thể yêu cầu một diện tích lớn hơn để vẽ. Nếu đây là một vấn đề, bạn có thể vô hiệu hóa các tùy chọn từ trình đơn **Xem**.

### Các tên đường dẫn khác biệt

Các tên đường dẫn dài có thể chiếm rất nhiều không gian và làm cho các hộp nút trở nên rất lớn. Sử dụng tùy chọn này để hiển thị chỉ một phần thay đổi của một đường dẫn, thay thế phần phổ biến với các dấu chấm. Ví dụ, nếu bạn tạo ra một chi nhánh `/branches/1.2.x/doc/html` từ `/trunk/doc/html` chi nhánh có thể được hiển thị trong hình thức thu gọn `/branches/1.2.x /..` bởi vì hai cấp cuối cùng, `doc` và `html`, không thay đổi.

### Hiện tất cả các bản sửa đ

Điều này làm những gì bạn mong đợi và cho thấy tất cả các sửa đổi nơi một vài thứ (trong cây mà bạn đang vẽ đồ thị) đã thay đổi. Đối với các lịch sử dài điều này có thể tạo ra một biểu đồ thực sự rất lớn.

### Hiện các sửa đổi CHÍNH

Điều này đảm bảo rằng bản sửa đổi mới nhất trên tất cả các chi nhánh luôn luôn được hiển thị trên đồ thị.

### Các nguồn bản sao chính xác

Khi một chi nhánh/thẻ được tạo ra, hành vi mặc định là hiển thị các chi nhánh như được lấy từ nút cuối cùng nơi thay đổi đã được thực hiện. Nói một cách chặt chẽ thì cái này không chính xác bởi vì khi các nhánh thường được làm từ bản CHÍNH hiện tại chứ không phải một sửa đổi nhất định. Vì vậy, có thể hiển thị các sửa đổi chính xác hơn (nhưng ít hữu ích) mà được sử dụng để tạo ra các bản sao. Lưu ý rằng bản sửa đổi này có thể mới hơn so với bản sửa đổi CHÍNH của chi nhánh nguồn.

### Gấp thẻ

Khi một dự án có nhiều thẻ, hiển thị mỗi thẻ như một nút riêng biệt trên đồ thị chiếm rất nhiều không gian và che khuất cấu trúc phát triển nhánh thú vị hơn. Đồng thời, bạn có thể cần để có thể truy cập vào các nội dung thẻ dễ dàng vì vậy mà bạn có thể so sánh các bản sửa đổi. Tùy chọn này che giấu các nút cho các thẻ và thay vào đó hiển thị chúng trong tooltip cho nút mà chúng đã được sao chép từ đó. Một biểu tượng thẻ trên phía bên phải của nút nguồn cho thấy rằng thẻ đã được tạo. Điều này giúp đơn giản hóa màn xem.

Lưu ý rằng nếu một thẻ là chính nó được sử dụng như là nguồn cho một bản sao, có lẽ là một chi nhánh mới dựa trên một thẻ, thì thẻ đó sẽ được hiển thị như một nút riêng biệt chứ không phải được gộp.

#### Giấu các đường dẫn bị xóa

Dấu các đường dẫn không còn có mặt tại sửa đổi CHÍNH của kho, ví dụ như các nhánh bị xóa.

Nếu bạn đã chọn tùy chọn **Gấp thẻ** thì chi nhánh đã bị xóa từ những thẻ nào đã được lấy vẫn sẽ được hiển thị, nếu không các thẻ sẽ biến mất. Sửa đổi cuối cùng được gán thẻ sẽ được hiển thị trong màu sắc được sử dụng cho các nút xóa thay vì hiển thị một phiên bản xóa riêng biệt.

Nếu bạn chọn lựa chọn **Ẩn thẻ** thì những nhánh này sẽ biến mất một lần nữa bởi chúng không cần thiết để hiển thị các thẻ.

#### Ẩn các nhánh không sử dụng

Ẩn các nhánh không có thay đổi nào được cam kết vào tập tin hay thư mục con tương ứng. Điều này không nhất thiết phải chỉ ra rằng chi nhánh không được sử dụng, chỉ là không có thay đổi nào được thực hiện cho phần *này* của nó.

#### Hiển thị sửa đổi WC

Đánh dấu sự sửa đổi trên đồ thị tương ứng với phiên bản cập nhật của các mục mà bạn đã nạp đồ thị cho chúng. Nếu bạn chỉ cần cập nhật, đây sẽ CHÍNH, nhưng nếu những người khác cam kết các thay đổi kể từ khi bạn cập nhật WC của bạn lần cuối có thể thấp hơn một phiên bản. Nút này được đánh dấu bằng một phác thảo in đậm.

#### Hiển thị các sửa đổi WC

Nếu WC của bạn có các thay đổi cục bộ, tùy chọn này rút nó ra như là một nút riêng biệt hình elip, được liên kết trở lại nút mà WC của bạn được cập nhật lần cuối. Màu phác thảo mặc định là màu đỏ. Bạn có thể cần phải làm mới đồ thị bằng cách sử dụng **F5** để nắm bắt những thay đổi gần đây.

#### Lọc

Đôi khi đồ thị sửa đổi chứa nhiều bản sửa đổi hơn bạn muốn xem. Tùy chọn này sẽ mở ra một hộp thoại cho phép bạn hạn chế phạm vi các phiên bản được hiển thị, và để ẩn các đường dẫn cụ thể bằng tên.

Nếu bạn giấu một đường dẫn đặc biệt và nút đó có các nút con, các nút con sẽ được hiển thị như một cây riêng biệt. Nếu bạn muốn ẩn tất cả các nút con, sử dụng hộp kiểm **Xóa toàn bộ một (hoặc nhiều) cây con**.

#### Các sọc của cây

Trong trường hợp đồ thị có nhiều cây, đôi khi hữu ích để sử dụng xen kẽ các màu sắc trên nền để giúp phân biệt giữa các cây.

#### Hiển thị tổng quan

Hiển thị một bức tranh nhỏ của toàn bộ đồ thị, với cửa sổ xem hiện tại như là một hình chữ nhật mà bạn có thể kéo. Điều này cho phép bạn định hướng đến đồ thị dễ dàng hơn. Lưu ý rằng cho các đồ thị rất lớn, tổng quan có thể trở nên vô dụng do yếu tố phóng to và do đó sẽ không được hiển thị trong các trường hợp như vậy.

### 4.25.3. Sử dụng Đồ thị

Để làm cho nó dễ dàng hơn để định hướng một đồ thị lớn, sử dụng cửa sổ tổng quan. Cái này cho thấy toàn bộ đồ thị trong một cửa sổ nhỏ, với phần đang hiển thị được làm nổi bật. Bạn có thể kéo các khu vực được làm nổi bật để thay đổi khu vực hiển thị.

Ngày sửa đổi, tác giả và bình luận được thể hiện trong một hộp gợi ý bất cứ khi nào di chuột lên một hộp sửa đổi.

Nếu bạn chọn hai phiên bản (Sử dụng **Ctrl**-chuột trái), bạn có thể sử dụng trình đơn ngữ cảnh để hiển thị sự khác biệt giữa các bản sửa đổi. Bạn có thể chọn để hiển thị các khác biệt như tại các điểm tạo ra chi nhánh, nhưng thông thường bạn sẽ muốn thể hiện các sự khác biệt tại điểm cuối chi nhánh, tức là ở sửa đổi CHÍNH.

Bạn có thể xem các khác biệt như là một tập tin khác biệt-thống nhất, trong đó cho thấy tất cả các sự khác biệt trong một tập tin đơn lẻ với bối cảnh tối thiểu. Nếu bạn lựa chọn **Trình đơn Ngữ cảnh** → **So Sánh các Sửa**

Đổi bạn sẽ được trình bày với một danh sách các tập tin được thay đổi. Kích đúp vào một tên tập tin để lấy cả hai bản sửa đổi của tập tin và so sánh chúng bằng cách sử dụng công cụ khác biệt trực quan.

Nếu bạn nhấp chuột phải trên một bản sửa đổi, bạn có thể sử dụng Trình Đơn Ngữ Cảnh → Hiện Tường Trình để xem lịch sử.

Bạn cũng có thể sáp nhập các thay đổi trong các bản sửa đổi được lựa chọn vào một bản sao làm việc khác. Một hộp thoại lựa chọn thư mục cho phép bạn chọn bản sao làm việc để sáp nhập vào, nhưng sau đó không có hộp thoại xác nhận, cũng không có bất kỳ cơ hội để thử một thử nghiệm hợp nhất. Đó là một ý tưởng tốt để kết hợp vào một bản sao hoạt động chưa sửa đổi để bạn có thể hoàn nguyên các thay đổi nếu việc này không xong! Đây là một tính năng hữu ích nếu bạn muốn kết hợp các phiên bản được lựa chọn từ một chi nhánh sang một chi nhánh khác.



### Tìm hiểu để đọc đồ thị sửa đổi

Lần đầu tiên người dùng có thể bị ngạc nhiên bởi thực tế là đồ thị sửa đổi cho thấy một cái gì đó không khớp với mô hình tương tượng của người dùng. Nếu một bản sửa đổi thay đổi nhiều bản sao hay chi nhánh của một tập tin hoặc thư mục, ví dụ là vậy, thì sẽ có nhiều nút cho bản sửa đổi đơn lẻ đó. Đó là một thói quen tốt để bắt đầu với các tùy chọn tận cùng bên trái trong thanh công cụ và tùy chỉnh đồ thị từng bước cho đến khi nó đến gần với mô hình tương tư của bạn.

Tất cả các tùy chọn bộ lọc cố gắng mất càng ít thông tin càng tốt. Điều đó có thể làm cho một số nút thay đổi màu sắc của chúng, ví dụ là vậy. Bất cứ khi nào kết quả là bất ngờ, hoàn tác hoạt động cuối cùng của bộ lọc và cố gắng hiểu những gì là đặc biệt về bản sửa đổi hoặc chi nhánh cụ thể đó. Trong hầu hết trường hợp, kết quả dự kiến ban đầu của hoạt động lọc hoặc là sẽ không chính xác hoặc gây hiểu nhầm.

#### 4.25.4. Làm mới màn hình xem

Nếu bạn muốn kiểm tra máy chủ một lần nữa cho thông tin mới hơn, bạn chỉ đơn giản là có thể làm mới màn hình xem bằng cách sử dụng **F5**. Nếu bạn đang sử dụng bộ nhớ cache tường trình (được kích hoạt theo mặc định), điều này sẽ kiểm tra kho lưu trữ cho các cam kết mới hơn và lấy chỉ những cái mới. Nếu bộ nhớ cache tường trình trong chế độ ngoại tuyến, hành động này cũng sẽ cố gắng để trở lại trực tuyến.

Nếu bạn đang sử dụng bộ nhớ cache tường trình và bạn nghĩ nội dung thông điệp hay tác giả có thể đã thay đổi, bạn nên sử dụng hộp thoại tường trình để làm mới các thông điệp mà bạn cần. Bởi vì đồ thị sửa đổi hoạt động từ gốc của kho, chúng ta sẽ làm mất hiệu lực bộ toàn bộ nhớ cache tường trình, và việc lấp đầy lại nó có thể mất một thời gian *rất* dài.

#### 4.25.5. Cắt tỉa các cây

Một cây lớn có thể là khó khăn để định hướng và đôi khi bạn sẽ muốn ẩn các phần của nó, hoặc phá vỡ nó thành một khu rừng các cây nhỏ hơn. Nếu bạn di chuyển chuột trên một nút nơi một liên kết nút đi vào hoặc rời khỏi nút, bạn sẽ thấy một hoặc nhiều nút bật lên mà cho phép bạn làm điều này.



Nhấp chuột vào nút bấm dấu trừ để sập các cây con gắn v.



Click vào nút dấu cộng để mở rộng một cây sụp đổ. Khi cây đã bị sụp đổ, nút này vẫn được nhìn thấy để chỉ ra cây con ẩn.





Nhấp chuột vào nút chéo để phân chia các con cây gắn theo và hiển thị nó như một cây riêng biệt trên đồ thị.

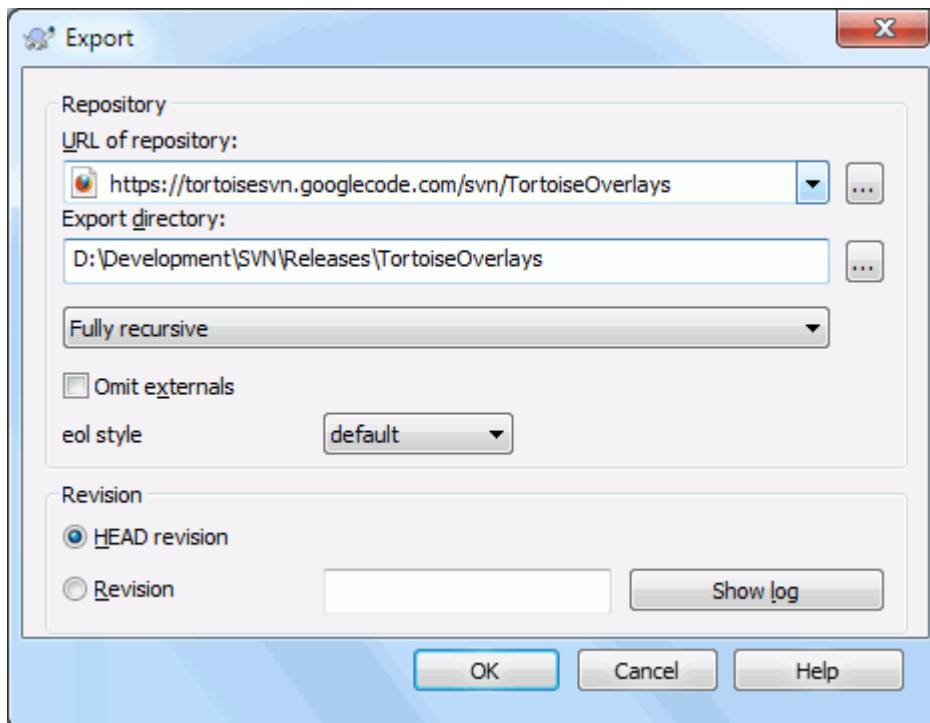


Nhấp chuột vào nút tròn để gắn lại một cây được chia tách. Khi một cây đã được chia đi, nút này vẫn được nhìn thấy để cho biết rằng có một cây con riêng biệt.

Nhấp chuột vào nền đồ thị cho trình đơn ngữ cảnh chính, trong đó cung cấp các tùy chọn để **Mở rộng tất cả** và **Tham gia tất cả**. Nếu không có chi nhánh nào đã bị sập, tách, trình đơn ngữ cảnh sẽ không được hiển thị.

## 4.26. Xuất khẩu một bản sao làm việc Subversion

Thỉnh thoảng bạn muốn có một bản sao sạch của cây làm việc của bạn mà không có thư mục `.svn`, ví dụ để tạo một zipped tarball của nguồn của bạn, hoặc xuất khẩu đến một máy chủ. Thay vì tạo một bản sao và xóa các thư mục `.svn` một cách thủ công, TortoiseSVN cung cấp lệnh TortoiseSVN → **Xuất khẩu...** Xuất khẩu từ một URL và xuất khẩu từ bản sao làm việc được đối xử khác nhau một chút.



**Hình 4.59. Hộp thoại xuất khẩu từ URL**

Nếu bạn thực hiện lệnh này trên một thư mục chưa phiên bản, TortoiseSVN sẽ giả định rằng thư mục được chọn là mục tiêu, và mở một hộp thoại cho bạn nhập URL và bản sửa đổi để xuất khẩu. Hộp thoại này có các tùy chọn để chỉ xuất khẩu các thư mục cấp cao nhất, bỏ qua các tham chiếu bên ngoài, và để ghi đè lên phong cách cuối dòng cho các tập tin có thuộc tính `svn:eol-style` được thiết lập.

Tất nhiên bạn có thể xuất khẩu trực tiếp từ kho lưu trữ. Sử dụng Trình duyệt Kho để định hướng đến các cây con có liên quan trong kho của bạn, sau đó sử dụng Trình Đơn Ngữ Cảnh → **Xuất khẩu**. Bạn sẽ nhận được hộp thoại **Xuất khẩu từ URL** như mô tả ở trên.

Nếu bạn thi hành lệnh này trên bản sao làm việc bạn sẽ được hỏi chỗ để lưu bản làm việc *sạch* mà không có thư mục `.svn`. Theo mặc định, chỉ có những tập tin được đánh phiên bản được xuất khẩu, nhưng bạn có thể dùng hộp kiểm **Xuất khẩu cả những tập tin không phiên bản** để xuất khẩu cả những tập tin không phiên bản tồn tại trong bản làm việc của bạn nhưng không có trong kho. Những tham chiếu bên ngoài sử dụng `svn:externals` có thể được bỏ qua nếu đòi hỏi.

Một cách khác để xuất khẩu một bản làm việc là kéo lê chuột phải thư mục bản sao đang làm và chọn Trình Đơn Ngữ Cảnh → SVN Xuất khẩu các mục có phiên bản tại đây hoặc Trình Đơn Ngữ Cảnh → SVN Xuất khẩu tất cả các mục tại đây or Trình Đơn Ngữ Cảnh → SVN Xuất khẩu các mục được thay đổi tại đây. Tùy chọn thứ hai cũng bao gồm các tập tin không phiên bản. Tùy chọn thứ ba chỉ xuất khẩu các mục được thay đổi, nhưng giữ nguyên cấu trúc thư mục.

Khi xuất khẩu từ một bản sao làm việc, nếu thư mục đích đã có một thư mục cùng tên như một trong những bản đang xuất khẩu, bạn sẽ được cung cấp tùy chọn để ghi đè lên các nội dung hiện có, hoặc tạo một thư mục mới với một tên tự động tạo ra, ví dụ như `Mục tiêu (1)`.



### Xuất khẩu các tập tin đơn l

Hộp thoại xuất khẩu không cho phép khẩu các tập tin xuất đơn lẻ, mặc dù Subversion có thể.

Để xuất khẩu các tập tin đơn lẻ với TortoiseSVN, bạn phải sử dụng trình duyệt kho ( [Phần 4.24, “Trình duyệt Kho”](#) ). Đơn giản chỉ cần kéo tập tin bạn muốn xuất khẩu từ trình duyệt kho nơi mà bạn muốn họ trong explorer, hoặc sử dụng trình đơn ngữ cảnh trong trình duyệt kho lưu trữ để xuất khẩu các tập tin.



### Xuất khẩu một Cây Biến đổi

Nếu bạn muốn xuất khẩu một bản sao của cấu trúc cây dự án của bạn nhưng chỉ chứa các tập tin đã thay đổi trong một phiên bản cụ thể, hoặc giữa bất kỳ hai phiên bản, sử dụng tính năng so sánh các bản sửa đổi được mô tả trong [Phần 4.10.3, “So sánh các thư mục”](#).

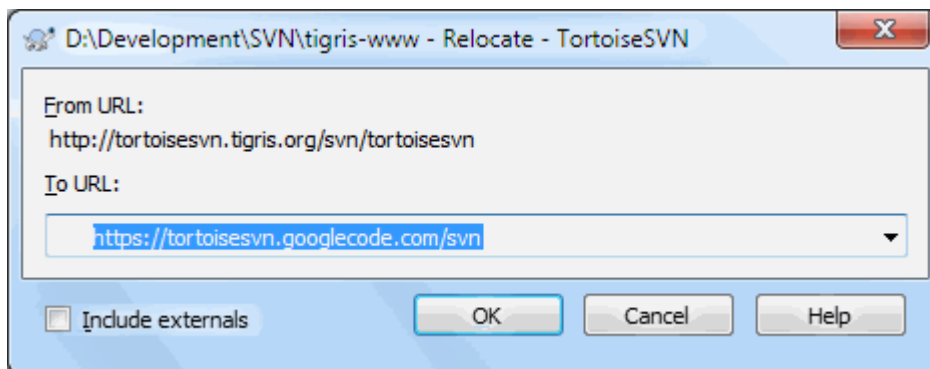
Nếu bạn muốn xuất khẩu cấu trúc cây của bản sao làm việc nhưng chỉ chứa các tập tin được sửa đổi cục bộ, tham khảo SVN Xuất khẩu các mục được thay đổi tại đây ở phía trên.

#### 4.26.1. Loại bỏ một bản sao làm việc từ kiểm soát phiên bản

Thình thoảng bạ có một bản sao làm việc mà bạn muốn chuyển đổi về lại thành một thư mục bình thường mà không có thư mục `.svn`. Tất cả bạn cần phải làm là xóa thư mục `.svn` từ gốc bản sao làm việc.

Một cách khác là bạn có thể xuất khẩu thư mục vào chính nó. Trong Windows Explorer dùng chuột phải kéo lê thư mục gốc của bản làm việc từ bảng thư mục vào chính nó trong bảng thư mục. TortoiseSVN phát hiện trường hợp đặc biệt này và hỏi nếu bạn muốn bỏ phiên bản khỏi bản sao làm việc. Nếu bạn trả lời *vâng* thư mục điều khiển sẽ bị bỏ và bạn sẽ có một cây thư mục trơn, không đánh phiên bản.

#### 4.27. Di dời một bản sao làm việc



Hình 4.60. Hộp Tho Di Dời

Nếu kho của bạn đã vì một số lý do nào đó thay đổi vị trí (IP/URL). Có lẽ bạn đang thậm chí còn bị mắc kẹt và không thể cam kết và bạn không muốn kiểm xuất bản sao làm việc của bạn một lần nữa từ vị trí mới và di chuyển

tất cả dữ liệu đã thay đổi của bạn trở lại vào bản sao làm việc mới, `TortoiseSVN → Di dời` là lệnh bạn đang tìm kiếm. Về cơ bản nó làm rất ít: nó ghi lại tất cả các URL có liên quan với mỗi tập tin và thư mục với URL mới.

## Ghi chú

Tác vụ này chỉ làm việc trên bản sao làm việc gốc. Vì vậy, các mục trình đơn ngữ cảnh chỉ được hiển thị cho các gốc của bản sao làm việc.

Bạn có thể ngạc nhiên khi thấy rằng TortoiseSVN liên hệ với kho lưu trữ như là một phần của hoạt động này. Tất cả những gì nó đang làm là thực hiện một số kiểm tra đơn giản để đảm bảo rằng các URL mới thực sự đề cập đến cùng một kho lưu trữ như là bản sao làm việc hiện tại.



## Cảnh báo

*Đây là một hoạt động rất không thường xuyên được sử dụng. Lệnh di dời chỉ được sử dụng nếu các URL của gốc kho đã thay đổi. Lý do có thể là:*

- Địa chỉ IP của máy chủ đã thay đổi.
- Giao thức đã thay đổi (ví dụ như `http://` sang `https://`).
- Các đường dẫn gốc của kho lưu trữ trong các thiết lập máy chủ đã thay đổi.

Nói cách khác, bạn cần phải di dời khi bản sao của bạn làm việc chỉ tới cùng một vị trí trong cùng một kho lưu trữ, nhưng bản thân kho lưu trữ đã bị di chuyển.

Nó không áp dụng nếu:

- Bạn muốn di chuyển đến một kho lưu trữ Subversion khác. Trong trường hợp đó, bạn nên thực hiện một kiểm xuất sạch từ vị trí kho lưu trữ mới.
- Bạn muốn chuyển sang một chi nhánh hoặc thư mục khác trong cùng một kho lưu trữ. Để làm điều đó bạn nên sử dụng `TortoiseSVN → Chuyển đổi ...`. Đọc [Phần 4.19.3, “Kiểm xu hoặc Chuyển mạch ...”](#) để biết thêm thông tin.

Nếu bạn sử dụng di dời trong một trong các trường hợp trên, nó sẽ làm hỏng bản sao của bạn làm việc và bạn sẽ nhận được nhiều thông báo lỗi không thể giải thích được trong khi cập nhật, cam kết, vv Một khi đã xảy ra, khắc phục duy nhất là làm một kiểm xuất mới.

## 4.28. Tích hợp với Hệ thống Theo dõi Lỗi / Trình theo dõi Vấn đề

Nó là rất phổ biến trong Phát triển Phần mềm cho các thay đổi có liên quan đến một lỗi cụ thể hoặc ID của vấn đề. Người sử dụng của hệ thống theo dõi lỗi (bộ theo dõi vấn đề) sẽ liên kết những thay đổi mà họ thực hiện trong Subversion với một ID cụ thể trong bộ theo dõi vấn đề của họ. Hầu hết các bộ theo dõi vấn đề do đó cung cấp một kịch bản hook trước cam kết mà sẽ phân tích cú pháp thông điệp tường trình để tìm ID của lỗi mà các cam kết có liên quan. Điều này có phần dễ bị lỗi vì nó phụ thuộc vào người sử dụng để viết thông điệp tường trình đúng cách mà kịch bản hook trước cam kết có thể phân tích nó một cách chính xác.

TortoiseSVN có thể giúp người sử dụng theo hai cách:

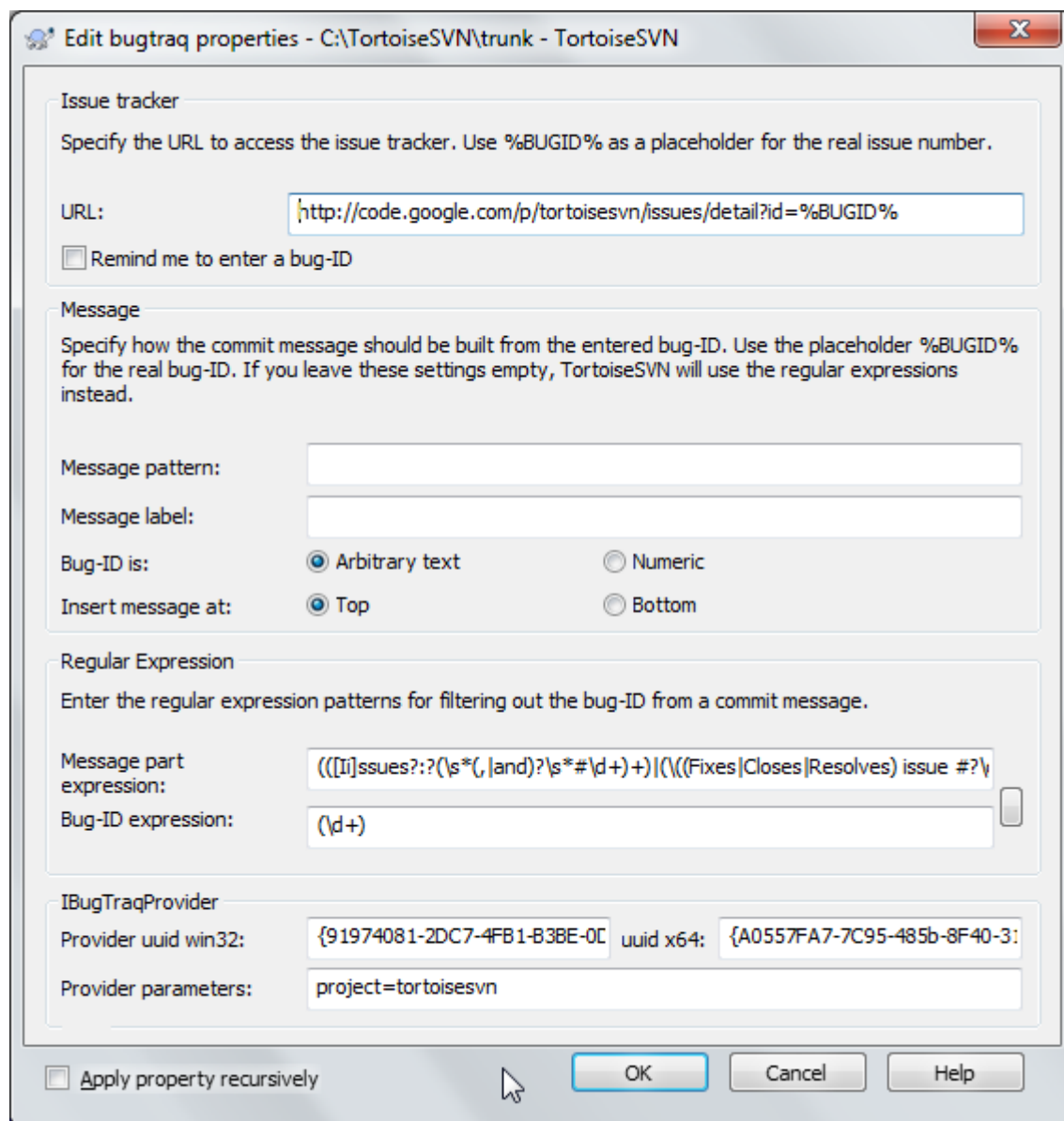
1. Khi người dùng nhập vào một thông điệp tường trình, một dòng được định nghĩa tốt bao gồm cả số vấn đề liên quan đến cam kết có thể được thêm vào tự động. Điều này làm giảm nguy cơ mà người dùng nhập vào số lượng vấn đề theo cách mà công cụ theo dõi lỗi không thể phân tích một cách chính xác.

Hoặc TortoiseSVN có thể làm nổi bật một phần của thông điệp tường trình đã nhập được công nhận bởi bộ theo dõi vấn đề. Bằng cách đó người sử dụng biết rằng các thông điệp tường trình có thể được phân tích một cách chính xác.

2. Khi người dùng duyệt các thông điệp tường trình, TortoiseSVN tạo ra một liên kết từ mỗi ID lỗi trong các tin nhắn đăng nhập mà sẽ chuyển trình duyệt tới vấn đề được đề cập.

#### 4.28.1. Thêm số Vấn đề vào Thông điệp Tường tr

Bạn có thể tích hợp một công cụ theo dõi lỗi do bạn chọn trong TortoiseSVN. Để làm điều này, bạn phải xác định một số thuộc tính, bắt đầu với `bugtraq:` . Chúng phải được thiết lập trên thư mục: ( [Phần 4.17](#), “**Thiết lập Dự án**” )



**Hình 4.61. Hội Thoại Thuộc Tính Bugtraq**

Khi bạn chỉnh sửa bất kỳ các thuộc tính Bugtraq một trình soạn thảo thuộc tính đặc biệt được sử dụng để làm cho nó dễ dàng hơn để thiết lập các giá trị thích hợp.

Có hai cách để tích hợp TortoiseSVN với bộ theo dõi vấn đề. Một là dựa trên các chuỗi đơn giản, cách khác là dựa trên *biểu thức thông thường*. Các thuộc tính được sử dụng bởi cả hai cách tiếp cận trên là:

`bugtraq:url`

Thiết lập thuộc tính này vào URL của công cụ theo dõi lỗi của bạn. Nó phải được mã hóa theo URI đúng và nó có chứa `%BUGID%` . `%BUGID%` được thay thế với số vấn đề mà bạn đã nhập. Điều này cho phép

TortoiseSVN hiển thị một liên kết trong hộp thoại đăng nhập, vì vậy khi bạn đang tìm kiếm các bản sửa đổi, bạn có thể nhảy trực tiếp đến công cụ theo dõi lỗi của bạn. Bạn không cần phải cung cấp thuộc tính này, nhưng sau đó TortoiseSVN chỉ hiển thị số phát hành và không phải là liên kết đến nó. ví dụ như dự án TortoiseSVN đang sử dụng `http://issues.tortoisesvn.net/?do=details&id=%BUGID`.

Bạn cũng có thể sử dụng các URL tương đối của những người tuyệt đối. Điều này rất hữu ích khi bộ theo dõi vấn đề của bạn ở trên cùng một tên miền/máy chủ như là kho lưu trữ nguồn của bạn. Trong trường hợp tên miền bao giờ thay đổi, bạn không phải điều chỉnh thuộc tính `bugtraq:url`. Có hai cách để chỉ định một URL có liên quan:

Nếu nó bắt đầu với chuỗi `^/` nó được giả định là tương đối đến chủ kho. Ví dụ, `^/../?do=details&id=%BUGID%` sẽ được phân giải thành `http://tortoisesvn.net/?do=details&id=%BUGID%` nếu kho của bạn nằm trên `http://tortoisesvn.net/svn/trunk/`.

URL bắt đầu với chuỗi `/` được giả định là tương đối đến tên chủ của máy chủ. Ví dụ `/do=?details&id=%BUGID%` sẽ phân giải thành `http://tortoisesvn.net/?do=details&id=%BUGID%` nếu kho của bạn được nằm bất cứ nơi nào trên `http://tortoisesvn.net`.

`bugtraq:warnifnoissue`

Đặt nó thành `đúng`, nếu bạn muốn TortoiseSVN cảnh báo bạn bởi vì một trường văn bản số vấn đề bị trống. Giá trị hợp lệ là `đúng/sai`. *Nếu không được xác định, sai sẽ được giả định.*

#### 4.28.1.1. Số vấn đề trong Hộp Văn bản

Trong cách tiếp cận đơn giản, TortoiseSVN cho người sử dụng thấy một trường nhập liệu riêng biệt, nơi một lỗi ID có thể được nhập vào. Sau đó, một dòng riêng biệt được nối đuôi/nối đầu vào thông điệp tường trình mà người dùng nhập vào.

`bugtraq:message`

Thuộc tính này kích hoạt hệ thống theo dõi lỗi trong chế độ *Trường Nhập liệu*. Nếu thuộc tính này được thiết lập, thì TortoiseSVN sẽ nhắc nhở bạn nhập vào một số vấn đề khi bạn cam kết thay đổi của bạn. Nó được sử dụng để thêm một dòng ở cuối thông điệp tường trình. Nó phải có `%BUGID%`, mà được thay thế bằng số vấn đề trên cam kết. Điều này đảm bảo rằng tường trình cam kết của bạn có chứa một tham chiếu đến số vấn đề mà luôn luôn là một định dạng nhất quán và có thể được phân tích bằng công cụ theo dõi lỗi của bạn để liên kết số vấn đề với một cam kết cụ thể. Một ví dụ là bạn có thể sử dụng `Vấn đề: %BUGID%`, nhưng điều này phụ thuộc vào Công Cụ của bạn.

`bugtraq:label`

Văn bản này được thể hiện bởi TortoiseSVN trên hộp thoại cam kết để gắn nhãn cho hộp biên tập, nơi bạn nhập số phát hành. Nếu nó không được thiết lập, `ID Lỗi / Số Vấn đề:` sẽ được hiển thị. Hãy nhớ rằng của số đó sẽ không được thay đổi kích cỡ để phù hợp với nhãn này, vì vậy giữ cho kích thước của nhãn dưới 20-25 ký tự.

`bugtraq:number`

Nếu được thiết lập `đúng` chỉ có các số được cho phép trong trường văn bản số vấn đề. Một ngoại lệ là các dấu phẩy, vì vậy bạn có thể dấu phẩy tách một số con số. Các giá trị hợp lệ là `đúng/sai`. *Nếu không xác định, giá trị đúng được giả định.*

`bugtraq:append`

Thuộc tính này xác định nếu ID của lỗi được nối (`đúng`) vào cuối thông điệp tường trình hoặc chèn (`sai`) vào đầu thông điệp tường trình. Giá trị hợp lệ `đúng/sai`. *Nếu không được định nghĩa, đúng được giả định, để dự án hiện có không bị phá vỡ.*

#### 4.28.1.2. Số Vấn Đề Sử Dụng Biểu Thức Thông Thư

Trong cách tiếp cận với *biểu thức thông thường* TortoiseSVN không hiển thị một trường nhập liệu riêng biệt, nhưng đánh dấu một phần của thông điệp tường trình mà người dùng nhập vào được nhận ra bởi bộ theo dõi vấn đề. Điều này được thực hiện khi người sử dụng viết thông điệp tường trình. Điều này cũng có nghĩa rằng các lỗi

ID có thể ở bất cứ nơi nào bên trong một thông điệp tường trình! Phương pháp này linh hoạt hơn nhiều, và là một trong những cách được sử dụng bởi bản thân dự án TortoiseSVN.

`bugtraq:logregex`

Thuộc tính này kích hoạt hệ thống theo dõi lỗi trong chế độ *Regex*. Nó chứa hoặc là một biểu thức thông thường duy nhất, hoặc hai biểu thức thông thường cách nhau bằng một dòng mới.

Nếu hai biểu thức được thiết lập, thì biểu thức đầu tiên được sử dụng như bộ lọc trước để tìm các biểu thức có chứa lỗi ID. Biểu thức thứ hai sau đó chiết xuất lỗi ID trần từ kết quả của regex đầu tiên. Điều này cho phép bạn sử dụng một danh sách các lỗi ID và các biểu thức ngôn ngữ tự nhiên nếu bạn muốn. ví dụ như bạn có thể sửa chữa một số lỗi và bao gồm một chuỗi gì đó như thế này: “ Thay đổi này giải quyết vấn đề #23, #24 và #25 ”.

Nếu bạn muốn bắt lỗi ID như được sử dụng trong các biểu thức trên bên trong một thông điệp tường trình, bạn có thể sử dụng các chuỗi regex sau đây, là những cái được sử dụng bởi dự án TortoiseSVN: `[Ii]ssues: (?\\S*(, |and)\\s*#\\d+)+ và (\\d+)`.

Biểu thức đầu tiên chọn ra các “ vấn đề #23, #24 và #25 ” từ thông điệp tường trình xung quanh. Regex thứ hai chiết xuất từ các số thập phân từ đầu ra của regex đầu tiên, vì vậy nó sẽ trả về “ 23 ”, “ 24 ” và “ 25 ” để sử dụng như là ID của lỗi.

Bề regex đầu tiên ra một chút, nó phải bắt đầu với từ “ issue ”, có thể được viết hoa. Đây là tùy chọn theo sau là một chữ “ s ” (nhiều hơn một vấn đề), và tùy chọn dấu hai chấm. Tiếp theo là một hoặc nhiều nhóm mà mỗi nhóm không có hoặc có nhiều hơn khoảng trắng ở phía trước, dấu phẩy tùy chọn hoặc “ và ” và nhiều không gian tùy chọn hơn nữa. Cuối cùng là dấu “ # ” bắt buộc và một số thập phân là bắt buộc.

Nếu chỉ một biểu thức được thiết lập, thì các lỗi ID trần phải được xuất hiện trong các nhóm của chuỗi regex. Ví dụ: `[Ii]ssue(s?)#(\\d+)` Phương pháp này là cần thiết bởi một vài bộ theo dõi vấn đề, ví dụ như trac, nhưng nó là khó để xây dựng các regex. Chúng tôi khuyên rằng bạn chỉ sử dụng phương pháp này nếu tài liệu của bộ theo dõi vấn đề của bạn bảo bạn l.

Nếu bạn không quen với các biểu thức thông thường, hãy nhìn vào việc giới thiệu tại [http://en.wikipedia.org/wiki/Regular\\_expression](http://en.wikipedia.org/wiki/Regular_expression), và tài liệu trực tuyến và hướng dẫn tại <http://www.regular-expressions.info/>.

Nó không phải luôn luôn dễ dàng để có được regex do đó để giúp đỡ thì có một hộp thoại thử nghiệm được xây dựng vào hộp thoại thuộc tính Bugtraq. Nhấp chuột vào nút bên phải của hộp chỉnh sửa để đưa nó lên. Ở đây bạn có thể nhập một số văn bản kiểm tra, và mỗi thay đổi mỗi regex để xem kết quả. Nếu regex là không hợp lệ thì màu nền hộp chỉnh sửa sẽ đổi thành màu đỏ.

Nếu cả hai thuộc tính `bugtraq:message` và `bugtraq:logregex` được thiết lập, `logregex` sẽ được ưu tiên.



## Mẹo

Ngay cả khi bạn không có bộ theo dõi vấn đề với hook trước-cam kết phân tích cú pháp thông điệp tường trình của bạn, bạn vẫn có thể sử dụng cái này để biến các vấn đề được đề cập trong các thông điệp tường trình của bạn vào các liên kết!

Và ngay cả khi bạn không cần các liên kết, những con số vấn đề hiển thị như là một cột riêng biệt trong hộp thoại tường trình, làm cho nó dễ dàng hơn để tìm thấy những thay đổi liên quan đến một vấn đề cụ thể.

Một số thuộc tính `tsvn:` đòi hỏi phải có giá trị đúng/sai. TortoiseSVN cũng hiểu `vâng` như một từ đồng nghĩa với `đúng` và `không` như một từ đồng nghĩa với `sai`.



## Thiết lập các Thuộc Tính trên các Thư Mục

Những thuộc tính này phải được thiết lập trên các thư mục hệ thống để làm việc. Khi bạn cam kết một tập tin hoặc thư mục các thuộc tính được đọc từ các thư mục đó. Nếu các thuộc tính không được

tìm thấy ở đó, TortoiseSVN sẽ tìm kiếm trở lên thông qua cây thư mục để tìm thấy chúng cho đến khi nó đến một thư mục chưa phiên bản, hoặc gốc cây (ví dụ: `C:\`) được tìm thấy. Nếu bạn có thể chắc chắn rằng mỗi người dùng chỉ kiểm xuất từ ví dụ như `trunk/` và không phải là một thư mục con, thì nó là đủ nếu bạn thiết lập các thuộc tính trên `trunk/`. Nếu bạn không thể chắc chắn, bạn nên thiết lập các thuộc tính một cách đệ quy trên mỗi thư mục con. Một thuộc tính thiết lập sâu hơn trong hệ thống phân cấp dự án sẽ ghi đè các thiết lập trên các mức độ cao hơn (gần với `trunk/`).

Đối với thuộc tính *chi* của dự án, tức là `tsvn:`, `bugtraq:` và `webviewer:` bạn có thể sử dụng hộp kiểm **Đệ quy** để thiết lập thuộc tính cho tất cả các thư mục con trong hệ thống phân cấp, mà không cần đặt nó trên tất cả các tập tin.

Khi bạn thêm mới các thư mục con vào một bản sao làm việc bằng cách sử dụng TortoiseSVN, bất kỳ thuộc tính dự án có trong thư mục mẹ sẽ tự động được thêm vào thư mục con.



## Không có Thông Tin về Bộ Theo Dõi Vấn Đề từ Trình Duyệt Kho

Bởi vì sự tích hợp theo dõi vấn đề phụ thuộc vào truy cập vào các thuộc tính Subversion, bạn sẽ chỉ thấy các kết quả khi sử dụng một bản sao làm việc được kiểm xuất. Lấy thuộc tính từ xa là một hoạt động chậm, vì vậy bạn sẽ không thấy tính năng này hoạt động động từ trình duyệt kho, trừ khi bạn bắt đầu trình duyệt kho từ bản sao làm việc của bạn. Nếu bạn bắt đầu trình duyệt kho bằng cách nhập vào URL của kho bạn sẽ không thấy tính năng này.

Vì lý do tương tự, các thuộc tính dự án sẽ không được truyền tự động khi một thư mục con được thêm vào bằng cách sử dụng trình duyệt kho.

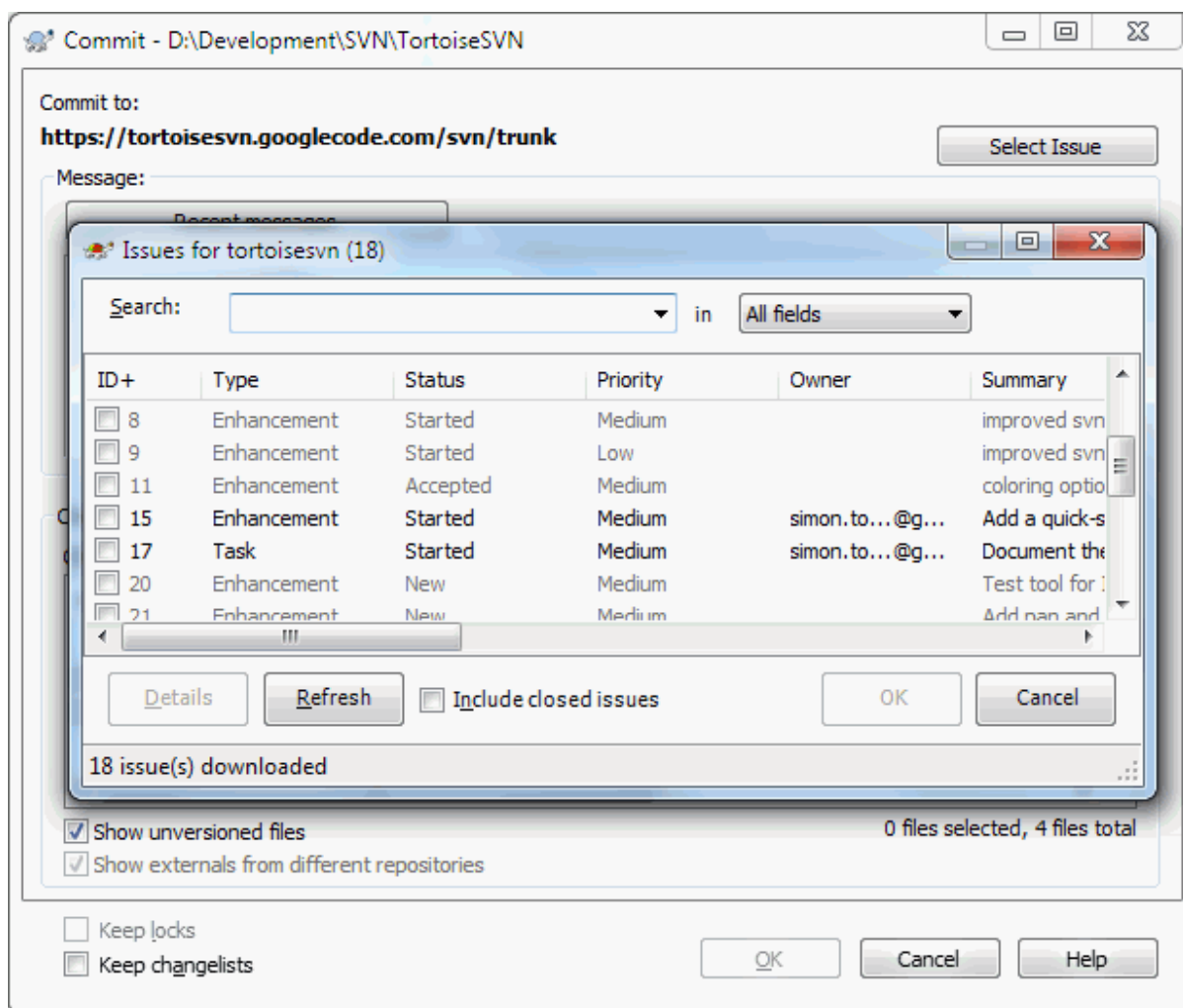
Việc tích hợp bộ theo dõi sự cố không chỉ giới hạn trong TortoiseSVN; nó có thể được dùng với bất kỳ máy khách Subversion nào. Để có thêm thông tin, đọc bản đầy đủ *Thông Số Tích Hợp Bộ Theo Dõi Sự Cố* [<http://tortoisesvn.googlecode.com/svn/trunk/doc/notes/issuetrackers.txt>] trong kho nguồn TortoiseSVN. (**Phần 3**, “**Giấy phép**” giải thích cách truy cập vào kho.)

### 4.28.2. Lấy Thông tin từ Bộ Theo Dõi Vấn Đề

Các phần trước đảm đương việc thêm thông tin vào các thông điệp tường trình. Nhưng nếu những gì cần để nhận được thông tin từ bộ theo dõi vấn đề? Hộp thoại cam kết có một giao diện COM cho phép tích hợp một chương trình bên ngoài mà có thể nói chuyện với bộ theo dõi của bạn. Thông thường bạn có thể muốn truy vấn bộ theo dõi để có được một danh sách các vấn đề mở được giao cho bạn, để bạn có thể chọn những vấn đề đang được đề cập trong cam kết này.

Bất kỳ giao diện nào như vậy là tất nhiên rất cụ thể cho hệ thống theo dõi vấn đề của bạn, vì vậy chúng tôi không thể cung cấp phần này, và mô tả làm thế nào để tạo ra một chương trình như vậy là vượt quá phạm vi của sổ tay này. Định nghĩa giao diện và các mẫu plugin bằng C# và C++/ATL có thể được lấy từ các thư mục đóng góp trong *TortoiseSVN kho* [<http://tortoisesvn.googlecode.com/svn/trunk/contrib/issue-tracker-plugins/>]. (**Phần 3**, “**Giấy phép**” giải thích làm thế nào để truy cập vào kho lưu trữ) Một bản tóm tắt của API cũng được đưa ra trong **Chương 6**, *giao diện IBugtraqProvider*. Một ví dụ plugin (chạy được) bằng C# *Gurtle* [<http://code.google.com/p/gurtle/>] mà thực hiện các giao diện COM cần thiết để tương tác với *Google Code* [<http://code.google.com/hosting/>] bộ theo dõi vấn đề.

Đối với mục đích minh họa, chúng ta hãy giả sử rằng quản trị viên hệ thống của bạn đã cung cấp cho bạn một plugin theo dõi vấn đề mà bạn đã cài đặt, và rằng bạn đã thiết lập một số bản sao làm việc của bạn sử dụng các plugin trong hộp thoại thiết lập TortoiseSVN của. Khi bạn mở hộp thoại cam kết từ một bản sao làm việc mà các plugin đã được chỉ định, bạn sẽ thấy một nút mới ở trên cùng của hộp thoại.



**Hình 4.62. Hộp thoại ví dụ truy vấn theo dõi vấn đề**

Trong ví dụ này, bạn có thể chọn một hoặc nhiều vấn đề mở. Các plugin sau đó có thể tạo ra văn bản được định dạng đặc biệt mà nó bổ sung vào thông điệp tường trình của bạn.

## 4.29. Tích hợp với các Trình Xem Kho Lưu Trữ dựa trên Web

Có một số trình xem kho lưu trữ dựa trên web có sẵn để sử dụng với Subversion *ViewVC* [<http://www.viewvc.org/>] và *WebSVN* [<http://websvn.tigris.org/>]. TortoiseSVN cung cấp phương tiện để liên kết với những chương trình xem n.

Bạn có thể tích hợp một trình xem kho tùy lựa chọn của bạn trong TortoiseSVN. Để làm điều này, bạn phải xác định một số thuộc tính xác định mối liên kết. Chúng phải được thiết lập trên thư mục: ( [Phần 4.17](#), “**Thiết lập Dự án**” )

webviewer:revision

Thiết lập thuộc tính này đến các URL của trình xem kho của bạn để xem tất cả thay đổi trong một phiên bản cụ thể. Nó phải được đúng URI mã hóa và nó phải chứa %REVISION%.%REVISION% được thay thế bằng số sửa đổi trong câu hỏi. Điều này cho phép TortoiseSVN để hiển thị một mục trình đơn ngữ cảnh trong hộp thoại tường trình Trình Đơn Ngữ Cảnh → Xem sửa đổi trong trình xem web .

webviewer:pathrevision

Đặt thuộc tính này để các URL của trình xem kho của bạn để xem các thay đổi vào một tập tin cụ thể trong một phiên bản cụ thể. Nó phải được mã hóa URI đúng và có chứa %REVISION% và %PATH%.%PATH% được thay thế bằng các đường dẫn tương đối đến gốc kho. Điều này cho phép TortoiseSVN hiển thị một mục trình



đơn ngữ cảnh trong hộp thoại tường trình Trình Đơn Ngữ Cảnh → Xem sửa đổi cho đường dẫn trong trình xem web Ví dụ, nếu bạn nhấp chuột phải trong cửa sổ phía dưới hộp thoại tường trình vào một mục tập tin `/trunk/src/file` thì `%PATH%` trong URL sẽ được thay thế bằng `/trunk/src/file`.

Bạn cũng có thể sử dụng các URL tương đối thay vì tuyệt đối. Điều này rất hữu ích trong trường hợp trình xem web của bạn là trên cùng một tên miền / máy chủ như là kho lưu trữ nguồn của bạn. Trong trường hợp tên miền có thay đổi, bạn không phải điều chỉnh thuộc tính `webviewer:revision` và `webviewer:pathrevision`. Định dạng tương tự như đối với các thuộc tính `bugtraq:url`. Xem [Phần 4.28, “Tích hợp với Hệ thống Theo dõi Lỗi / Trình theo dõi Vấn đề”](#).



## Thiết lập các Thuộc Tính trên các Thư Mục

Những thuộc tính này phải được thiết lập trên các thư mục hệ thống để làm việc. Khi bạn cam kết một tập tin hoặc thư mục các thuộc tính được đọc từ các thư mục đó. Nếu các thuộc tính không được tìm thấy ở đó, TortoiseSVN sẽ tìm kiếm trở lên thông qua cây thư mục để tìm thấy chúng cho đến khi nó đến một thư mục chưa phiên bản, hoặc gốc cây (ví dụ: `C:\`) được tìm thấy. Nếu bạn có thể chắc chắn rằng mỗi người dùng chỉ kiểm xuất từ ví dụ như `trunk/` và không phải là một thư mục con, thì nó là đủ nếu bạn thiết lập các thuộc tính trên `trunk/`. Nếu bạn không thể chắc chắn, bạn nên thiết lập các thuộc tính một cách đệ quy trên mỗi thư mục con. Một thuộc tính thiết lập sâu hơn trong hệ thống phân cấp dự án sẽ ghi đè các thiết lập trên các mức độ cao hơn (gần với `trunk/`).

Đối với thuộc tính *chi* của dự án, tức là `tsvn:`, `bugtraq:` và `webviewer:` bạn có thể sử dụng hộp kiểm **Đệ quy** để thiết lập thuộc tính cho tất cả các thư mục con trong hệ thống phân cấp, mà không cần đặt nó trên tất cả các tập tin.

Khi bạn thêm mới các thư mục con vào một bản sao làm việc bằng cách sử dụng TortoiseSVN, bất kỳ thuộc tính dự án có trong thư mục mẹ sẽ tự động được thêm vào thư mục con.



## Các Hạn chế khi Sử dụng Trình duyệt kho

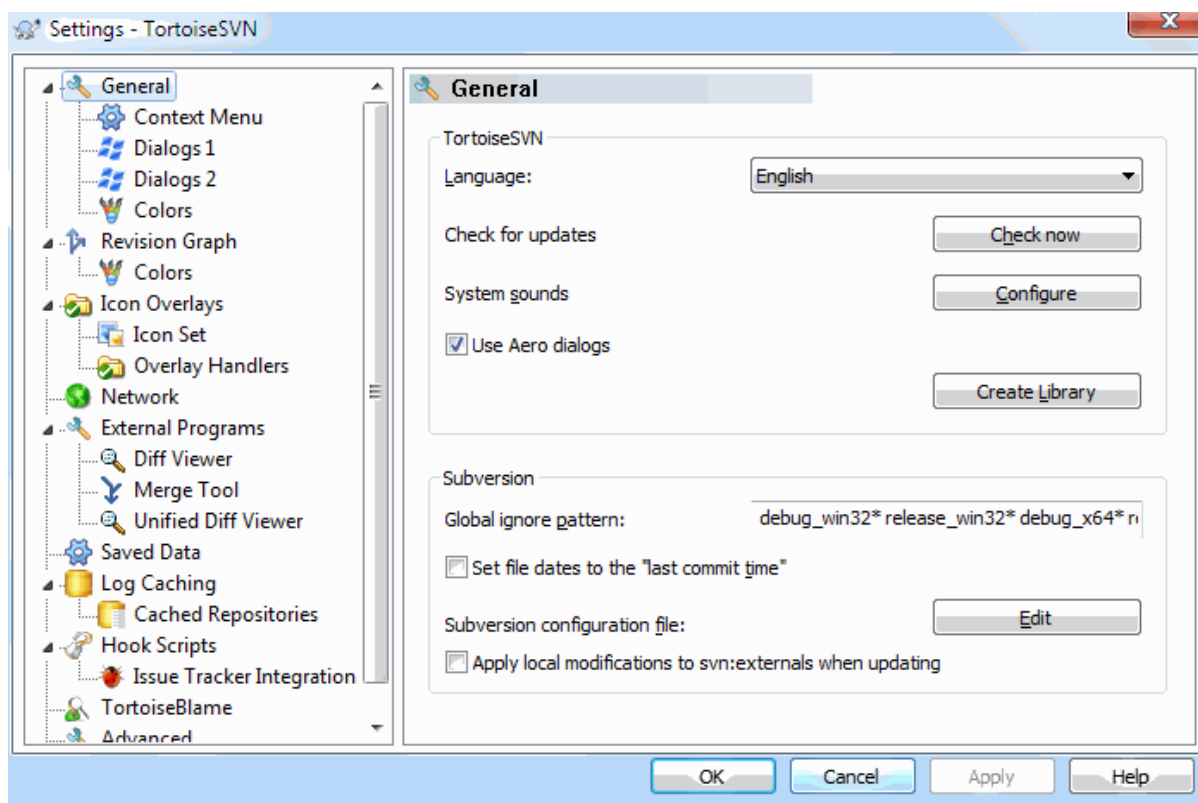
Bởi vì sự tích hợp màn xem kho phụ thuộc vào truy cập vào các thuộc tính Subversion, bạn sẽ chỉ thấy các kết quả khi sử dụng một bản sao làm việc được kiểm xuất. Lấy thuộc tính từ xa là một hoạt động chậm, vì vậy bạn sẽ không thấy tính năng này hoạt động từ trình duyệt kho, trừ khi bạn bắt đầu trình duyệt kho từ bản sao làm việc của bạn. Nếu bạn bắt đầu trình duyệt kho bằng cách nhập vào URL của kho bạn sẽ không thấy tính năng này.

Vì lý do tương tự, các thuộc tính dự án sẽ không được truyền tự động khi một thư mục con được thêm vào bằng cách sử dụng trình duyệt kho.

## 4.30. Các thiết I của TortoiseSVN

Để tìm hiểu các thiết lập khác nhau để làm gì, chỉ việc dùng con trỏ của bạn một giây trên hộp biên tập / hộp kiểm ... và một tooltip hữu ích sẽ bật lên.

### 4.30.1. Các Thiết Lập Tổng Qu



**Hình 4.63. Hộp Thoại Thiết Lập, Trang Tổng Quát**

Hộp thoại này cho phép bạn chỉ định ngôn ngữ ưa thích của bạn, và các thiết lập cụ thể cho Subversion.

#### Ngôn ngữ

Chọn ngôn ngữ giao diện người dùng của bạn. Bạn còn mong đợi cái gì khác?

#### Kiểm tra xem có các bản cập nhật

TortoiseSVN sẽ liên lạc với trang web tải về của nó theo định kỳ để xem nếu có một phiên bản mới hơn của chương trình. Nếu có, nó sẽ hiển thị một liên kết thông báo trong hộp thoại cam kết. Sử dụng **Kiểm tra ngay** nếu bạn muốn một câu trả lời ngay lập tức. Phiên bản mới sẽ không được tải về; bạn chỉ đơn giản nhận được một hộp thoại thông tin cho biết rằng phiên bản mới có sẵn.

#### Âm thanh hệ thống

TortoiseSVN có ba tùy chỉnh âm thanh đã được cài đặt theo mặc định.

- Lỗi
- Lưu
- Cảnh báo

Bạn có thể chọn các âm thanh khác nhau (hoặc tắt những âm thanh này hoàn toàn) bằng cách sử dụng Windows Control Panel. **Cấu hình** là một phím tắt vào Control Panel.

#### Sử dụng Hộp thoại Aero

Trên Windows Vista và các hệ thống mới hơn việc này điều khiển liệu các hộp thoại có sử dụng phong cách Aero.

#### Tạo Thư Viện

Trên Windows 7, bạn có thể tạo ra một thư viện, trong đó nhóm các bản sao làm việc nằm rải rác ở những nơi khác nhau trên hệ thống của bạn.

#### Khuôn mẫu bỏ qua toàn cục

Các khuôn mẫu bỏ qua toàn cục được sử dụng để ngăn chặn các tập tin không phiên bản được hiển thị ví dụ như trong hộp thoại cam kết. Tập tin phù hợp với các khuôn mẫu được bỏ qua bởi nhập khẩu. Bỏ qua các tập tin hoặc thư mục bằng cách gõ tên hoặc phần mở rộng. Mẫu được phân cách bằng dấu cách ví dụ như `bin obj *.bak *.*?? *.jar *.[Tt]mp`. Những khuôn mẫu không nên bao gồm bất kỳ dải phân cách đường dẫn. Cũng lưu ý rằng không có cách nào để phân biệt giữa các tập tin và thư mục. Đọc [Phần 4.13.1, “Việc Khớp các Mẫu trong Danh sách Bỏ qua”](#) để có thêm thông tin về cú pháp khớp khuôn m.

Lưu ý rằng các khuôn mẫu bỏ qua mà bạn chỉ định ở đây cũng sẽ ảnh hưởng đến các máy khách Subversion khác đang chạy trên máy tính của bạn, bao gồm cả các máy khách dùng dòng lệnh đầu nh.



## Cẩn thận

Nếu bạn sử dụng tập tin cấu hình Subversion để thiết lập một khuôn mẫu bỏ qua toàn cục, nó sẽ ghi đè lên các thiết lập bạn thực hiện ở đây. Tập tin cấu hình Subversion được truy cập bằng cách sử dụng [Biên tập](#) như mô tả dưới đây.

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 [Phần 4.13, “Bỏ qua các Tập Tin và Thư mục”](#) for more information.

Thiết lập ngày tháng tập tin đến “thời gian cam kết cuối cùng”

Tùy chọn sẽ bảo TortoiseSVN đặt ngày tập tin về thời gian cam kết cuối cùng khi làm một kiểm xuất hoặc cập nhật. Nếu không TortoiseSVN sẽ sử dụng ngày hiện tại. Nếu bạn đang phát triển phần mềm nói chung là tốt nhất để sử dụng ngày tháng hiện tại bởi vì các hệ thống xây dựng bình thường nhìn vào những con dấu ngày để quyết định những tập tin nào cần biên dịch. Nếu bạn sử dụng “thời gian cam kết cuối cùng” và hoàn nguyên với một bản chỉnh sửa tập tin cũ hơn, dự án của bạn không thể biên dịch như bạn mong đợi.

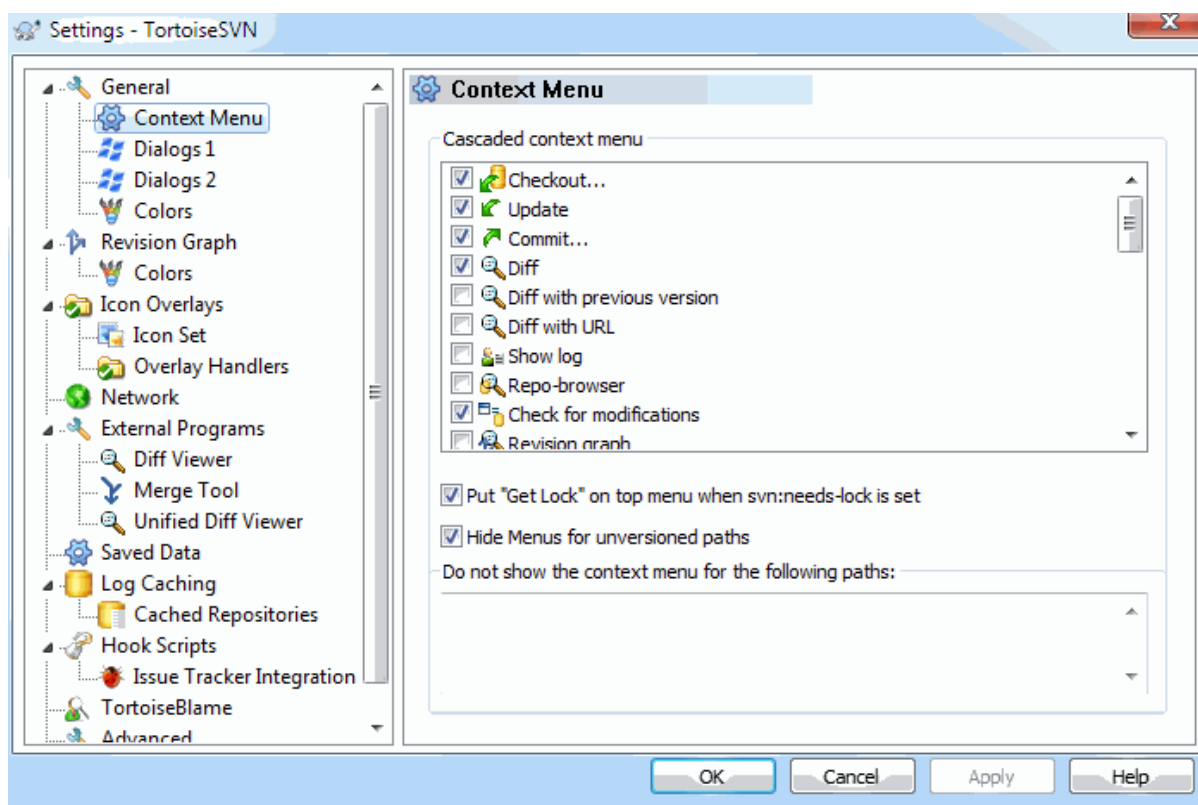
tập tin cấu hình Subversion

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.

Áp dụng các thay đổi địa phương đến `svn:externals` khi cập nhật

Tùy chọn này bảo TortoiseSVN luôn luôn áp dụng thay đổi địa phương đến thuộc tính `svn:externals` khi cập nhật các bản sao làm việc.

### 4.30.1.1. Các Thiết Lập Trình Đơn Ngữ Cảnh



**Hình 4.64. Hộp Thoại Thiết Lập, Trang Trình Đơn Ngữ Cảnh**

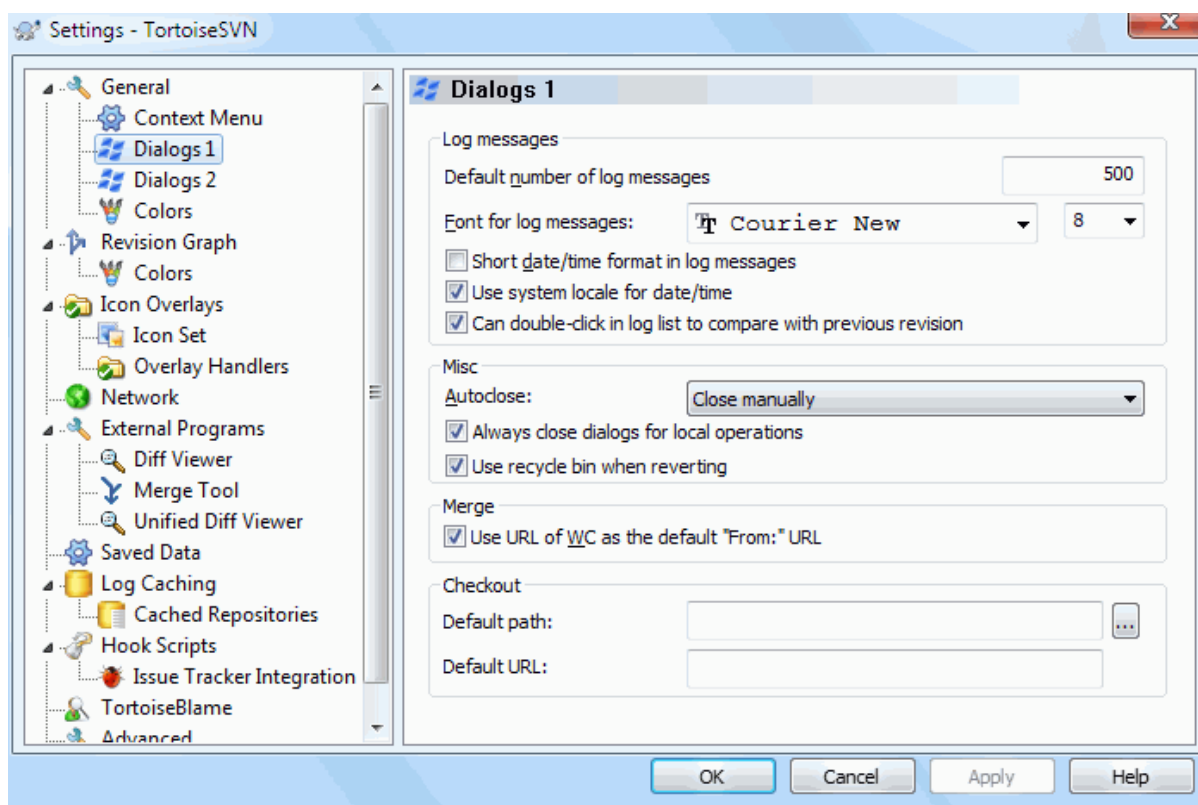
Trang này cho phép bạn xác định các mục menu trong trình đơn ngữ cảnh của TortoiseSVN sẽ hiển thị trong trình đơn ngữ cảnh chính, và sẽ xuất hiện trong trình đơn phụ TortoiseSVN. Theo mặc định, hầu hết các mục không được kiểm soát và xuất hiện trong trình đơn con.

Có một trường hợp đặc biệt cho **Nhận Khóa**. Tất nhiên bạn có thể thúc đẩy nó đến cấp cao nhất bằng cách sử dụng danh sách kê trên, nhưng hầu hết các tập tin không cần khóa việc này chỉ thêm lộn xộn. Tuy nhiên, một tập tin với thuộc tính `svn:needs-lock` cần hành động này mỗi khi nó được biên tập, vì vậy trong trường hợp đó là rất hữu ích để có ở cấp cao nhất. Đánh dấu chọn hộp ở đây có nghĩa là khi một tập tin được chọn có các thuộc tính `svn:needs-lock` được thiết lập, **Nhận Khóa** sẽ luôn luôn xuất hiện ở cấp cao nhất.

Hầu hết thời gian, bạn sẽ không cần trình đơn ngữ cảnh TortoiseSVN, ngoài các thư mục đang được kiểm soát phiên bản bằng Subversion. Đối với các thư mục không phải là phiên bản, bạn chỉ thực sự cần trình đơn ngữ cảnh khi bạn muốn làm một kiểm xuất. Nếu bạn kiểm tra tùy chọn **Giấu trình đơn** cho các đường dẫn không phiên bản TortoiseSVN sẽ không thêm các lựa chọn trình đơn ngữ cảnh cho các thư mục không phiên bản. Tuy nhiên, các lựa chọn được thêm vào cho tất cả các mục và đường dẫn trong một thư mục có phiên bản. Và bạn có thể nhận được các mục cho các thư mục không phiên bản bằng cách giữ phím **Shift** trong khi hiển thị trình đ ngữ cảnh.

Nếu có một số đường dẫn trên máy tính của bạn mà bạn chỉ không muốn trình đơn ngữ cảnh của TortoiseSVN xuất hiện chút nào, bạn có thể liệt kê chúng trong hộp ở phía dưới.

### 4.30.1.2. Các Thiết Lập Hộp Thoại TortoiseSVN 1



**Hình 4.65. Trang Hộp Thoại Thiết Lập, Các Hộp Thoại 1**

Hộp thoại này cho phép bạn cấu hình một số hộp thoại TortoiseSVN theo cách bạn thích.

Số mặc định của các thông điệp tường trình

Giới hạn số thông điệp tường trình TortoiseSVN nạp khi bạn lần đầu tiên chọn TortoiseSVN → Hiện Tường Trình Hữu ích cho các kết nối máy chủ chậm. Bạn luôn có thể sử dụng Hiện Tất Cả hoặc 100 Tiếp theo để nhận được nhiều thông điệp hơn.

Phông chữ cho thông điệp tường trình

Lựa chọn các mặt chữ và kích thước được sử dụng để hiển thị các thông điệp tường trình trong phần bảng giữa của hộp thoại Tường Trình Sửa Đổi, và khi soạn thông điệp tường trình vào hộp thoại Cam Kết.

Định dạng ngày / thời gian ngắn trong các thông điệp tường trình

Nếu các thông điệp dài tiêu chuẩn sử dụng quá nhiều không gian trên màn hình của bạn hãy sử dụng định dạng ngắn.

Có thể nhấn kép chuột trong danh sách tường trình để so sánh với bản sửa đổi trước

Nếu bạn thường xuyên thấy mình so sánh các bản sửa đổi trong cửa sổ trên cùng của hộp thoại tường trình, bạn có thể sử dụng tùy chọn này để cho phép hành động đó khi nhấp đúp chuột. Nó không được kích hoạt mặc định vì việc nạp các khác thường là một quá trình lâu dài, và nhiều người muốn tránh sự chờ đợi sau khi tình cờ nhấp đúp chuột vào, đó là lý do tại sao tùy chọn này không được kích hoạt mặc định.

Tự động-đóng

TortoiseSVN có thể tự động đóng tất cả các hộp thoại tiến độ khi hành động được hoàn thành mà không có lỗi. Thiết lập này cho phép bạn chọn các điều kiện để đóng các hộp thoại. Thiết lập mặc định (được đề nghị) Đóng bằng tay cho phép bạn xem lại tất cả các thông điệp và kiểm tra những gì đã xảy ra. Tuy nhiên, bạn có thể quyết định rằng bạn muốn bỏ qua một số loại thông điệp và hộp thoại tự động đóng lại nếu không có thay đổi quan trọng.

Tự động đóng nếu không có sáp nhập, thêm, xóa có nghĩa là hợp thoại tiền độ sẽ đóng nếu có những cập nhật đơn giản, nhưng nếu các thay đổi từ kho lưu trữ đã được sáp nhập với những thay đổi của bạn, hoặc nếu có bất kỳ tập tin đã được thêm vào hoặc bị xóa, hợp thoại sẽ vẫn mở. Nó cũng sẽ mở nếu có bất kỳ xung đột hoặc lỗi trong quá trình hoạt động.

Tự động đóng nếu không có xung đột làm giảm nhẹ các tiêu chuẩn hơn nữa và sẽ đóng hợp thoại ngay cả khi có kết hợp, bổ sung thêm hoặc xóa. Tuy nhiên, nếu có bất kỳ mâu thuẫn hoặc sai sót, hợp thoại vẫn mở.

Tự động đóng nếu không có lỗi luôn luôn đóng hợp thoại ngay cả khi có xung đột. Điều kiện duy nhất mà giữ hợp thoại mở là một điều kiện lỗi, xảy ra khi Subversion là không thể hoàn thành nhiệm vụ. Ví dụ, cập nhật thất bại bởi vì máy chủ là không thể tiếp cận được, hoặc cam kết bị thất bại bởi vì bản sao làm việc quá cũ.

#### Luôn đóng hợp thoại cho các tác vụ cục bộ

Các tác vụ cục bộ như thêm các tập tin hoặc hoàn nguyên lại thay đổi không cần liên hệ với các kho lưu trữ và hoàn thành một cách nhanh chóng, do đó, hợp thoại tiến triển thường ít được quan tâm đến. Chọn tùy chọn này nếu bạn muốn hợp thoại tiến triển tự động đóng lại sau các hoạt động này, trừ khi có lỗi.

#### Sử dụng thùng rác khi quay hoàn nguyên

Khi bạn hoàn nguyên các thay đổi cục bộ, những thay đổi của bạn sẽ bị loại bỏ. TortoiseSVN cung cấp cho bạn một mạng lưới an toàn hơn bằng cách gửi các tập tin được sửa đổi vào thùng rác trước khi đưa trở lại bản sao nguyên sơ. Nếu bạn muốn bỏ qua thùng rác, bỏ chọn tùy chọn này.

#### Sử dụng URL của WC như mặc định URL “ Từ: ”

Trong hợp thoại hợp nhất, hành vi mặc định là cho URL Từ: được ghi nhớ giữa các sáp nhập. Tuy nhiên, một số người muốn thực hiện kết hợp từ nhiều điểm khác nhau trong hệ thống phân cấp của chúng, và thấy dễ dàng hơn để bắt đầu với URL của bản sao làm việc hiện tại. Điều này sau đó có thể được chỉnh sửa để chỉ đến một đường dẫn song song trên một chi nhánh khác.

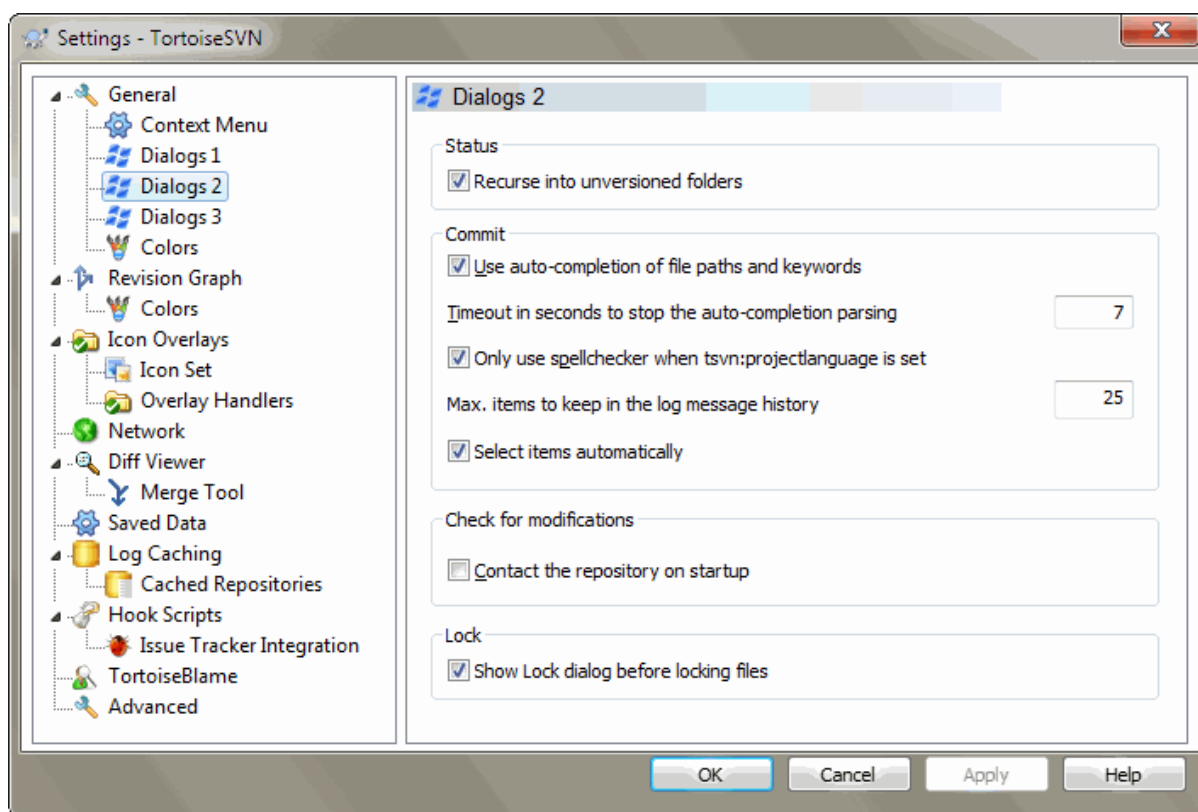
#### Đường dẫn kiểm xuất mặc định

Bạn có thể chỉ định đường dẫn mặc định cho các kiểm xuất. Nếu bạn giữ tất cả các kiểm xuất của bạn ở một nơi, nó rất hữu ích để có ổ đĩa và thư mục được điền sẵn, do đó bạn chỉ cần thêm tên thư mục mới vào phía cuối.

#### URL kiểm xuất mặc định

Bạn cũng có thể chỉ định các URL mặc định cho các kiểm xuất. Nếu bạn thường xuyên kiểm xuất các dự án con của một số dự án rất lớn, nó có thể hữu ích để có địa chỉ URL được điền sẵn, do đó bạn chỉ cần thêm tên dự án con vào phía cuối.

### 4.30.1.3. Hộp Thoại Thiết Lập TortoiseSVN 2



**Hình 4.66. Trang Hộp Thoại Thiết Lập, Hộp Thoại 2**

Đệ quy vào thư mục các chưa phiên bản

Nếu hộp này được đánh dấu (trạng thái mặc định), sau đó bất cứ khi nào tình trạng của một thư mục chưa phiên bản được hiển thị trong hộp thoại **Thêm**, **Cam kết** hoặc **Kiểm tra cho sửa đổi**, tất cả các tập tin và thư mục con cũng được hiển thị. Nếu bạn bỏ chọn ô này, chỉ có cha mẹ chưa phiên bản được hiển thị. Bỏ chọn làm giảm sự lộn xộn trong các hộp thoại. Trong trường hợp đó nếu bạn chọn một thư mục chưa phiên bản cho **Thêm**, nó được thêm một cách đệ quy.

Trong hộp thoại **Kiểm Tra cho Sửa Đổi**, bạn có thể lựa chọn để xem các mục bị bỏ qua. Nếu hộp này được kiểm tra thì bất cứ khi nào một thư mục được bỏ qua được tìm thấy, tất cả các mục con cũng sẽ được hiển thị.

Sử dụng tự động hoàn thành các đường dẫn tập tin và các từ khóa

Hộp thoại cam kết bao gồm một cơ sở để phân tích cú pháp danh sách các tên tập tin được cam kết. Khi bạn gõ 3 chữ cái đầu tiên của một mục trong danh sách, hộp hoàn thành tự động bật lên, và bạn có thể nhấn Enter để hoàn thành tên tập tin. Đánh dấu hộp kiểm để kích hoạt tính năng này.

Thời gian chờ tính bằng giây để ngăn chặn sự phân tích cú pháp tự động hoàn thành

Phân tích cú pháp tự động hoàn thành có thể khá chậm nếu có nhiều các tập tin lớn để kiểm tra. Thời gian chờ này nhằm ngăn hộp thoại cam kết bị giữ quá lâu. Nếu bạn đang thiếu các thông tin quan trọng về tự động hoàn thành, bạn có thể kéo dài thời gian chờ.

Chỉ sử dụng kiểm tra chính tả khi `tsvn:projectlanguage` được thiết lập

Nếu bạn không muốn sử dụng kiểm tra chính tả cho tất cả các cam kết, đánh dấu hộp kiểm này. Kiểm tra chính tả vẫn sẽ được kích hoạt nơi các thuộc tính của dự án yêu cầu nó.

Số lượng lớn nhất các mục để giữ trong lịch sử thông điệp tường trình

Khi bạn gõ vào một thông điệp tường trình trong hộp thoại cam kết, TortoiseSVN lưu nó cho việc tái sử dụng có thể sau này. Theo mặc định, nó sẽ giữ 25 thông điệp tường trình cuối cùng cho mỗi kho, nhưng bạn có thể tùy chỉnh con số ở đây. Nếu bạn có nhiều kho khác nhau, bạn có thể để giảm bớt để tránh làm đầy registry của bạn.

Lưu ý rằng thiết lập này chỉ áp dụng cho các thông điệp mà bạn gõ trên máy tính này. Nó không có gì để làm với bộ nhớ cache tường trình.

Mở lại hộp thoại cam kết và chi nhánh / thẻ sau khi một cam kết thất bại

Khi một cam kết thất bại vì một số lý do (bản sao làm việc cần phải cập nhật, hook tiền-cam kết bác bỏ cam kết, lỗi mạng, vv), bạn có thể chọn tùy chọn này để giữ cho hộp thoại cam kết mở sẵn sàng để thử lại.

Chọn các mục một cách tự động

Các hành vi bình thường trong hộp thoại cam kết cho tất cả các mục được sửa đổi (có phiên bản) được lựa chọn để cam kết tự động. Nếu bạn muốn bắt đầu với không có gì được lựa chọn và chọn các mục cho cam kết bằng tay, bỏ chọn ô này.

Liên hệ với các kho lưu trữ khi khởi động

Hộp thoại Kiểm Tra Sửa Đổi kiểm tra các bản sao làm việc theo mặc định, và chỉ liên lạc kho lưu trữ khi bạn nhấp vào **Kiểm tra kho lưu trữ**. Nếu bạn luôn luôn muốn kiểm tra các kho lưu trữ, bạn có thể sử dụng thiết lập này để làm cho hành động đó xảy ra tự động.

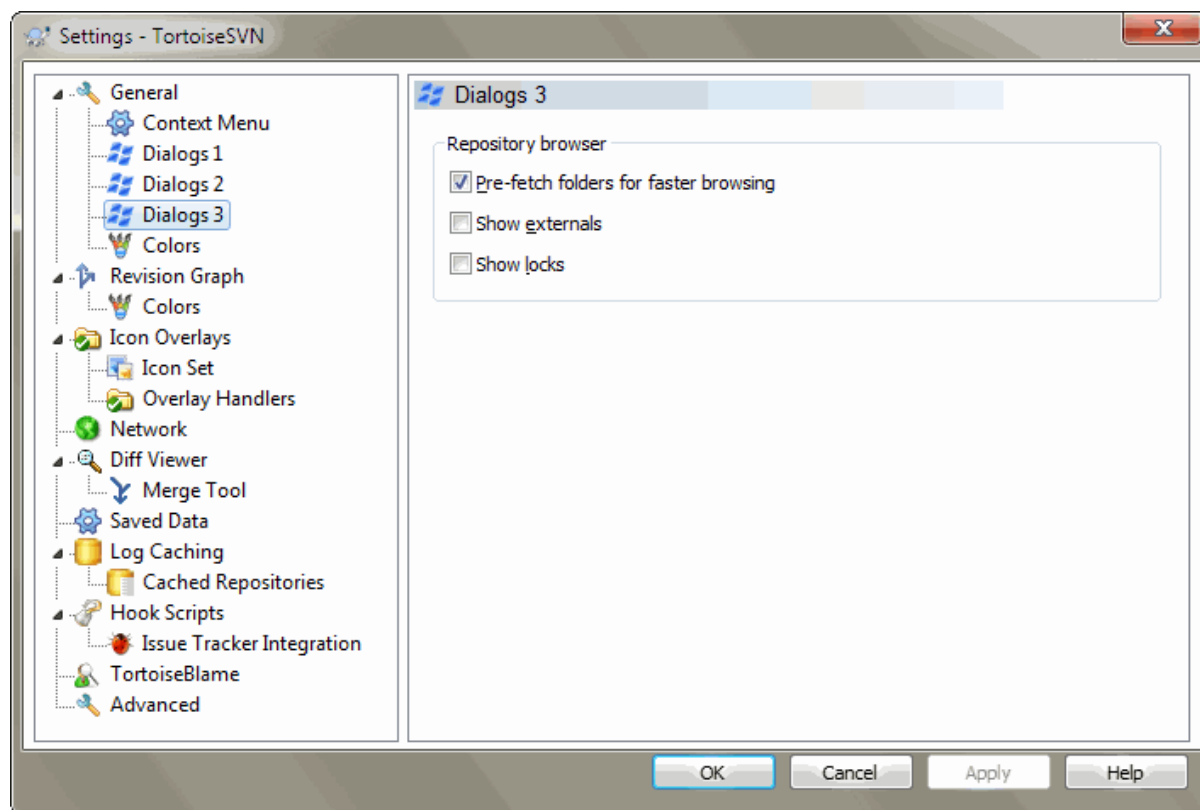
Hiện thị hộp thoại Khóa trước khi khóa các tập tin

Khi bạn chọn một hay nhiều tập tin và sau đó sử dụng **TortoiseSVN** → **Khóa** để lấy khóa ra khỏi các tập tin, trên một số dự án, nó là phong tục để viết một thông điệp khóa giải thích lý do tại sao bạn đã bị khóa các tập tin. Nếu bạn không sử dụng thông điệp khóa, bạn có thể bỏ chọn hộp này để bỏ qua hộp thoại và khóa các tập tin ngay lập tức.

Nếu bạn sử dụng lệnh khóa trên một thư mục, bạn luôn luôn được trình bày với hộp thoại khóa bởi nó cũng cung cấp cho bạn tùy chọn để chọn các tập tin để khóa.

Nếu dự án của bạn đang sử dụng thuộc tính `tsvn:lockmsgminsize`, bạn sẽ thấy hộp thoại khóa bất kể thiết lập này vì dự án *yêu cầu* thông điệp khóa.

#### 4.30.1.4. Thiết Lập Hộp Thoại TortoiseSVN 3



Hình 4.67. Hộp Thoại Thiết Lập, Trang Hộp Thoại 3



Nạp trước các thư mục để duyệt nhanh hơn

Nếu hộp này được đánh dấu (trạng thái mặc định), thì trình duyệt kho sẽ ngầm nạp thông tin về các thư mục được hiển thị. Theo cách đó khi bạn duyệt tới một trong các thư mục, thông tin sẽ được sẵn sàng

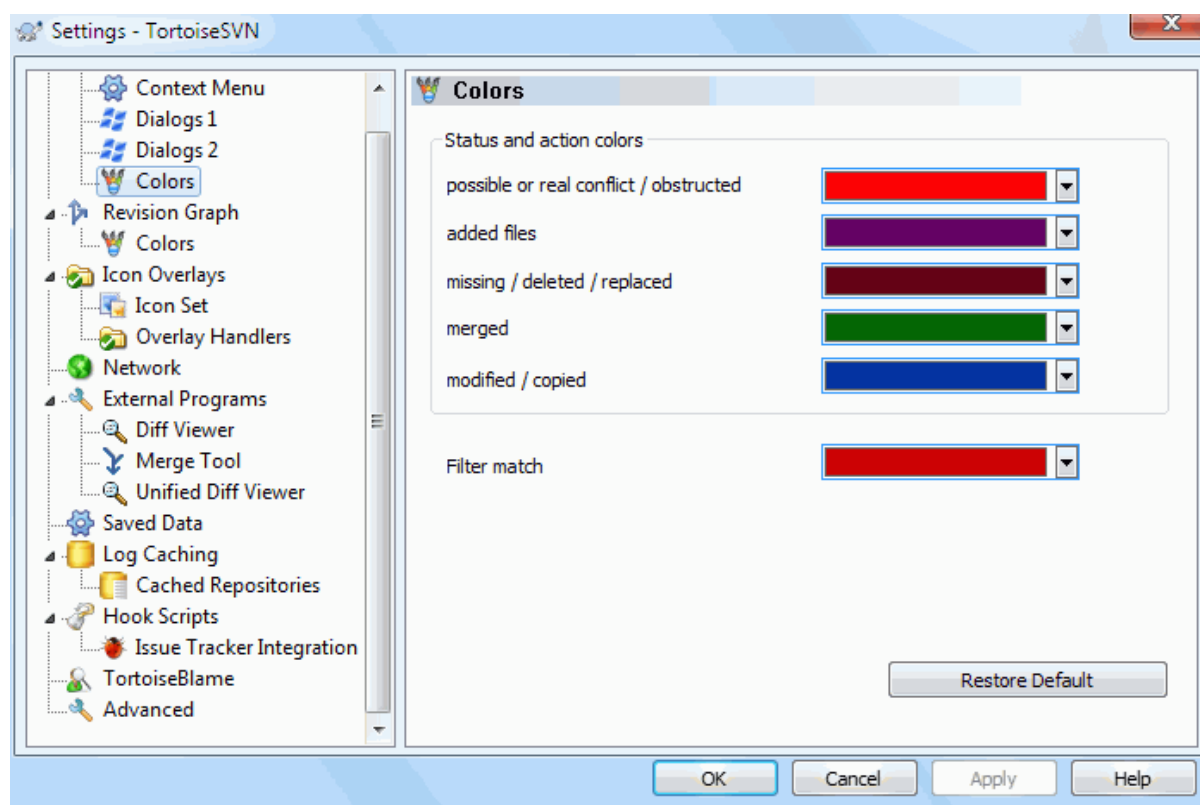
Tuy nhiên một số máy chủ không thể xử lý nhiều truy xuất mà việc này gây ra hoặc khi không được cấu hình đúng sẽ xem những truy xuất bị hỏng hoặc sẽ chặn chúng. Trong trường hợp này bạn có thể vô hiệu hóa việc nạp trước ở đây.

Hiển thị các thành phần ngoài

Nếu hộp này được đánh dấu (trạng thái mặc định), thì trình duyệt kho sẽ hiển thị các tập tin và thư mục được bao gồm với thuộc tính `svn:externals` như những tập tin và thư mục bình thường, nhưng với biểu tượng che phủ để đánh dấu chúng là từ nguồn bên ngoài.

Với tính năng nạp-trước được giải thích phía trên, việc này cũng có thể tạo nhiều áp lực lên các máy chủ. Trong trường hợp này bạn có thể vô hiệu hóa tính năng này ở đây.

#### 4.30.1.5. Thiết Lập Màu TortoiseSVN



**Hình 4.68. Trang Hộp Thoại Thiết Lập, Màu**

Hộp thoại này cho phép bạn cấu hình các màu sắc văn bản được sử dụng trong hộp thoại TortoiseSVN theo cách của bạn.

Sự xung đột có thể có hoặc có thực / tác nghẽn

Một cuộc xung đột đã xảy ra trong quá trình cập nhật, hoặc có thể xảy ra trong quá trình hợp nhất. Cập nhật bị che khuất bởi một tập tin/thư mục chưa phiên bản đang tồn tại cùng tên như là được phiên bản.

Màu này cũng được sử dụng cho các thông báo lỗi trong hộp thoại tiến triển.

Các tập tin được thêm

Các mục được bổ sung vào kho lưu trữ.

#### Thiếu / bị xóa / bị thay thế

Các mục đã được xóa từ kho, bị mất từ bản sao làm việc, hoặc bị xóa từ bản sao làm việc và thay thế bằng một tập tin cùng tên.

#### Được hợp nhất

Những thay đổi từ kho được sáp nhập thành công vào WC mà không tạo ra bất kỳ xung đột.

#### Được sửa đổi / sao chép

Thêm vào với lịch sử, hoặc các đường dẫn được sao chép trong kho. Cũng được sử dụng trong hộp thoại tường trình cho các mục trong đó bao gồm các mục được sao chép.

#### Nút đã bị xóa

Một mục đã bị xóa khỏi kho.

#### Nút được thêm vào

Một mục đã được bổ sung vào kho lưu trữ, bởi một tác vụ thêm, sao chép hoặc di chuyển.

#### Nút được đổi tên

Một mục đã được đổi tên trong kho.

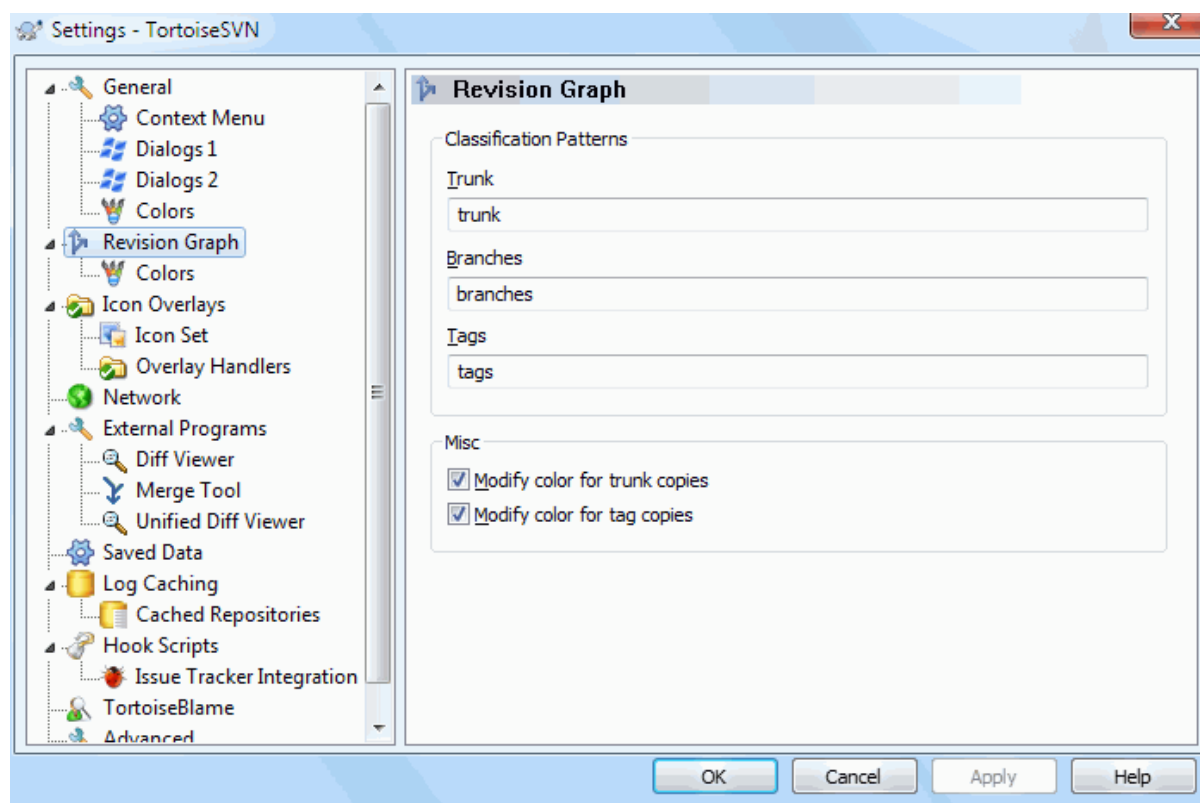
#### Nút được thay thế

Mục ban đầu đã bị xóa và một mục mới có cùng tên thay thế nó.

#### Bộ lọc phù hợp

Khi sử dụng bộ lọc trong hộp thoại tường trình, các từ tìm kiếm được làm nổi bật trong các kết quả bằng cách sử dụng màu sắc này.

### 4.30.2. Các Thiết Lập Cho Đồ thị Sửa Đổi



**Hình 4.69. Hộp Thoại Thiết Lập, Trang Biểu Đồ Sửa Đổi**

#### Các Mẫu Phân Loại

Đồ thị sửa đổi cố gắng để hiển thị một hình ảnh rõ ràng hơn về cấu trúc kho của bạn bằng cách phân biệt giữa thân cây, các nhánh và các thẻ. Vì không có phân loại được xây dựng sẵn vào Subversion, thông tin này được

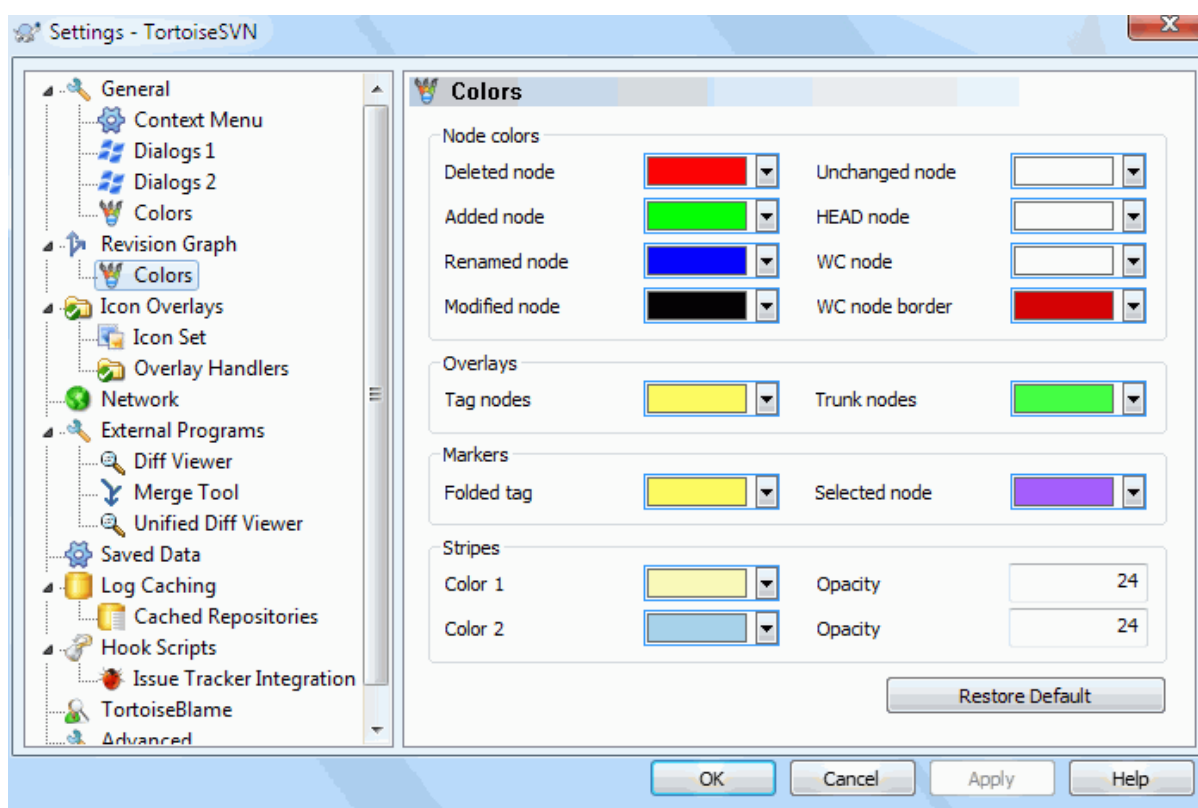
chiết xuất từ các tên đường dẫn. Các thiết lập giả định rằng bạn sử dụng tên tiếng Anh thông thường như đề xuất trong tài liệu Subversion, nhưng tất nhiên việc sử dụng của bạn có thể thay đổi.

Xác định các mẫu được sử dụng để nhận ra những đường dẫn trong ba hộp được cung cấp. Các mẫu sẽ được khớp có phân biệt chữ hoa, nhưng bạn phải chỉ định chung trong chữ thường. Ký tự đặc biệt \* và ? sẽ làm việc như thường lệ, và bạn có thể sử dụng ; để tách biệt nhiều mẫu. Không bao gồm bất kỳ khoảng trắng vì nó sẽ được bao gồm trong thông số khớp.

#### Sửa đổi Màu sắc

Màu sắc được sử dụng trong đồ thị sửa đổi để chỉ ra các loại nút, tức là có một nút được thêm vào, xóa, đổi tên. Để chọn ra phân loại nút, bạn có thể cho phép biểu đồ sửa đổi pha trộn màu sắc để đưa ra một dấu hiệu cho thấy cả hai loại nút và phân loại. Nếu hộp được kiểm tra, pha trộn được sử dụng. Nếu hộp được đánh dấu chọn, màu sắc được sử dụng để chỉ loại nút. Sử dụng hộp thoại lựa chọn màu sắc để phân bổ màu sắc cụ thể được sử dụng.

### 4.30.2.1. Các Màu Sắc Của Đồ Thị Sửa Đổi



**Hình 4.70. Hộp Thoại Thiết Lập, Trang Màu Sắc Đồ Thị Sửa Đổi**

Trang này cho phép bạn cấu hình các màu sắc được sử dụng. Lưu ý rằng các màu sắc quy định ở đây là màu trộn. Hầu hết các nút được tô màu bằng cách sử dụng một sự pha trộn của màu sắc kiểu nút, màu nền và tùy chọn màu sắc phân loại.

#### Nút Đã Bị Xóa

Các mục đã bị xóa và không được sao chép bất cứ nơi nào khác trong cùng một bản sửa đổi.

#### Nút Được Thêm Vào

Mục mới được bổ sung, hoặc sao chép (thêm với lịch sử).

#### Nút Được Đổi Tên

Các mục đã xóa từ một địa điểm và thêm vào trong địa điểm khác của cùng bản sửa đổi.

#### Nút Được Sửa Đổi

Sửa đổi đơn giản mà không cần bất kỳ bổ sung thêm hoặc xóa.

#### Nút Không Thay Đổi

Có thể được sử dụng để hiển thị các sửa đổi được sử dụng như là nguồn của một bản sao, ngay cả khi không có thay đổi (mục được vẽ đồ thị) diễn ra trong bản chỉnh sửa đó.

#### Nút CHÍNH

Bản sửa đổi CHÍNH hiện tại trong kho.

#### Nút WC

Nếu bạn chọn để hiển thị một nút dự cho bản sao làm việc sửa đổi của bạn, gắn vào sửa đổi cam kết cuối cùng của nó trên đồ thị, sử dụng màu sắc này.

#### Đường Biên Nút WC

Nếu bạn chọn để xem bản sao làm việc có được sửa đổi không, sử dụng đường biên với màu sắc cho nút WC khi các sửa đổi được tìm thấy.

#### Đánh Thẻ Các Nút

Các nút được phân loại như các thẻ có thể được pha trộn với màu sắc này.

#### Các Nút Thân Cây

Các nút phân loại là thân cây có thể được pha trộn với màu sắc này.

#### Đánh Dấu Thẻ Được Gấp

Nếu bạn sử dụng gấp thẻ để tiết kiệm không gian, thẻ đánh được dấu trên nguồn sao chép bằng cách sử dụng một khối bằng màu sắc này.

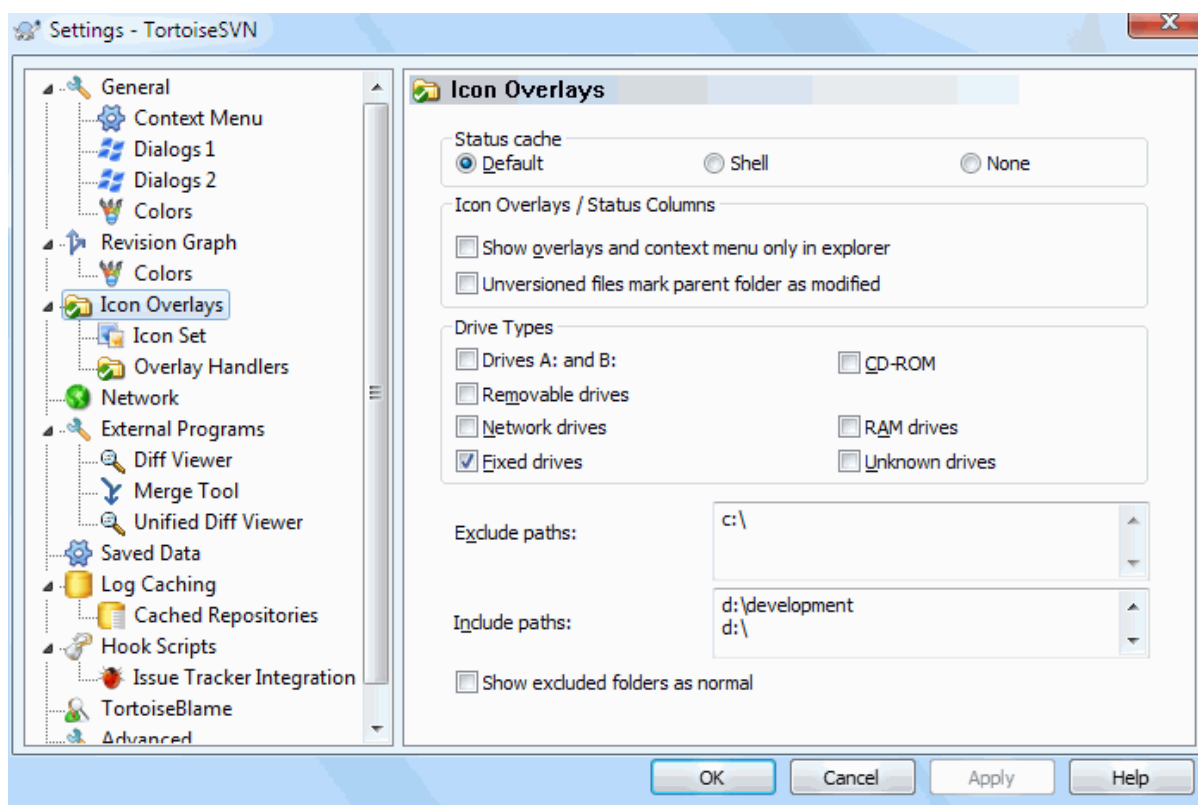
#### Đánh Dấu Cho Nút Được Chọn

Khi bạn nhấp chuột trái vào một nút để chọn nó, đánh dấu được sử dụng để chỉ ra lựa chọn là một khối trong màu sắc này.

#### Các vết sọc

Những màu sắc được sử dụng khi biểu đồ được chia thành các cây con và nền là màu sọc xen kẽ để giúp chọn ra những cây riêng biệt.

### 4.30.3. Các Thiết Lập Cho Lớp Phủ Biểu Tượng



**Hình 4.71. Hộp Thoại Thiết Lập, Trang Lớp Phủ Biểu Tượng**

Trang này cho phép bạn chọn các mục mà TortoiseSVN sẽ hiển thị các lớp phủ biểu tượng.

Bởi vì phải mất nhiều thời gian để lấy trạng thái của một bản sao làm việc, TortoiseSVN sử dụng một bộ nhớ cache để lưu trữ các trạng thái để explorer không bị ngốn quá nhiều khi hiển thị các lớp phủ. Bạn có thể chọn loại bộ nhớ cache mà TortoiseSVN nên sử dụng tùy theo hệ thống của bạn và kích thước bản sao làm việc ở đây:

#### Mặc định

Lưu trữ tất cả các thông tin trạng thái trong một tiến trình riêng biệt ( `TSVNCache.exe` ). Quá trình đó theo dõi tất cả các ổ đĩa cho thay đổi và lấy trạng thái một lần nữa nếu các tập tin bên trong một bản sao làm việc được sửa đổi. Tiến trình này chạy với ưu tiên ít nhất có thể để các chương trình khác không bị ngốn hết vì nó. Điều đó cũng có nghĩa là thông tin trạng thái không phải là *thời gian thực* nhưng nó có thể mất một vài giây cho các lớp phủ để thay đổi.

Ưu điểm: các lớp phủ hiển thị trạng thái đệ quy, nghĩa là nếu một tập tin sâu bên trong một bản sao làm việc được sửa đổi, tất cả các thư mục trở lên đến gốc của bản sao làm việc cũng sẽ hiển thị các lớp phủ sửa đổi. Và bởi vì quá trình này có thể gửi thông báo cho shell, lớp phủ trên màn hình cây trái thường cũng thay đổi.

Nhược điểm: quá trình này chạy liên tục, thậm chí nếu bạn không làm việc trên các dự án của bạn. Nó cũng sử dụng khoảng 10-50 MB RAM tùy thuộc vào số lượng và kích thước của bản sao của bạn làm việc.

#### Shell

Bộ nhớ đệm được thực hiện trực tiếp bên trong các dll mở rộng của shell, nhưng chỉ dành cho các thư mục đang nhìn thấy. Mỗi lần bạn định hướng đến thư mục khác, các thông tin trạng thái được nạp một lần nữa.

Ưu điểm: nhu cầu bộ nhớ chỉ rất ít (khoảng 1 MB bộ nhớ RAM) và có thể hiển thị trạng thái trong *thời gian thực* .

Nhược điểm: Vì chỉ có một thư mục lưu trữ, các lớp phủ không hiển thị trạng thái một cách đệ quy. Đối với các bản sao làm việc lớn, nó có thể mất nhiều thời gian hơn để hiển thị một thư mục trong explorer hơn so với bộ nhớ cache mặc định. Ngoài ra các cột mime-type không có sẵn.

## Không

Với thiết lập này, các TortoiseSVN không lấy trạng thái chút nào trong Explorer. Do đó, các tập tin không nhận được một lớp phủ và các thư mục chỉ có được một lớp phủ "bình thường" nếu chúng có phiên bản. Không có lớp phủ khác được hiển thị, và cũng không có cột dư nào có sẵn.

Ưu điểm: hoàn toàn không sử dụng bộ nhớ bổ sung và không làm chậm Explorer chút nào trong khi đang duyệt.

Bất lợi: Tình trạng thông tin của các tập tin và thư mục không được hiển thị trong Explorer. Để xem bản sao làm việc của bạn có bị sửa đổi không, bạn phải sử dụng hộp thoại "Kiểm tra cho sửa đổi".

Theo mặc định, biểu tượng che phủ và các trình đơn ngữ cảnh sẽ xuất hiện trong tất cả các hộp thoại mở / lưu cũng như trong Windows Explorer. Nếu bạn muốn chúng *chỉ* xuất hiện trong Windows Explorer, chọn hộp kiểm **Hiển thị lớp phủ và trình đơn ngữ cảnh chỉ trong explorer**.

Bạn cũng có thể chọn đánh dấu các thư mục được sửa đổi nếu chúng có các mục chưa phiên bản. Điều này có thể hữu ích để nhắc nhở bạn rằng bạn đã tạo ra các tập tin mới chưa được phiên bản. Tùy chọn này chỉ có sẵn khi bạn sử dụng tùy chọn bộ nhớ cache tình trạng *mặc định* (xem dưới đây).

Các nhóm tiếp theo cho phép bạn chọn các lớp lưu trữ nào để hiển thị lớp phủ. Theo mặc định, chỉ có các ổ đĩa cứng được chọn. Bạn thậm chí có thể vô hiệu hóa tất cả các lớp phủ biểu tượng, nhưng có gì vui trong việc đó?

Ổ đĩa mạng có thể rất chậm, vì vậy các biểu tượng mặc định không được hiển thị cho các bản sao làm việc trên mạng chia sẻ.

Ổ đĩa USB Flash được xem là một trường hợp đặc biệt trong đó loại ổ đĩa được xác định bởi bản thân các thiết bị đó. Một số xuất hiện như ổ đĩa cố định, và một số như là các ổ đĩa di động.

Các **Đường Dẫn Loại Trừ** được sử dụng để báo cho TortoiseSVN những đường dẫn mà nó *không* nên hiển thị biểu tượng lớp phủ và các cột trạng thái. Điều này rất hữu ích nếu bạn có một số bản sao làm việc rất lớn chỉ chứa các thư viện mà bạn sẽ không thay đổi chút nào và do đó không cần các lớp phủ, hoặc nếu bạn chỉ muốn TortoiseSVN tìm trong thư mục cụ thể.

Bất kỳ đường dẫn nào mà bạn chỉ định ở đây được giả định để áp dụng một cách đệ quy, do đó, không có thư mục con nào hiển thị lớp phủ. Nếu bạn *chỉ* muốn loại bỏ thư mục có tên, nói thêm `?` sau đường dẫn.

Điều tương tự cũng áp dụng cho **Bao gồm các đường dẫn**. Ngoại trừ đối với những đường dẫn, các lớp phủ được thể hiện ngay cả khi các lớp phủ bị vô hiệu hóa cho loại ổ đĩa cụ thể, hoặc một đường dẫn loại trừ được chỉ ra ở trên.

Người dùng đôi khi hỏi làm thế nào 3 thiết lập tương tác. Đối với bất kỳ đường dẫn nào, kiểm tra các danh sách bao gồm và loại trừ, tìm kiếm trở lên trên thông qua cấu trúc thư mục cho đến khi một bản khớp được tìm thấy. Khi bản khớp đầu tiên được tìm thấy, tuân thủ nguyên tắc bao gồm hoặc loại trừ. Nếu có xung đột, một chi tiết kỹ thuật thư mục đơn lẻ có được ưu tiên hơn một chi tiết kỹ thuật đệ quy, sau đó việc bao gồm được ưu tiên hơn loại trừ.

Một ví dụ sẽ giúp ở đây:

Loại trừ:

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

Bao gồm:

```
C:\develop
```

Các thiết lập này vô hiệu hóa các lớp phủ biểu tượng cho ổ đĩa C:, trừ `c:\develop`. Tất cả các dự án bên dưới thư mục đó sẽ hiển thị các lớp phủ, ngoại trừ bản thân thư mục `c:\develop`, mà đặc biệt được bỏ qua. Các thư mục nhị phân khuấy đảo cao cũng bị loại trừ.

TSVNCache.exe cũng sử dụng các đường dẫn này để hạn chế chức năng quét của nó. Nếu bạn muốn nó tìm trong các thư mục cụ thể, vô hiệu hóa tất cả các loại ổ đĩa và chỉ bao gồm các thư mục bạn đặc biệt muốn được quét.



## Loại trừ ổ đĩa SUBST

Nó thường được thuận tiện để sử dụng một ổ đĩa SUBST để truy cập các bản sao làm việc của bạn, ví dụ như bằng cách sử dụng lệnh

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

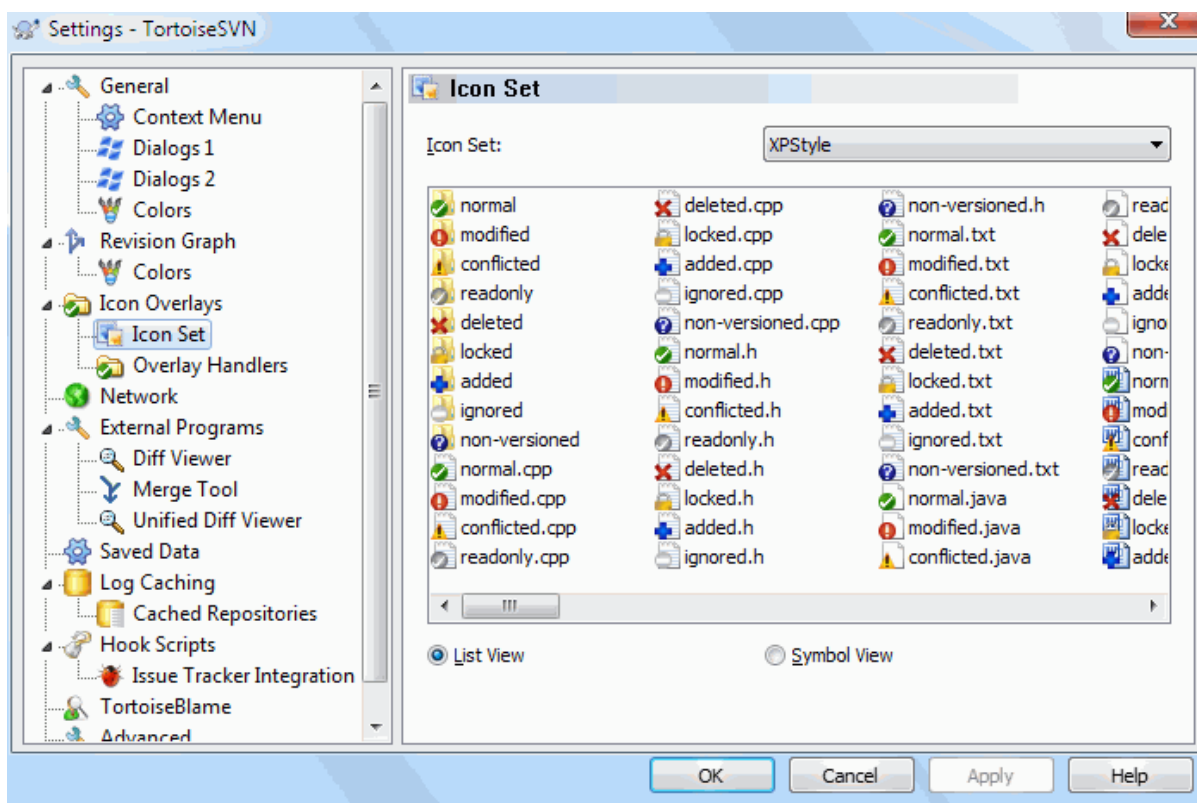
Tuy nhiên điều này có thể gây ra các lớp phủ không cập nhật được, TSVNCache sẽ chỉ nhận được một thông báo khi một thay đổi tập tin, và đó là bình thường cho đường dẫn nguyên thủy. Điều này có nghĩa là lớp phủ của bạn trên đường dẫn subst không bao giờ có thể được cập nhật.

Một cách dễ dàng để làm việc này là loại trừ các đường dẫn nguyên thủy khỏi việc được hiển thị lớp phủ, để từ đó các lớp phủ hiện ra trên đường dẫn subst thay vào đó.

Đôi khi bạn sẽ loại trừ các khu vực có chứa các bản sao làm việc, giúp TSVNCache tránh quét và theo dõi các thay đổi, nhưng bạn vẫn muốn có một dấu hiệu trực quan thể hiện một thư mục có chứa một bản sao làm việc. Các hộp kiểm **Hiển thị thư mục gốc** loại trừ như là 'bình thường' cho phép bạn làm điều này. Với tùy chọn này, thư mục gốc của bản sao trong bất kỳ khu vực bị loại trừ (loại ổ đĩa không được chọn, hoặc bị loại trừ) sẽ hiển thị như bình thường và cập nhật, với một dấu kiểm màu xanh lá cây. Điều này nhắc nhở bạn rằng bạn đang nhìn vào một bản sao làm việc, mặc dù các lớp phủ thư mục có thể không được chính xác. Các tập tin không nhận được một lớp phủ. Lưu ý rằng các trình đơn ngữ cảnh vẫn làm việc, mặc dù các lớp phủ không được hiển thị.

Là một ngoại lệ đặc biệt cho việc này, các ổ đĩa A: và B: không bao giờ được xem xét cho tùy chọn **Hiển thị các thư mục loại trừ** như 'bình thường'. Điều này là bởi vì Windows buộc phải nhìn vào ổ đĩa, mà có thể dẫn đến một sự chậm trễ một vài giây khi bắt đầu Explorer, ngay cả khi máy tính của bạn có một ổ đĩa mềm.

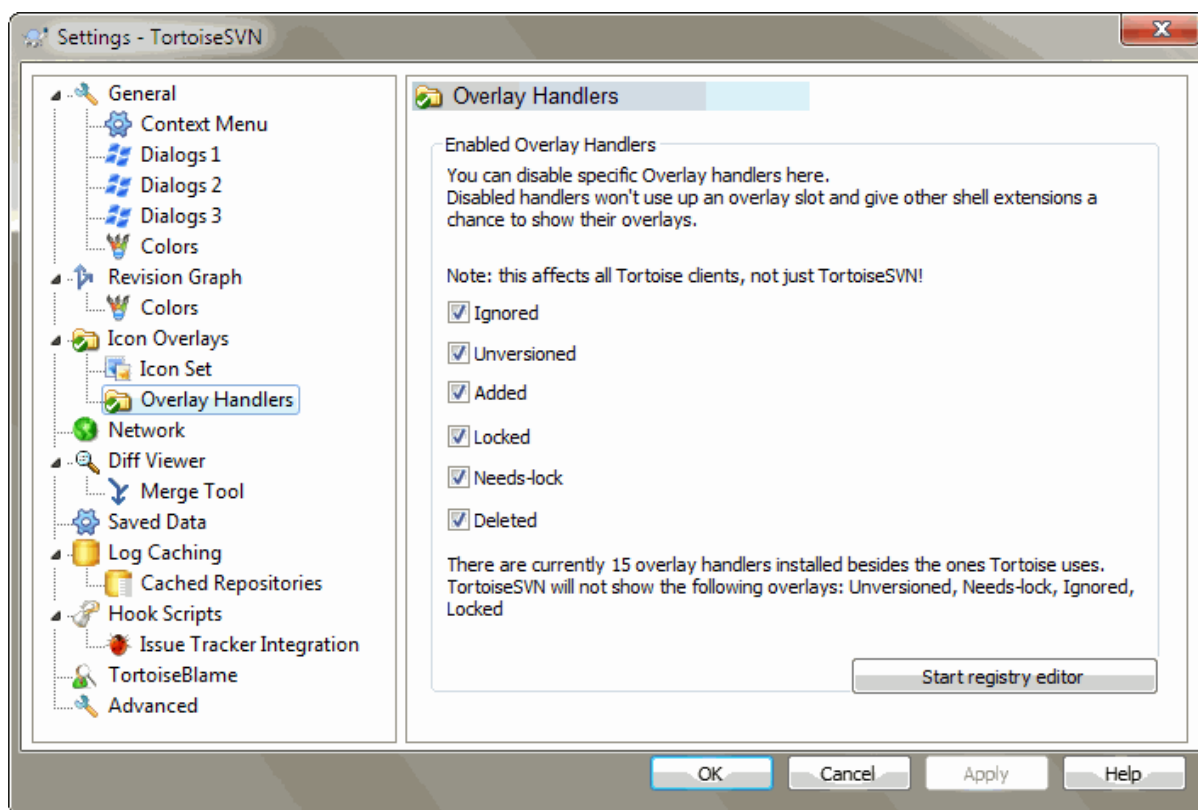
### 4.30.3.1. Lựa Chọn Tập Hợp Biểu Tượng



Hình 4.72. Hộp Thoại Thiết Lập, Trang Tập Hợp Biểu Tượng

Bạn có thể thay đổi tập hợp biểu tượng lớp phủ cho cái bạn thích nhất. Lưu ý rằng nếu bạn thay đổi tập hợp lớp phủ, bạn có thể phải khởi động lại máy tính của bạn để những thay đổi có hiệu lực.

### 4.30.3.2. Bật Các Trình Xử Lý Lớp Phủ

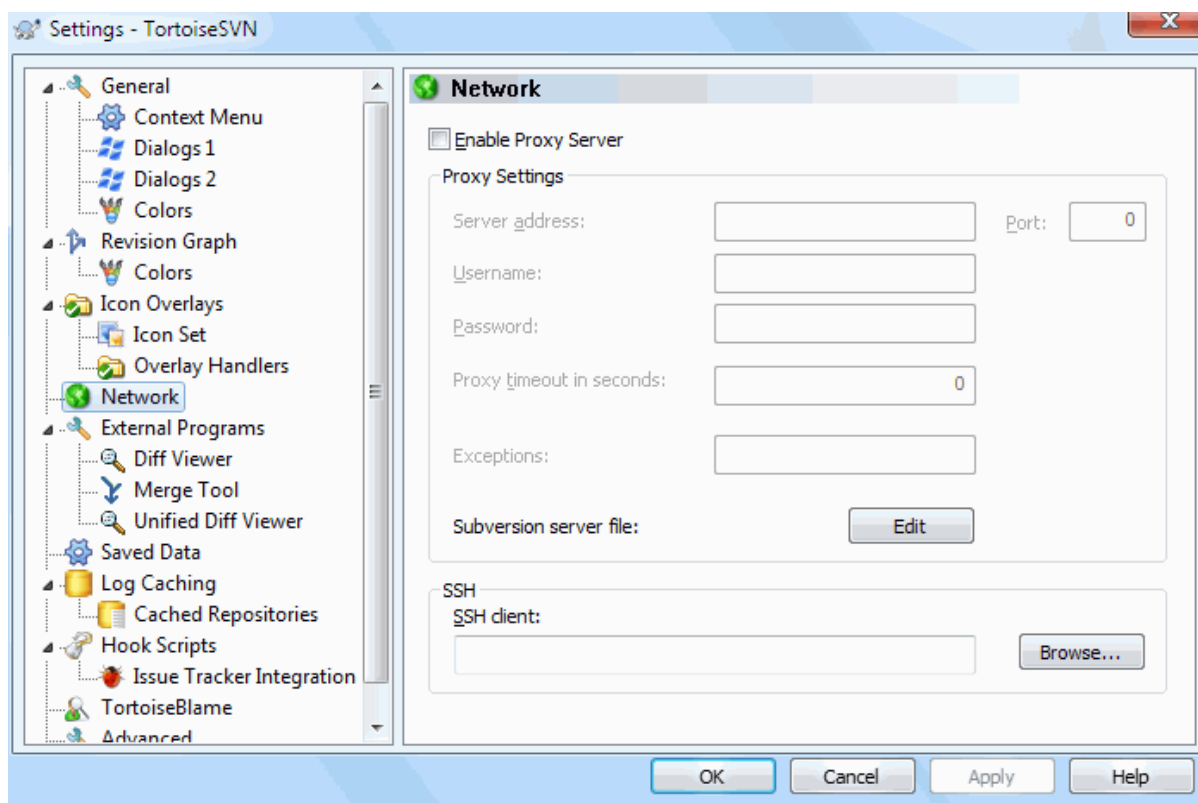


**Hình 4.73. Hộp Thoại Thiết Lập, Trang Xử Lý Biểu Tượng**

Bởi vì số lượng các lớp phủ có sẵn bị hạn chế, bạn có thể chọn để vô hiệu hóa một số xử lý để đảm bảo rằng những cái mà bạn muốn sẽ được nạp. Bởi vì TortoiseSVN sử dụng thành phần TortoiseOverlays phổ biến được chia sẻ với các trình khách Tortoise khác (ví dụ như TortoiseCVS, TortoiseHg) thiết lập này cũng sẽ ảnh hưởng đến những trình khách đó.

### 4.30.4. Các Thiết Lập Mạng





**Hình 4.74. Hộp Thoại Thiết Lập, Trang Mạng**

Ở đây bạn có thể cấu hình máy chủ proxy của bạn, nếu bạn cần cái một để đi thông qua tường lửa của công ty bạn.

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.

Bạn cũng có thể chỉ định chương trình nào mà TortoiseSVN nên sử dụng để thiết lập một kết nối an toàn với một kho lưu trữ `svn+ssh`. Chúng tôi khuyên bạn nên sử dụng `TortoisePlink.exe`. Đây là một phiên bản của chương trình phổ biến `Plink`, và được bao gồm với TortoiseSVN, nhưng nó được biên dịch như là một ứng dụng không dùng cửa sổ, do đó bạn không nhận được một hộp DOS bật lên mỗi khi bạn xác thực đăng nhập.

Bạn phải chỉ định đường dẫn đầy đủ tới tập tin thực thi. Đối với `TortoisePlink.exe` đây là thư mục TortoiseSVN bin tiêu chuẩn. Sử dụng nút **Duyệt** để giúp xác định vị trí nó. Lưu ý nếu đường dẫn có khoảng trắng, bạn phải đặt nó bên trong dấu ngoặc kép, ví dụ

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

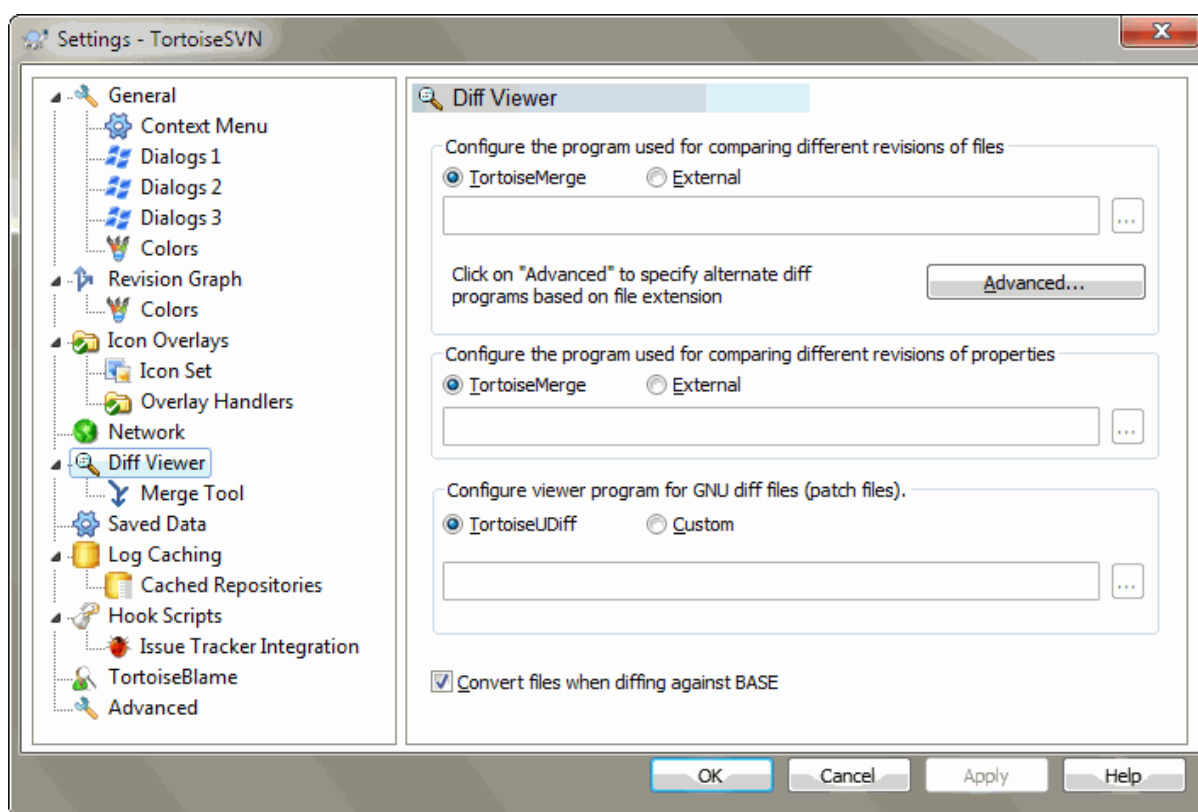
Một tác dụng phụ của việc không có một cửa sổ là không có nơi nào cho bất kỳ thông báo lỗi, vì vậy nếu việc chứng thực không thành, bạn chỉ đơn giản là sẽ nhận được một thông báo nói rằng một cái gì đó như “ Không thể ghi vào đầu ra tiêu chuẩn ”. Vì lý do này, chúng tôi đề nghị rằng trước tiên bạn thiết lập bằng cách sử dụng tiêu chuẩn `Plink`. Khi tất cả mọi thứ đang làm việc, bạn có thể sử dụng `TortoisePlink` với chính xác cùng các thông số.

`TortoisePlink` không có bất kỳ tài liệu của riêng nó bởi vì nó chỉ là một biến thể nhỏ của `Plink`. Tìm hiểu về các tham số dòng lệnh từ [trang web PuTTY](http://www.chiark.greenend.org.uk/~sgtatham/putty/) [http://www.chiark.greenend.org.uk/~sgtatham/putty/].

Để tránh bị nhắc nhở cho một mật khẩu lặp đi lặp lại, bạn cũng có thể xem xét sử dụng một công cụ bộ nhớ đệm mật khẩu chẳng hạn như `Pageant`. Nó cũng có sẵn để tải về từ trang web của `PuTTY`.

Cuối cùng, thiết lập SSH trên máy chủ và khách là một quá trình không tầm thường đó là vượt ra ngoài phạm vi của tập tin trợ giúp này. Tuy nhiên, bạn có thể tìm thấy một hướng dẫn trong FAQ của TortoiseSVN được liệt kê dưới [Subversion / TortoiseSVN SSH Làm thế nào](http://tortoisesvn.net/ssh_howto) [http://tortoisesvn.net/ssh\_howto].

## 4.30.5. Các Thiết Lập Cho Chương Trình Bên Ngoài



**Hình 4.75. Hộp Thoại Thiết Lập, Trang Xem Khác Biệt**

Ở đây bạn có thể định nghĩa chương trình khác / hộp nhắt của riêng bạn mà TortoiseSVN nên sử dụng. Thiết lập mặc định là sử dụng TortoiseMerge được cài đặt cùng với TortoiseSVN.

Đọc [Phần 4.10.6, “Công cụ Khác Biệt / Hộp Nhắt Bên ngoài”](#) cho một danh sách một số các chương trình khác biệt / hộp nhắt bên ngoài mà mọi người đang sử dụng với TortoiseSVN.

### 4.30.5.1. Trình Xem Khác Biệt

Một chương trình tìm khác biệt bên ngoài có thể được sử dụng để so sánh các bản sửa đổi khác nhau của tập tin. Các chương trình bên ngoài sẽ cần có được những tên tập tin từ dòng lệnh, cùng với bất kỳ tùy chọn dòng lệnh khác. TortoiseSVN sử dụng các thông số thay thế với tiền tố % . Khi gặp một trong những cái này nó sẽ thay thế các giá trị thích hợp. Trình tự của các tham số sẽ phụ thuộc vào chương trình Tìm Khác Biệt bạn sử dụng.

%base

Các tập tin nguyên thủy mà không có thay đổi của bạn

%bname

Tiêu đề cửa sổ cho tập tin cơ sở

%mine

Tập tin riêng của bạn, với những thay đổi của bạn

%yname

Tiêu đề cửa sổ cho tập tin của bạn

%burl

URL của tập tin gốc, nếu có sẵn

`%yurl`  
URL của tập tin thứ hai, nếu có sẵn

`%brev`  
Các sửa đổi của tập tin gốc, nếu có

`%yrev`  
Việc sửa đổi các tập tin thứ hai, nếu có

`%peg`  
Bản sửa đổi peg, nếu có

Các tiêu đề cửa sổ không phải là tên tập tin thuần túy. TortoiseSVN coi chúng như là một cái tên để hiển thị và tạo ra các tên phù hợp. Vì vậy, ví dụ như nếu bạn đang làm một việc tìm khác biệt từ một tập tin trong bản sửa đổi 123 với một tập tin trong bản sao làm việc của bạn, các tên sẽ là `tên tập tin: sửa đổi 123` và `tên tập tin: bản sao làm việc` .

Ví dụ, với ExamDiff Pro:

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

or with KDiff3:

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

or with WinMerge:

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

or with Araxis:

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

or with UltraCompare:

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

or with DiffMerge:

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

Nếu bạn sử dụng thuộc tính `svn:keyword` để mở rộng các từ khóa, và đặc biệt là *sửa đổi* của một tập tin, thì có thể có một sự khác biệt giữa các tập tin mà hoàn toàn do giá trị hiện tại của từ khóa. Ngoài ra nếu bạn sử dụng `svn:eol-style = native` các tập tin CỎ SỔ sẽ có kết thúc dòng LF thuần túy trong khi tập tin của bạn sẽ có kết thúc dòng CR-LF. TortoiseSVN bình thường sẽ tự động ẩn những khác biệt này bằng cách phân tích các tập tin CỎ SỔ để mở rộng các từ khóa và kết thúc dòng trước khi thực hiện các hoạt động khác. Tuy nhiên, điều này có thể mất một thời gian dài với các tập tin lớn. Nếu **Chuyển đổi các tập tin** khi tìm khác biệt với CỎ SỔ không được đánh dấu thì TortoiseSVN sẽ bỏ qua việc tiền xử lý các tập tin.

Bạn cũng có thể chỉ định một công cụ tìm khác biệt khác nhau để sử dụng trên các thuộc tính Subversion. Bởi vì những việc này có xu hướng là các chuỗi văn bản ngắn đơn giản, bạn có thể muốn sử dụng một trình xem đơn giản nhỏ gọn hơn.

Nếu bạn đã cấu hình một công cụ tìm khác biệt thay thế, bạn có thể truy cập TortoiseMerge và công cụ của bên thứ ba từ trình đơn ngữ cảnh. Trình đơn ngữ cảnh → Khác Biệt sử dụng các công cụ tìm khác biệt chính, và **Shift** + Trình đơn ngữ cảnh → Khác Biệt sử dụng công cụ tìm khác biệt thứ cấp.

Tại cuối hộp thoại bạn có thể cấu hình một chương trình xem các tập tin khác biệt thống nhất (các tập tin vá). Không cần tham số nào. Thiết lập Default là để dùng TortoiseUDiff được cài đặt theo TortoiseSVN, và dùng các mã màu cho các dòng được thêm và xóa.

Bởi vì Khác Biệt Thống Nhất chỉ là một định dạng văn bản, bạn có thể sử dụng trình soạn thảo văn bản yêu thích của bạn nếu bạn thích.

#### 4.30.5.2. Công Cụ Hợp Nhất

Một chương trình hợp nhất bên ngoài được sử dụng để giải quyết các tập tin mâu thuẫn. Thay thế tham số được sử dụng trong cùng một cách như với Chương trình Diff.

`%base`  
tập tin ban đầu mà không có thay đổi của bạn hoặc người khác

`%bname`  
Tiêu đề cửa sổ cho tập tin cơ sở

`%mine`  
tập tin của riêng bạn, với những thay đổi của bạn

`%yname`  
Tiêu đề cửa sổ cho tập tin của bạn

`%theirs`  
tập tin như nó có trong kho

`%tname`  
Tiêu đề cửa sổ cho các tập tin trong kho

`%merged`  
tập tin bị xung đột, kết quả của tác vụ hợp nhất

`%mname`  
Tiêu đề cửa sổ cho các tập tin bị sáp nhập

Ví dụ, với 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
```

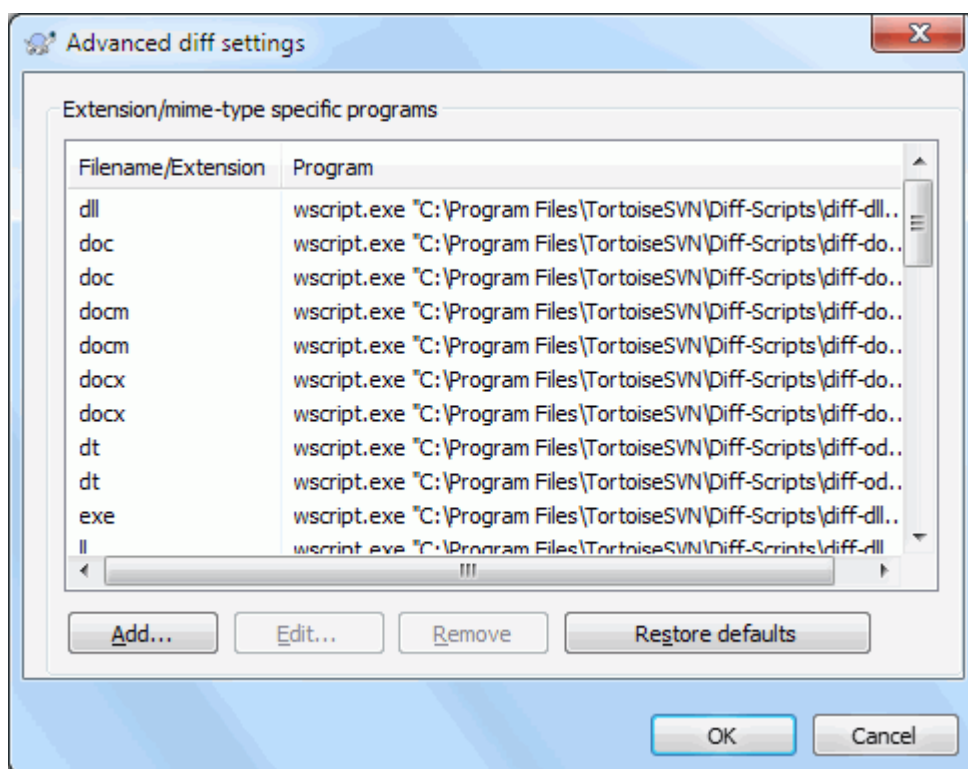
or with WinMerge (2.8 or later):

```
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
```

### 4.30.5.3. Thiết Lập Nâng Cao Cho Khác Biệt/Hợp Nhất

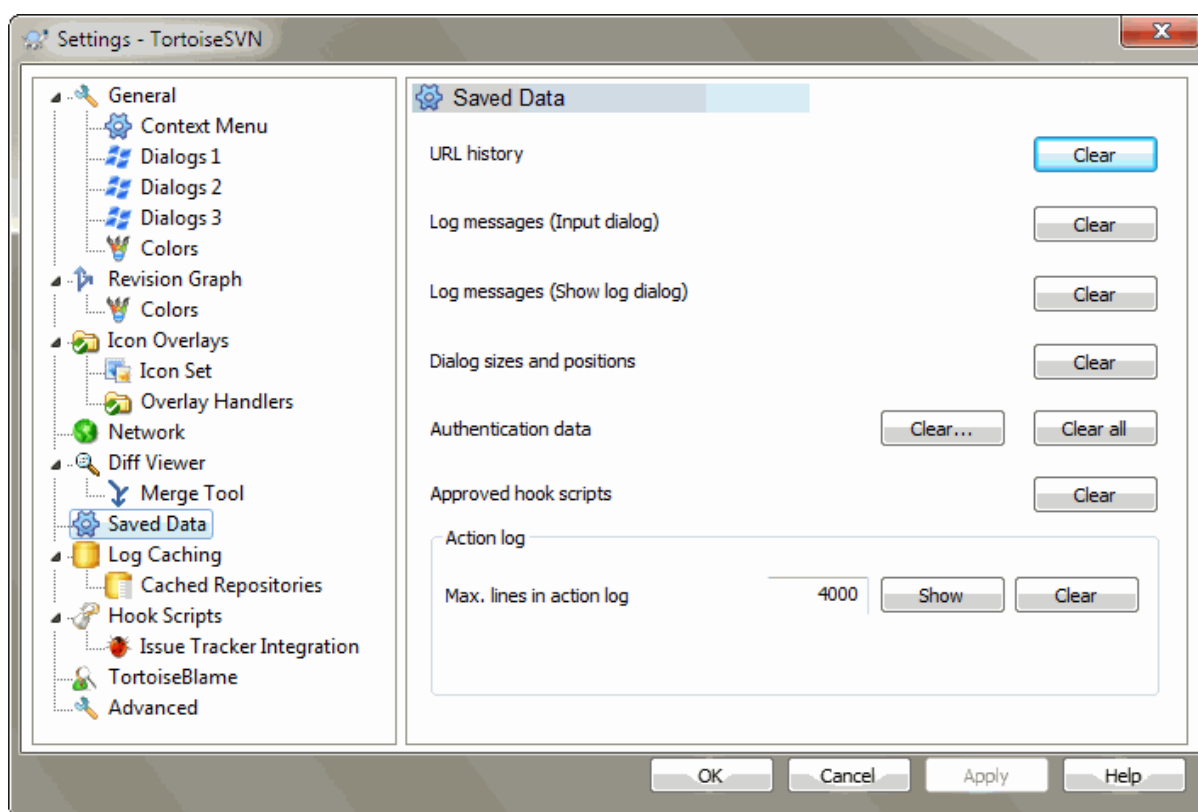


**Hình 4.76. Hộp thoại Thiết Lập, Hộp Thoại Khác Biệt/Hợp Nhất Nâng Cao**

Trong các thiết lập nâng cao, bạn có thể xác định một chương trình tìm khác biệt và hợp nhất khác nhau cho mỗi phần mở rộng của tập tin. Ví dụ, bạn có thể kết hợp Photoshop là Chương trình “Khác Biệt” cho các tập tin .jpg :-). Bạn cũng có thể kết hợp thuộc tính `svn:mime-type` với một chương trình khác biệt hoặc hợp nhất.

Để kết hợp bằng cách sử dụng một phần mở rộng tập tin, bạn cần phải xác định phần mở rộng. Sử dụng `.bmp` để mô tả các tập tin ảnh Windows. Để kết hợp bằng cách sử dụng thuộc tính `svn:mime-type`, xác định loại mime, trong đó có một dấu xuyệt, ví dụ `text/xml`.

## 4.30.6. Thiết Lập Cho Dữ Liệu Đã Lưu



**Hình 4.77. Hộp Thoại Thiết Lập, Trang Đã Lưu Dữ Liệu**

Để tiện cho bạn, TortoiseSVN lưu rất nhiều các thiết lập mà bạn sử dụng, và nhớ nơi bạn đã tới trong thời gian gần đây. Nếu bạn muốn dọn dẹp bộ nhớ cache dữ liệu, bạn có thể làm nó ở đây.

### lịch sử của URL

Bất cứ khi nào bạn kiểm xuất một bản sao hoạt động, sáp nhập, thay đổi hoặc sử dụng trình duyệt kho lưu trữ, TortoiseSVN giữ một bản ghi của các URL gần đây được sử dụng và cung cấp chúng trong một hộp kết hợp. Đôi khi danh sách đó bị lộn xộn với các URL đã lỗi thời vì vậy rất hữu ích để đẩy nó ra định kỳ.

Nếu bạn muốn loại bỏ một mục duy nhất từ một trong các hộp kết hợp, bạn có thể làm điều đó tại chỗ. Chỉ cần nhấp vào mũi tên để thả các hộp kết hợp, di chuyển chuột lên mục mà bạn muốn loại bỏ và loại **Shift+Del**.

### Thông điệp tường trình (Hộp thoại nhập liệu)

TortoiseSVN lưu trữ các thông điệp tường trình cho cam kết gần đây mà bạn nhập vào. Chúng được lưu trữ theo từng kho lưu trữ, vì vậy nếu bạn truy cập vào nhiều kho lưu trữ danh sách này có thể phát triển khá lớn.

### Các thông điệp tường trình (Hiện thị hộp thoại tường trình)

TortoiseSVN lưu trữ các thông điệp tường trình được nạp bởi hộp thoại Hiện Thị Tường Trình để tiết kiệm thời gian khi bạn hiện thị tường trình lần tới. Nếu một người nào khác chỉnh sửa một thông điệp tường trình và bạn đã có các thông điệp đó được lưu trữ, bạn sẽ không thấy sự thay đổi cho đến khi bạn xóa bộ nhớ cache. Bộ nhớ đệm cho thông điệp tường trình được kích hoạt trên tab **Bộ nhớ đệm cho tường trình**.

### Kích thước và vị trí hộp thoại

Nhiều hộp thoại nhớ kích thước và vị trí màn hình mà bạn sử dụng lần cuối.

### Dữ liệu xác thực

Khi bạn xác thực với một máy chủ Subversion, tên người dùng và mật khẩu được lưu trữ tại địa phương do đó bạn không phải tiếp tục nhập. Bạn có thể muốn rõ ràng điều này vì lý do an ninh, hoặc bởi vì bạn muốn

truy cập vào kho lưu trữ dưới một tên người dùng khác nhau ... John có biết bạn đang sử dụng máy tính của mình không?

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

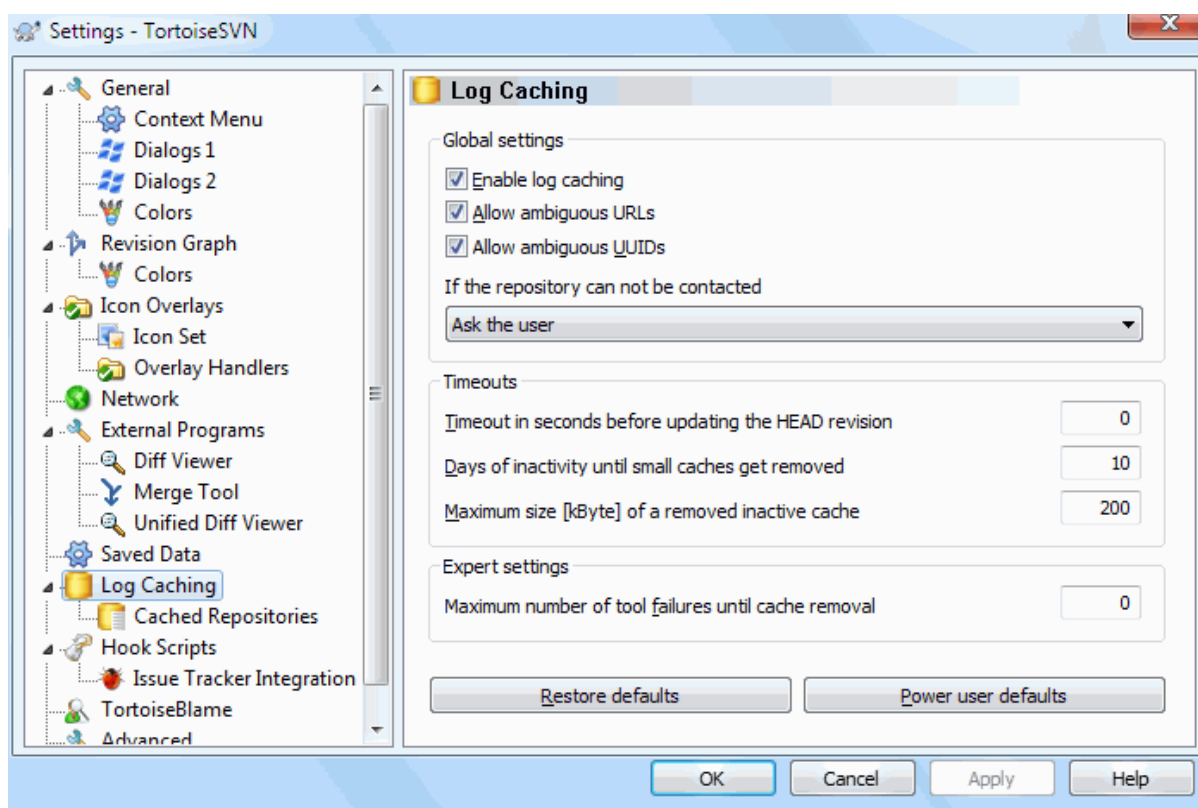
#### Tường trình hành động

TortoiseSVN giữ một nhật ký của tất cả mọi thứ ghi vào các hộp thoại tiến triển của nó. Điều này có thể hữu ích, ví dụ, khi bạn muốn kiểm tra những gì đã xảy ra trong một lệnh cập nhật gần đây.

Các tập tin tường trình bị giới hạn về chiều dài và khi nó phát triển quá lớn nội dung lâu đời nhất sẽ bị bỏ đi. Theo mặc định 4000 dòng được lưu giữ, nhưng bạn có thể tùy chỉnh con số đó.

Từ đây, bạn có thể xem nội dung của tập tin tường trình, và cũng có thể xóa nó.

### 4.30.7. Lưu Đệm Tường Trình



**Hình 4.78. Hộp Thoại Thiết Lập, Trang bộ nhớ đệm tường trình**

Hộp thoại này cho phép bạn cấu hình các tính năng bộ nhớ đệm tường trình của TortoiseSVN, giữ lại một bản sao địa phương của các thông điệp tường trình và thay đổi đường dẫn để tránh tốn thời gian tải từ máy chủ. Sử dụng bộ nhớ cache tường trình có thể tăng đáng kể tốc độ của hộp thoại tường trình và đồ thị sửa đổi. Một tính năng hữu ích là các thông điệp tường trình vẫn có thể được truy cập khi ngoại tuyến.

#### Kích hoạt tính năng bộ nhớ đệm tường trình

Cho phép bộ nhớ đệm tường trình bất cứ khi nào dữ liệu tường trình được yêu cầu. Nếu được duyệt, dữ liệu sẽ được lấy từ bộ nhớ đệm khi có sẵn, và bất kỳ tường trình nào không có trong bộ nhớ đệm sẽ được lấy từ máy chủ và thêm vào bộ nhớ đệm.

Nếu bộ nhớ đệm bị vô hiệu hóa, dữ liệu sẽ luôn luôn được lấy trực tiếp từ máy chủ và không được lưu trữ tại địa phương.

#### Cho phép các URL không rõ ràng

Đôi khi bạn có thể kết nối với một máy chủ mà sử dụng cùng một URL cho tất cả các kho. Các phiên bản cũ của `svnbridge` sẽ làm điều này. Nếu bạn cần truy cập vào kho như vậy, bạn sẽ phải duyệt tùy chọn này. Nếu bạn không muốn, hãy để nó không được duyệt để cải thiện hiệu suất.

#### Cho phép các UUID mơ hồ

Một số dịch vụ lưu trữ cung cấp cho tất cả các kho chứa của chúng cùng một UUID. Bạn thậm chí có thể đã làm điều này bằng cách sao chép một thư mục kho lưu trữ để tạo cái mới. Đối với tất cả các loại lý do đây là một ý tưởng tồi - một UUID nên là *duy nhất*. Tuy nhiên, bộ nhớ đệm tường trình vẫn sẽ làm việc trong tình huống này nếu bạn chọn hộp kiểm này. Nếu bạn không cần nó, để nó không được chọn để cải thiện hiệu suất.

#### Nếu các kho lưu trữ không thể liên lạc được

Nếu bạn đang làm việc ngoại tuyến, hoặc nếu máy chủ kho không hoạt động, bộ nhớ đệm tường trình vẫn có thể được sử dụng để cung cấp các thông báo tường trình đã được tổ chức trong bộ nhớ đệm. Tất nhiên, bộ nhớ đệm có thể không được cập nhật, vì vậy có những lựa chọn để cho phép bạn lựa chọn có dùng tính năng này hay không.

Khi dữ liệu đệm được lấy từ bộ nhớ đệm mà không cần liên hệ với máy chủ, hộp thoại bằng cách sử dụng những thông điệp đó sẽ hiển thị trạng thái ngoại tuyến trong thanh tiêu đề của nó.

#### Hết thời gian chờ trước khi cập nhật bản sửa đổi CHÍNH

Khi bạn gọi hộp thoại tường trình, bạn thường sẽ muốn liên lạc với máy chủ để kiểm tra có thông điệp tường trình mới không. Nếu thiết lập thời gian chờ hết giờ ở đây là khác không thì máy chủ sẽ chỉ được liên lạc, khi thời gian chờ đã trôi qua kể từ khi liên lạc lần cuối cùng. Điều này có thể làm giảm các chuyến đi vòng tròn của máy chủ nếu bạn mở hộp thoại tường trình thường xuyên và máy chủ bị chậm, nhưng các dữ liệu được hiển thị có thể không hoàn toàn cập nhật. Nếu bạn muốn sử dụng tính năng này, chúng tôi đề nghị sử dụng một giá trị từ 300 (5 phút) như một sự thỏa hiệp.

#### Số ngày không hoạt động cho đến khi các bộ đệm nhỏ có được loại bỏ

Nếu bạn duyệt qua xung quanh rất nhiều các kho, bạn sẽ tích lũy rất nhiều bộ nhớ đệm tường trình. Nếu bạn không chủ động sử dụng chúng, bộ nhớ đệm sẽ không phát triển rất lớn, vì vậy TortoiseSVN cuộc thanh lọc chúng sau một thời gian thiết lập mặc định. Sử dụng mục này để kiểm soát việc thanh lọc bộ nhớ đệm.

#### Kích thước tối đa của các bộ lưu trữ đệm không hoạt động bị loại bỏ

Bộ lưu trữ đệm lớn hơn sẽ tốn kém nhiều hơn để nạp lại, vì vậy TortoiseSVN chỉ thanh lọc các bộ đệm nhỏ. Tinh chỉnh các ngưỡng với giá trị này.

#### Số lượng tối đa các thất bại của công cụ trước khi loại bỏ bộ nhớ đệm

Đôi khi có gì đó sai sót với bộ nhớ đệm và gây ra một vụ đổ vỡ chương trình. Nếu điều này xảy ra bộ nhớ đệm thường được xóa tự động để ngăn chặn sự tái phát của vấn đề. Nếu bạn sử dụng các phiên bản xây dựng hàng đêm ít ổn định, bạn có thể lựa chọn để giữ cho bộ nhớ đệm.

### 4.30.7.1. Kho Được Lưu Trữ Đệm

Trên trang này, bạn có thể thấy một danh sách của kho được lưu trữ tại địa phương, và không gian sử dụng cho bộ nhớ đệm. Nếu bạn chọn một trong các kho, sau đó bạn có thể sử dụng các nút bên dưới.

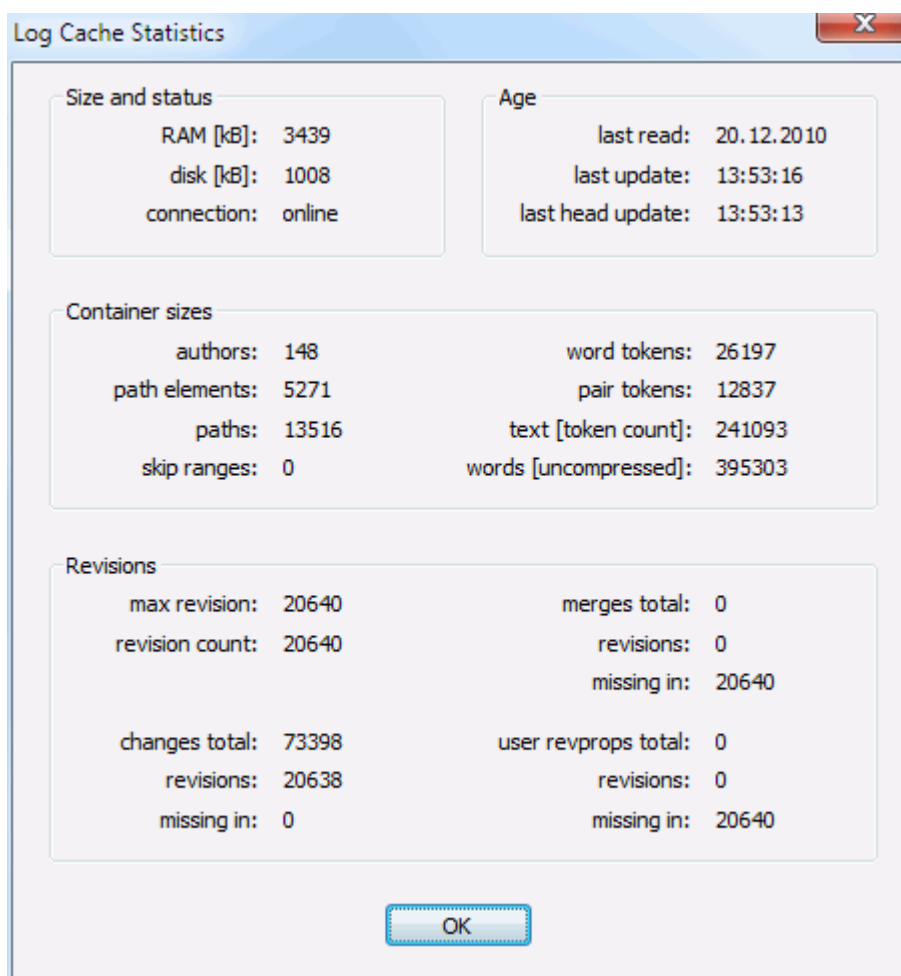
Click vào **Cập nhật** để hoàn toàn làm mới bộ nhớ đệm và điền vào bất cứ lỗ hổng. Đối với một kho lưu trữ lớn, điều này có thể sẽ rất tốn thời gian, nhưng hữu ích nếu bạn đang chuẩn bị ngoại tuyến và muốn bộ nhớ đệm tốt nhất hiện có.

Bấm vào nút **Xuất khẩu** để xuất khẩu toàn bộ bộ nhớ đệm như là một tập hợp các tập tin CSV. Điều này có thể hữu ích nếu bạn muốn xử lý dữ liệu tường trình bằng cách sử dụng một chương trình bên ngoài, mặc dù nó là chủ yếu hữu ích cho các nhà phát triển.

Bấm vào **Xóa** để loại bỏ tất cả các dữ liệu lưu trữ trong các kho được lựa chọn. Điều này không vô hiệu hóa bộ nhớ đệm cho các kho lưu trữ để lần sau khi bạn yêu cầu dữ liệu tường trình, một bộ nhớ đệm mới sẽ được tạo ra.



#### 4.30.7.2. Thống Kê Bộ Nhớ Đệm Lưu Trữ Tường Trình



**Hình 4.79. Hộp Thoại Thiết Lập, Thống kê Bộ Nhớ Đệm Lưu Trữ Tường Trình**

Bấm vào nút **Thông tin chi tiết** để xem thống kê chi tiết cho một bộ nhớ đệm cụ thể. Nhiều lĩnh vực được hiển thị ở đây chủ yếu là được các nhà phát triển của TortoiseSVN quan tâm, vì vậy không phải tất cả chúng được mô tả chi tiết.

##### RAM

Số lượng bộ nhớ cần thiết để phục vụ bộ nhớ đệm này.

##### Đĩa

Số lượng không gian đĩa được sử dụng cho bộ nhớ đệm. Dữ liệu được nén, do đó, lượng đĩa sử dụng nói chung là khá khiêm tốn.

##### Kết nối

Hiện thị có hay không việc các kho lưu trữ đã có sẵn lần cuối mà bộ nhớ đệm được sử dụng.

##### Lần cập nhật cuối

Lần cuối nội dung bộ nhớ đệm đã được thay đổi.

##### Lần cuối cập nhật chính

Lần cuối chúng tôi yêu cầu bản sửa đổi CHÍNH từ máy chủ.

##### Tác giả

Số tác giả khác nhau với những thông điệp được ghi trong bộ nhớ đệm.

Các đường d

Số lượng các đường dẫn được liệt kê, bạn sẽ thấy sử dụng `svn log -v`.

Bỏ qua các phạm vi

Số lượng các phạm vi sửa đổi mà chúng tôi đã không nạp, đơn giản chỉ đơn giản vì chúng đã không được yêu cầu. Đây là một cách đo số lượng lỗi hỏng trong bộ nhớ đệm.

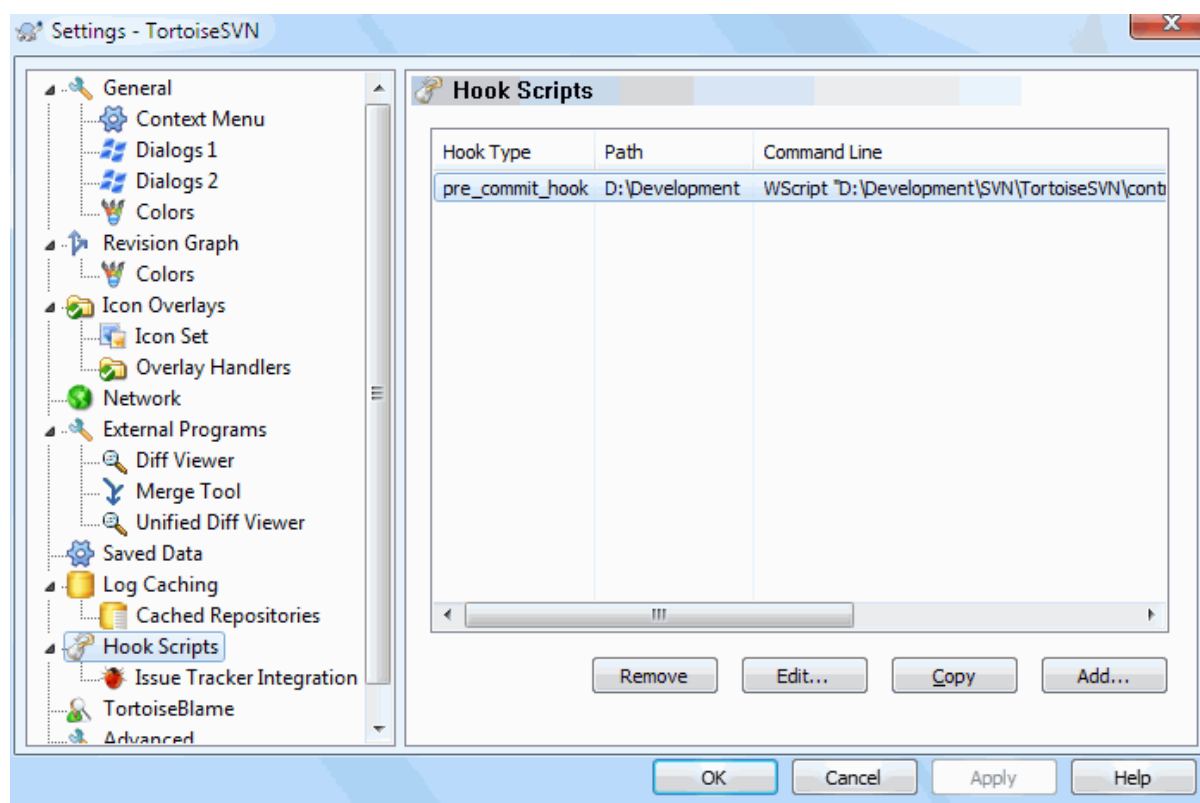
Số lượng sửa đổi lớn nhất

Số phiên bản cao nhất được lưu trữ trong bộ nhớ đệm.

Số lượng sửa đổi

Số lượng các phiên bản được lưu trữ trong bộ nhớ đệm. Đây là một cách đo tính đầy đủ của bộ nhớ đệm.

#### 4.30.8. Kịch Bản Hook Phía Máy Khách

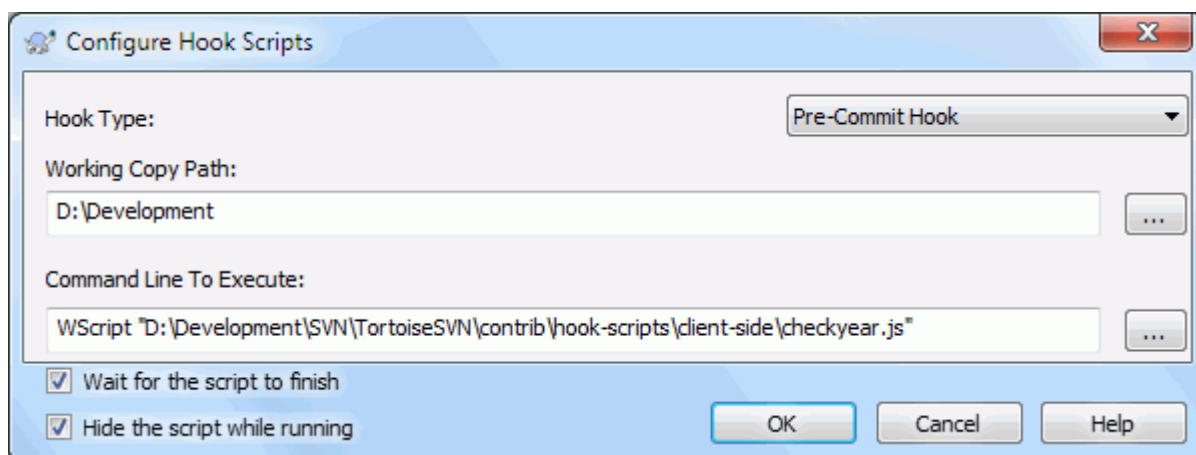


**Hình 4.80. Hộp Thoại Thiết Lập, Trang Kịch Bản Hook**

Hộp thoại này cho phép bạn thiết lập các kịch bản móc mà sẽ được thực hiện tự động khi một số hành động Subversion nhất định được thực hiện. Trái ngược với các kịch bản móc giải thích trong [Phần 3.3, “kịch bản móc phía máy chủ”](#), các kịch bản này được thực hiện tại địa phương trên máy khách.

Một ứng dụng cho các móc này có thể là để gọi một chương trình như `SubWCRev.exe` để cập nhật số phiên bản sau một cam kết, và có lẽ để kích hoạt việc xây dựng lại.

Note that you can also specify such hook scripts using special properties on your working copy. See the section [Phần 4.17.2, “Thuộc tính dự án TortoiseSVN”](#) for details.



**Hình 4.81. Hộp Thoại Thiết Lập, Cấu Hình Kịch Bản Hook**

Để thêm một kịch bản móc mới, chỉ cần bấm vào **Thêm** và điền vào các chi tiết.

Hiện nay có sáu loại kịch bản móc có sẵn

#### Bắt đầu-cam kết

Được gọi là trước khi hộp thoại cam kết được hiển thị. Bạn có thể muốn sử dụng điều này nếu hook thay đổi một tập tin được phiên bản và ảnh hưởng đến danh sách các tập tin cần được cam kết và / hoặc thông báo cam kết. Tuy nhiên bạn nên lưu ý rằng bởi vì hook được gọi là ở giai đoạn đầu, danh sách đầy đủ của các đối tượng được lựa chọn cho cam kết là không có sẵn.

#### Tiền-cam kết

Được gọi sau khi người dùng nhấp chuột vào **OK** trong hộp thoại cam kết, và trước khi cam kết thực sự bắt đầu. Hook này có một danh sách chính xác những gì sẽ được thực hiện.

#### Hậu-cam kết

Được gọi là sau khi cam kết hoàn tất (cho dù thành công hay không).

#### Khởi động-cập nhật

Được gọi là trước khi hộp thoại cập nhật-đến-bản sửa đổi được hiển thị.

#### Tiền-cập nhật

Được gọi là trước khi cập nhật Subversion thực tế hoặc chuyển đổi bắt đầu.

#### Hậu-cập nhật

Được gọi là sau khi cập nhật, chuyển đổi, hoặc kiểm xuất kết thúc (cho dù thành công hay không).

#### Tiền-Kết nối

Được gọi trước khi cố gắng liên lạc với các kho lưu trữ. Được gọi nhiều nhất một lần trong năm phút.

Móc được định nghĩa cho một đường dẫn bản sao làm việc đặc biệt. Bạn chỉ cần phải xác định đường dẫn cấp cao nhất; nếu bạn thực hiện một tác vụ trong một thư mục con, TortoiseSVN sẽ tự động tìm kiếm trở lên cho một đường dẫn phù hợp.

Tiếp theo, bạn phải xác định dòng lệnh để thực hiện, bắt đầu với đường dẫn đến kịch bản móc hoặc tập tin thực thi. Điều này có thể là một tập tin batch, một tập tin thực thi hay tập tin nào khác trong đó có một cửa sổ kết hợp tập tin hợp lệ, ví dụ như một kịch bản perl. Lưu ý rằng kịch bản không phải được xác định bằng cách sử dụng một đường dẫn UNC bởi Windows shell thực thi sẽ không cho phép các kịch bản như vậy hoạt động do hạn chế bảo mật.

Các dòng lệnh bao gồm một số thông số được điền bởi TortoiseSVN. Các thông số thông qua phụ thuộc vào móc nào được gọi. Mỗi móc có các tham số riêng của mình được thông qua theo trình tự sau:

#### Bắt đầu-cam kết

PATHMESSAGEFILECWD

#### Tiền-cam kết

PATHDEPTHMESSAGEFILECWD

#### Hậu-cam kết

PATHDEPTHMESSAGEFILEREVISIONERRORCWD

#### Khởi động-cập nhật

PATHCWD

#### Tiền-cập nhật

PATHDEPTHREVISIONCWD

#### Hậu-cập nhật

PATHDEPTHREVISIONERRORCWD

#### Tiền-Kết nối

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

Ý nghĩa của các tham số này được mô tả ở đây:

#### PATH

Một đường dẫn vào một tập tin tạm thời trong đó có tất cả các đường dẫn mà tác vụ được bắt đầu. Mỗi đường dẫn là một dòng riêng trong tập tin tạm thời.

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

Độ sâu mà cam kết / cập nhật được thực hiện.

Giá trị có thể là:

-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

Đường dẫn đến một tập tin có chứa các thông điệp tường trình để thực hiện cam kết. Tập tin có chứa văn bản trong bảng mã UTF-8. Sau khi thực hiện thành công móc-bắt đầu-cam kết, các thông điệp tường trình được đọc lại, cho móc một cơ hội để sửa đổi nó.

#### REVISION

Bản sửa đổi kho mà cập nhật nên được thực hiện hoặc sau khi một cam kết hoàn tất.

## ERROR

Đường dẫn vào một tập tin có chứa các thông báo lỗi. Nếu không có lỗi, tập tin sẽ được làm rỗng.

## CWD

Thư mục làm việc hiện tại mà kịch bản được chạy. Nó được thiết lập vào thư mục gốc chung của tất cả các đường dẫn bị ảnh hưởng.

Lưu ý rằng mặc dù chúng tôi đã đưa ra những tên thông số để cho thuận tiện, bạn không phải tham khảo những tên đó trong các thiết lập móc. Tất cả các tham số được liệt kê cho một móc cụ thể luôn luôn được thông qua, cho dù bạn muốn hay không ;-)

Nếu bạn muốn tác vụ Subversion chờ đến khi móc đã hoàn thành, kiểm tra `Chờ cho kịch bản kết thúc` .

Thông thường bạn sẽ muốn ẩn các hộp DOS xấu xí khi kịch bản chạy, vì vậy `Hide kịch bản` trong khi đang chạy được kiểm tra theo mặc định.

Kịch bản móc máy khách mẫu có thể được tìm thấy trong thư mục `contrib` trong *Kho TortoiseSVN* [<http://tortoisesvn.googlecode.com/svn/trunk/contrib/hook-scripts>] . ( [Phần 3](#), “[Giấy phép](#)” giải thích làm thế nào để truy cập vào kho lưu trữ.)

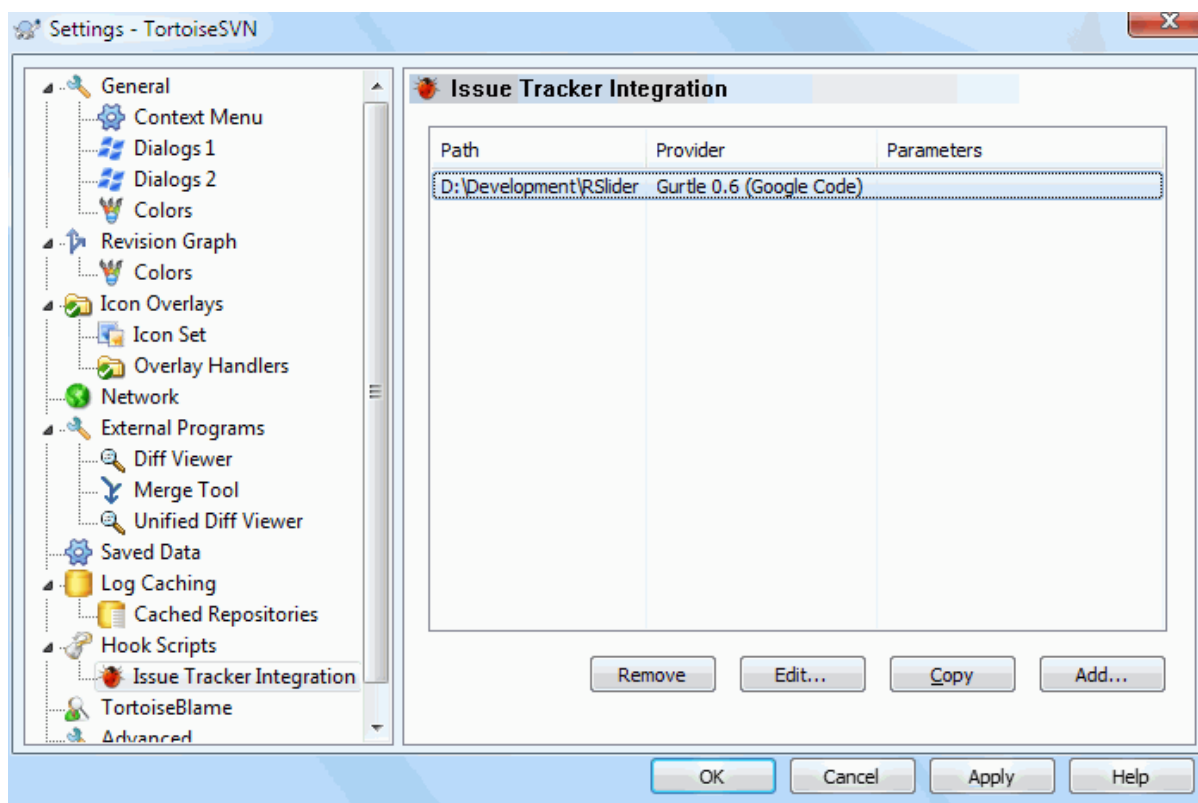
Khi gỡ lỗi các kịch bản móc bạn có thể muốn echo các dòng tiến triển vào màn hình điều khiển DOS, hoặc chèn lệnh dừng để ngăn cửa sổ điều khiển khỏi biến mất khi kịch bản hoàn tất. Bởi vì I/O được chuyển hướng việc này thường không làm được. Tuy nhiên bạn có thể chuyển hướng nhập và xuất một cách rõ ràng đến CON để khắc phục việc này. Ví dụ

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

Một công cụ nhỏ được bao gồm trong thư mục cài đặt TortoiseSVN được đặt tên `ConnectVPN.exe` . Bạn có thể sử dụng công cụ này được cấu hình như một móc kết nối tự động kết nối tới mạng của bạn trước khi TortoiseSVN cố gắng kết nối với một kho lưu trữ. Chỉ cần nạp tên của kết nối VPN như tham số đầu tiên vào công cụ này.

### 4.30.8.1. Tích hợp bộ theo dõi lỗi

TortoiseSVN có thể sử dụng một plugin COM để truy vấn theo dõi vấn đề khi đang ở trong hộp thoại cam kết. Việc sử dụng các plugin này được mô tả trong [Phần 4.28.2](#), “[Lấy Thông tin từ Bộ Theo Dõi Vấn Đề](#)” . Nếu người quản trị hệ thống của bạn đã cung cấp cho bạn một plugin mà bạn đã cài đặt và đăng ký, đây là nơi để xác định làm thế nào mà nó tích hợp với bản sao làm việc của bạn.



**Hình 4.82. Hộp Thoại Thiết Lập, Trang Tích Hợp Bộ Theo Dõi Vấn Đề**

Bấm vào **Bổ Sung ...** sử dụng các plugin với một bản sao làm việc cụ thể. Ở đây bạn có thể chỉ định đường dẫn bản sao làm việc, chọn plugin để sử dụng từ một danh sách thả xuống của tất cả các bộ theo dõi vấn đề được đăng ký, và bất kỳ thông số nào để chuyển vào. Các thông số sẽ là cụ thể đối với plugin, nhưng có thể bao gồm tên người dùng của bạn trên bộ theo dõi vấn đề để các plugin có thể truy vấn các vấn đề được giao cho bạn.

Nếu bạn muốn tất cả người dùng sử dụng cùng một plugin COM cho dự án của bạn, bạn có thể chỉ định các plugin cũng với các thuộc tính `bugtraq:provideruuid`, `bugtraq:provideruuid64` và `bugtraq:providerparams`.

#### `bugtraq:provideruuid`

Thuộc tính này xác định UUID COM của `IBugtraqProvider`, ví dụ `{91974081-2DC7-4FB1-B3BE-0DE1C8D6CE4E}`. (Ví dụ này là UUID của *Nhà cung cấp Gurtle bugtraq* [<http://code.google.com/p/gurtle/>], là một nhà cung cấp dịch vụ cho theo dõi vấn đề trên *Google Code* [<http://code.google.com/hosting/>]).

#### `bugtraq:provideruuid64`

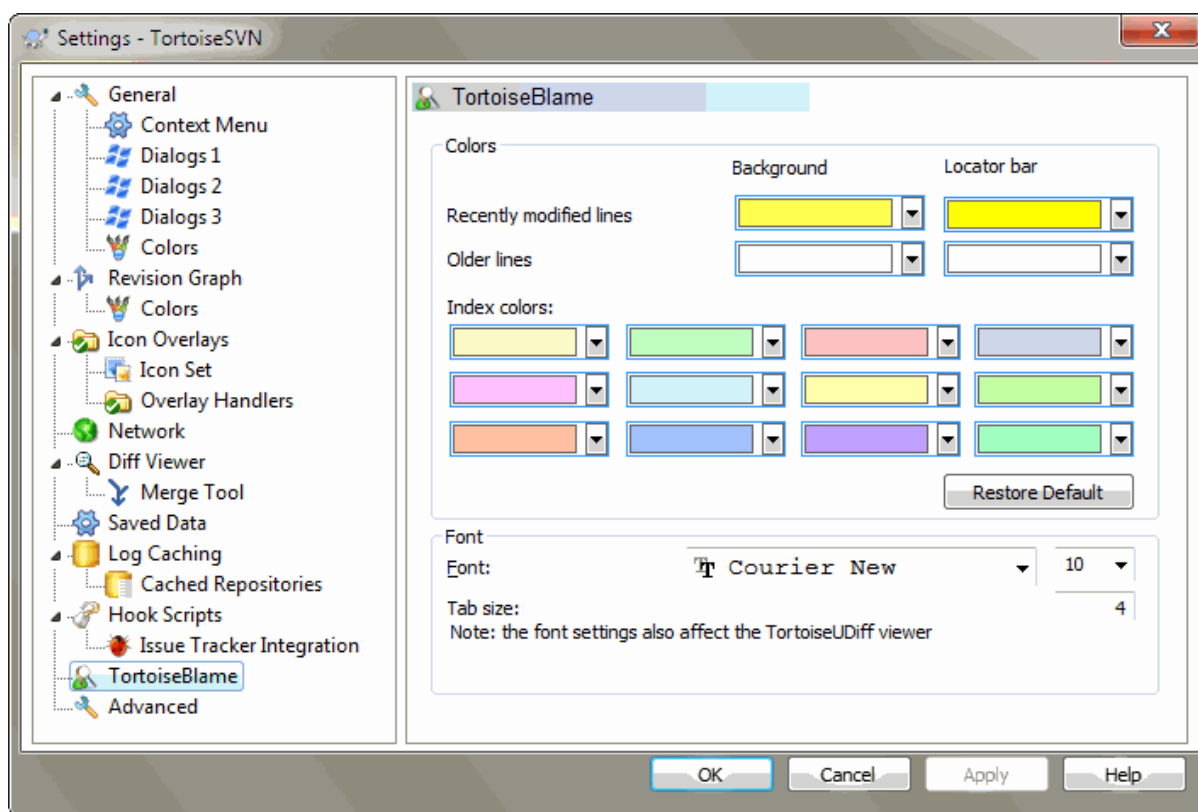
Đây cũng giống như `bugtraq:provideruuid`, nhưng là cho phiên bản 64-bit của `IBugtraqProvider`.

#### `bugtraq:providerparams`

Thuộc tính này quy định các thông số được chuyển đến `IBugtraqProvider`.

Xin vui lòng kiểm tra tài liệu của plugin `IBugtraqProvider` của bạn để tìm hiểu những gì phải chỉ định trong hai thuộc tính này.

## 4.30.9. Thiết Lập TortoiseBlame



**Hình 4.83. Hộp Thoại Thiết Lập, Trang TortoiseBlame**

Các thiết lập được sử dụng bởi TortoiseBlame được điều khiển từ trình đơn ngữ cảnh chính, chứ không phải trực tiếp với bản thân TortoiseBlame.

### Màu sắc

TortoiseBlame có thể sử dụng màu nền để chỉ ra tuổi của các dòng trong một tập tin. Bạn đặt các thiết bị đầu cuối bằng cách xác định màu sắc cho các phiên bản mới nhất và lâu đời nhất, và TortoiseBlame sử dụng một suy tuyến tính giữa các màu sắc tùy theo sửa đổi kho lưu trữ được ra dấu cho mỗi dòng.

Bạn có thể chỉ định các màu sắc khác nhau để sử dụng cho thanh định vị. Mặc định là sử dụng sự tương phản mạnh mẽ trên thanh định vị trong khi vẫn giữ màu nền sáng cho cửa sổ chính để bạn vẫn có thể đọc các văn bản.

### Phông Chữ

Bạn có thể chọn phông chữ được sử dụng để hiển thị văn bản, và kích thước điểm để sử dụng. Điều này áp dụng cho cả hai nội dung tập tin, và tác giả và thông tin sửa đổi được hiển thị trong khung bên trái.

### Các tab

Xác định bao nhiêu không gian để sử dụng cho việc mở rộng khi một ký tự tab được tìm thấy trong nội dung tập tin.

## 4.30.10. Thiết Lập Nâng Cao

Một vài thiết lập không thường xuyên được sử dụng chỉ có sẵn trong trang nâng cao của hộp thoại thiết lập. Các thiết lập này thay đổi registry trực tiếp và bạn cần phải biết những gì các thiết lập này được sử dụng cho và những gì nó làm. Không sửa đổi các thiết lập này, trừ khi bạn chắc chắn bạn cần phải thay đổi chúng.

### AllowAuthSave

Đôi khi nhiều người dùng sử dụng cùng một tài khoản trên cùng một máy tính. Trong tình huống như vậy, nó không thực sự được mong muốn để lưu các dữ liệu xác thực đăng nhập. Thiết lập giá trị này thành sai để vô hiệu hóa nút lưu xác thực đăng nhập trong hộp thoại xác thực.

#### AllowUnversionedObstruction

Nếu một cập nhật thêm vào một tập tin mới từ kho lưu trữ đã tồn tại trong bản sao làm việc địa phương như là một tập tin chưa phiên bản, hành động mặc định là giữ tập tin địa phương, cho thấy nó như là một phiên bản (có thể) được sửa đổi của tập tin mới từ kho. Nếu bạn muốn TortoiseSVN tạo ra một xung đột trong tình huống như vậy, thiết lập giá trị này thành `sai` .

#### AlwaysExtendedMenu

Cũng như với explorer, TortoiseSVN hiển thị các lệnh bổ sung nếu phím **Shift** được nhấn trong khi được mở ra trình đơn ngữ cảnh. Buộc TortoiseSVN luôn luôn hiển thị các lệnh mở rộng, thiết lập giá trị này thành `đúng` .

#### AutoCompleteMinChars

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

#### AutocompleteRemovesExtensions

Danh sách hoàn thành tự động được hiển thị trong trình soạn thảo thông điệp tường trình sẽ hiển thị tên của các tập tin được liệt kê cho cam kết. Để bao gồm các tên này với phần mở rộng được loại bỏ, thiết lập giá trị này thành `đúng` .

#### BlockStatus

Nếu bạn không muốn explorer cập nhật tình trạng các lớp phủ trong khi một lệnh TortoiseSVN khác đang chạy (ví dụ như Cập Nhật, Cam Kết, ...) thì đặt giá trị này thành `đúng` .

#### CacheTrayIcon

Để thêm một biểu tượng trên khay bộ nhớ cache cho chương trình TSVNCache, thiết lập giá trị này thành `đúng` . Điều này thực sự chỉ hữu ích cho các nhà phát triển vì nó cho phép bạn chấm dứt chương trình một cách uyển chuyển.

#### ColumnsEveryWhere

Các cột thêm TortoiseSVN thêm vào để xem chi tiết trong Windows Explorer thường chỉ hoạt động trong một bản sao làm việc. Nếu bạn muốn những cột đó có thể truy cập ở khắp mọi nơi, không chỉ ở bản sao làm việc, thiết lập giá trị này thành `đúng` . Lưu ý rằng các cột thêm chỉ có sẵn trong XP, Vista và mới hơn không hỗ trợ tính năng đó nữa.

#### ConfigDir

Bạn có thể chỉ định một vị trí khác cho các tập tin cấu hình Subversion. Điều này sẽ ảnh hưởng đến tất cả các tác vụ TortoiseSVN.

#### CtrlEnter

Trong hầu hết các hộp thoại trong TortoiseSVN, bạn có thể sử dụng **Ctrl+Enter** để bỏ qua hộp thoại như nếu bạn nhấp vào nút OK. Nếu bạn không muốn điều này, thiết lập giá trị này thành `sai` .

#### Debug

Đặt nó thành `đúng` nếu bạn muốn một hộp thoại bật lên cho mỗi lệnh hiển thị các dòng lệnh được sử dụng để bắt đầu TortoiseProc.exe.

#### DebugOutputString

Đặt cái này để `true` nếu bạn muốn TortoiseSVN in ra các thông điệp gỡ lỗi trong quá trình thực hiện. Các thông điệp có thể được thu hồi chỉ với các công cụ gỡ lỗi đặc biệt.

#### DialogTitles

Các định dạng mặc định (giá trị 0) tiêu đề hộp thoại `url/path - name of dialog - TortoiseSVN` . Nếu bạn đặt giá trị này là 1, định dạng thay đổi thành `tên của hộp thoại - url/path - TortoiseSVN` .

#### DiffBlamesWithTortoiseMerge

TortoiseSVN cho phép bạn chỉ định một trình xem khác biệt bên ngoài. Hầu hết các trình xem như vậy, tuy nhiên, không phù hợp cho đổ lỗi thay đổi ( [Phần 4.23.2](#), “**Đổ lỗi cho khác biệt**” ), Vì vậy bạn có thể quay trở lại TortoiseMerge trong trường hợp này. Để làm như vậy, thiết lập giá trị này thành `đúng` .



### FixCaseRenames

Một số ứng dụng thay đổi kiểu chữ hoa của tên tập tin mà không thông báo nhưng những thay đổi đó thật sự không cần thiết hoặc không mong muốn. Ví dụ một thay đổi từ `file.txt` to `FILE.TXT` sẽ không làm phiền các ứng dụng Windows, nhưng Subversion thì phân biệt chữ hoa trong những tình huống này. Vì vậy TortoiseSVN tự động sửa những thay đổi chữ hoa.

Nếu bạn không muốn TortoiseSVN tự động sửa những thay đổi chữ hoa cho bạn, bạn có thể đặt giá trị này thành `false`.

### FullRowSelect

Việc kiểm soát danh sách tình trạng được sử dụng trong các hộp thoại khác nhau (ví dụ: cam kết, kiểm tra-cho-sửa đổi, bổ sung, hoàn nguyên, ...) sử dụng lựa chọn cả hàng (ví dụ, nếu bạn chọn một mục, cả hàng đầy đủ được chọn, không chỉ cột đầu tiên). Điều này là tốt, nhưng hàng được lựa chọn sau đó cũng bao gồm các hình ảnh nên trên góc dưới bên phải, có thể trông xấu xí. Để vô hiệu hóa việc lựa chọn hàng đầy đủ, thiết lập giá trị này thành `sai`.

### GroupTaskbarIconsPerRepo

Tùy chọn này xác định cách mà các biểu tượng thanh tác vụ Win7 của các hộp thoại TortoiseSVN khác nhau và các cửa sổ được nhóm lại với nhau. Tùy chọn này không có tác dụng trên Windows XP hoặc Vista!

1. Giá trị mặc định là 0. Với thiết lập này, các biểu tượng được nhóm lại với nhau theo loại ứng dụng. Tất cả các hộp thoại từ TortoiseSVN được nhóm lại với nhau, tất cả các cửa sổ từ TortoiseMerge được nhóm lại với nhau, ...



**Hình 4.84. Thanh tác vụ với cách nhóm mặc định**

2. Nếu đặt thành 1, thì thay vì tất cả các hộp thoại trong một nhóm duy nhất cho mỗi ứng dụng, chúng được nhóm lại với nhau bởi kho lưu trữ. Ví dụ, nếu bạn có một hộp thoại tường trình và một hộp thoại cam kết mở cho kho A, và một hộp thoại kiểm tra-cho-thay đổi và một hộp thoại tường trình cho kho B, thì có hai nhóm biểu tượng ứng dụng được hiển thị trong thanh tác vụ Win7, một nhóm cho mỗi kho. Tuy nhiên, cửa sổ TortoiseMerge không được nhóm lại với nhau với các hộp thoại TortoiseSVN.



**Hình 4.85. Thanh tác vụ với cách nhóm kho**

3. Nếu đặt thành 2, thì việc lập nhóm hoạt động như với các thiết lập thành 1, ngoại trừ là các cửa sổ TortoiseSVN, TortoiseMerge, TortoiseBlame, TortoiseIDiff và TortoiseUDiff được nhóm lại với nhau. Ví dụ, nếu bạn có hộp thoại cam kết được mở và sau đó nhấp kép chuột vào một tập tin được sửa đổi, cửa sổ tìm khác biệt TortoiseMerge được mở sẽ được đặt trong cùng một nhóm biểu tượng trên thanh tác vụ như biểu tượng hộp thoại cam kết.



**Hình 4.86. Thanh tác vụ với cách nhóm kho**

4. Nếu thiết lập thành 3, thì việc lập nhóm hoạt động như với thiết lập với 1, nhưng việc lập nhóm không được thực hiện theo các kho lưu trữ nhưng theo bản sao làm việc. Điều này rất hữu ích nếu bạn có tất cả các dự án của bạn trong cùng một kho lưu trữ nhưng là các bản sao làm việc khác nhau cho từng dự án.
5. Nếu đặt bằng 4, thì việc lập nhóm hoạt động như với thiết lập bằng 2, nhưng việc lập nhóm không được thực hiện theo các kho lưu trữ mà là theo bản sao làm việc.

#### HideExternalInfo

Nếu được đặt thành `false`, thì mọi `svn:externals` sẽ được hiển thị riêng rẽ trong quá trình cập nhật.

Nếu được đặt thành `true` (the default), thì cập nhật thông tin cho các thành phần ngoài sẽ chỉ hiển thị nếu các thành phần ngoài bị ảnh hưởng bởi việc cập nhật, có nghĩa là bị thay đổi qua cách nào đó. Nếu không thì không có gì được hiển thị như đối với tập tin và thư mục.

#### GroupTaskbarIconsPerRepoOverlay

Điều này không có tác dụng nếu tùy chọn `GroupTaskbarIconsPerRepo` được thiết lập là 0 (xem ở trên).

Nếu tùy chọn này được thiết lập thành `đúng`, thì mỗi biểu tượng trên thanh tác vụ Win7 hiển thị một lớp phủ hình chữ nhật nhỏ có màu, cho thấy các kho lưu trữ mà các hộp thoại / cửa sổ được sử dụng cho.



**Hình 4.87. Nhóm thanh tác vụ với lớp phủ màu cho kho lưu trữ**

#### IncludeExternals

Theo mặc định, TortoiseSVN luôn luôn chạy một bản cập nhật với các thành phần bên ngoài được bao gồm. Điều này nhằm tránh các vấn đề với bản sao làm việc không phù hợp. Tuy nhiên, nếu bạn có rất nhiều thành phần bên ngoài được thiết lập, một bản cập nhật có thể mất nhiều thời gian. Đặt giá trị này thành `sai` để chạy các bản cập nhật mặc định với thành phần bên ngoài bị loại trừ. Để cập nhật với thành phần bên ngoài được bao gồm, hoặc là chạy hộp thoại `Cập nhật cho đến sửa đổi ...` hoặc đặt giá trị này thành `đúng` một lần nữa.

#### LogFindCopyFrom

Khi hộp thoại tường trình được bắt đầu từ trình hướng dẫn sáp nhập, sửa đổi đã được sáp nhập được hiển thị bằng màu xám, nhưng sửa đổi vượt ra ngoài điểm mà chi nhánh đã được tạo ra cũng được hiển thị. Những sửa đổi này được thể hiện trong màu đen bởi vì chúng không thể được sáp nhập.

Nếu tùy chọn này được thiết lập thành `true` thì TortoiseSVN sẽ cố gắng để tìm những sửa đổi nơi chi nhánh đã được tạo ra và ẩn tất cả các bản nằm ngoài bản sửa đổi đó. Bởi vì điều này có thể mất nhiều thời gian, tùy chọn này bị vô hiệu hóa theo mặc định. Ngoài ra tùy chọn này không làm việc với một số máy chủ SVN (ví dụ, Google Code Hosting, xem [vấn đề #5471](http://code.google.com/p/support/issues/detail?id=5471) [http://code.google.com/p/support/issues/detail?id=5471]).

#### LogStatusCheck

Hộp thoại tường trình cho thấy việc sửa đổi đường dẫn bản sao làm việc được in đậm. Nhưng điều này đòi hỏi rằng các hộp thoại tường trình lấy tình trạng của đường dẫn đó. Bởi vì bản sao làm việc rất lớn, điều này có thể mất một thời gian, bạn có thể thiết lập giá trị này thành `sai` để tắt tính năng này.

#### MergeLogSeparator

Khi bạn kết hợp sửa đổi từ chi nhánh khác, và thông tin theo dõi kết hợp được có sẵn, các thông điệp tường trình từ các phiên bản mà bạn hợp nhất sẽ được thu thập để làm một thông điệp tường trình cam kết. Một chuỗi được định nghĩa trước được sử dụng để tách các thông điệp tường trình cá nhân của các phiên bản bị sáp nhập. Nếu bạn thích, bạn có thể thiết lập nó thành giá trị có chứa một chuỗi phân cách tùy sự lựa chọn của bạn.

#### 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 kiểm tra xem liệu có một phiên bản mới có sẵn khoảng mỗi tuần một lần. Nếu có một phiên bản cập nhật được tìm thấy, hộp thoại cam kết sẽ hiển thị một điều khiển liên kết với thông tin đó. Nếu bạn thích các hành vi cũ trở về trước khi mà một hộp thoại hiện lên thông báo cho bạn về các cập nhật, thiết lập giá trị này thành `đúng` .

#### 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`.

#### ScintillaDirect2D

Tùy chọn này sẽ bật Direct2D tăng tốc vẽ trong thành phần điều khiển Scintilla được dùng trong hộp biên tập trong, ví dụ như hộp thoại cam kết, và cũng cho trình xem diff thống nhất. Tuy nhiên với vài thẻ đồ họa tùy chọn này thỉnh thoảng không làm việc đúng cho nên con trỏ để nhập văn bản không luôn luôn hiện. Nếu việc này xảy ra, bạn có thể tắt tính năng này bằng cách đặt giá trị này thành `false`.

#### OutOfDateRetry

Nếu bạn không muốn TortoiseSVN yêu cầu bạn cập nhật các bản sao hoạt động tự động sau một lỗi `Lỗi thời` , thiết lập giá trị này thành `false` .

#### ShellMenuAccelerators

TortoiseSVN sử dụng phím gia tốc cho các mục trình đơn ngữ cảnh của nó trong explorer. Bởi vì điều này có thể dẫn đến việc trùng lặp phím gia tốc (ví dụ như `SVN Cam kết` có phím tăng tốc `Alt-C` , nhưng mục `Sao chép` trong explorer cũng vậy). Nếu bạn không muốn hoặc cần các phím gia tốc của các mục TortoiseSVN, thiết lập giá trị này thành `sai` .

#### ShowContextMenuIcons

Điều này có thể hữu ích nếu bạn sử dụng một cái gì đó khác windows explorer hoặc nếu bạn gặp vấn đề với các trình đơn ngữ cảnh hiển thị không chính xác. Đặt giá trị này `sai` nếu bạn không muốn TortoiseSVN hiển thị các biểu tượng cho các mục trình đơn ngữ cảnh shell. Đặt giá trị này `đúng` để hiển thị lại các biểu tượng.

#### ShowAppContextMenuIcons

Nếu bạn không muốn TortoiseSVN hiển thị các biểu tượng cho các trình đơn ngữ cảnh trong các hộp thoại riêng của mình, thiết lập giá trị này `sai` .

#### StyleCommitMessages

Hộp thoại cam kết và tường trình sử dụng phong cách (ví dụ như in đậm, in nghiêng) trong các thông điệp cam kết (xem [Phần 4.4.5](#), “[Thông Điệp Tường Trình Cam Kết](#)” để biết thêm chi tiết). Nếu bạn không muốn làm điều này, thiết lập giá trị thành `sai` .

#### UpdateCheckURL

Giá trị này có chứa URL mà từ đó TortoiseSVN cố gắng để tải về một tập tin văn bản để tìm hiểu xem có bản cập nhật hay không. Điều này có thể có ích cho các quản trị viên công ty không muốn người dùng của họ cập nhật TortoiseSVN cho đến khi họ chấp thuận nó.

#### VersionCheck

TortoiseSVN kiểm tra xem liệu có một phiên bản mới có sẵn khoảng mỗi tuần một lần. Nếu bạn không muốn TortoiseSVN làm việc kiểm tra, thiết lập giá trị này thành `sai`.

### 4.30.11. Các Thiết Lập Xuất Khẩu TSVN

Nếu bạn muốn xuất khẩu tất cả các thiết lập máy khách của bạn để sử dụng trên một máy tính khác, bạn có thể làm như vậy bằng cách sử dụng trình biên tập Windows registry `regedt32.exe`. Đi tới khóa registry `HKCU\Software\TortoiseSVN` và xuất khẩu nó vào một tập tin reg. Trên các máy tính khác, chỉ cần nhập khẩu tập tin đó một lần nữa (thông thường, kích đúp vào tập tin reg sẽ làm điều đó).

Luôn nhớ lưu các thiết lập thông thường của Subversion, mà bạn có thể tìm trong tập tin cấu hình Subversion `%APPDATA%\Subversion\config`.

## 4.31. Bước Cuối cùng

### Đóng Góp!

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.

Trong khi làm việc trên TortoiseSVN, chúng tôi thích nghe nhạc. Và bởi vì chúng ta dành nhiều thời gian trên dự án chúng ta cần *rất nhiều* âm nhạc. Vì vậy chúng tôi đã thiết lập một sổ muốn danh sách mong muốn với đĩa CD nhạc và DVD yêu thích của chúng tôi: <http://tortoisesvn.net/donate.html> Cũng vui lòng nhìn vào danh sách những người đã đóng góp cho dự án bằng cách gửi các bản vá lỗi hoặc bản dịch.

# Chương 5. Chương trình SubWCRev

SubWCRev là một chương trình giao diện điều khiển Windows có thể được sử dụng để đọc trạng thái của một bản sao làm việc Subversion và được tùy chọn thực hiện thay thế từ khóa trong một tập tin mẫu. Điều này thường được sử dụng như là một phần của tiến trình xây dựng như một phương tiện kết hợp chặt chẽ với những thông tin bản sao hoạt động vào đối tượng bạn đang xây dựng. Thông thường nó có thể được sử dụng để bao gồm các số sửa đổi trong một hộp “Nói Về”.

## 5.1. Dòng Lệnh SubWCRev

SubWCRev đọc tình trạng Subversion của tất cả các tập tin trong một bản sao làm việc, không bao gồm thành phần bên ngoài theo mặc định. Nó ghi lại số phiên bản cam kết cao nhất được tìm thấy, và các dấu thời gian cam kết sửa đổi đó, nó cũng ghi lại xem có thay đổi địa phương trong các bản sao làm việc, hoặc các phiên bản cập nhật hỗn hợp. Số sửa đổi, cập nhật, phạm vi sửa đổi và tình trạng thay đổi được hiển thị trên thiết bị xuất chuẩn.

SubWCRev.exe được gọi từ dòng lệnh hoặc một kịch bản, và được điều khiển bằng cách sử dụng các thông số dòng lệnh.

```
SubWCRev WorkingCopyPath [SrcVersionFile DstVersionFile] [-nmdfe]
```

`WorkingCopyPath` là đường dẫn đến bản sao làm việc được kiểm tra. Bạn chỉ có thể sử dụng SubWCRev trên bản sao làm việc, không trực tiếp trên kho lưu trữ. Đường dẫn có thể tuyệt đối hay tương đối so với thư mục làm việc hiện tại.

Nếu bạn muốn SubWCRev thực hiện thay thế từ khóa, ẽ các trường như sửa đổi kho lưu trữ và URL được lưu vào một tập tin văn bản, bạn cần phải cung cấp một tập tin mẫu `SrcVersionFile` và một tập tin đầu ra `DstVersionFile` trong đó có các phiên bản thay thế của mẫu.

Có một số tùy chọn chuyển mạch mà ảnh hưởng đến cách SubWCRev hoạt động. Nếu bạn sử dụng nhiều hơn một, chúng phải được quy định cụ thể như một nhóm duy nhất, ví dụ như `-nm`, Chứ không phải `-n-m`.

Chuyển	Mô tả
<code>-n</code>	Nếu chuyển đổi này được đưa ra, SubWCRev sẽ thoát ra với <code>ERRORLEVEL 7</code> nếu bản sao làm việc có thay đổi địa phương. Điều này có thể được sử dụng để ngăn chặn xây dựng với sự có mặt của những thay đổi không được cam kết.

### Bảng 5.1. Danh sách các thiết bị chuyển mạch dòng lệnh có sẵn

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	Mô tả
1	Syntax error. One or more command line parameters are invalid.

### Bảng 5.2. List of SubWCRev error codes

## 5.2. Thay Thế Từ Khoá

Nếu một tập tin nguồn và đích được cung cấp, SubWCRev sao chép từ nguồn đến đích, thực hiện thay thế từ khóa như sau:

Từ khoá	Mô tả
<code>\$WCREV\$</code>	Thay thế với sửa đổi cam kết cao nhất trong bản sao làm việc.

### Bảng 5.3. Danh sách các từ khoá có sẵn

SubWCRev không trực tiếp hỗ trợ lồng ghép các biểu thức, ví dụ bạn không thể dùng một biểu thức như:

```
#define SVN_REVISION "$WCMIXED?$WCRANGE$: $WCREV$$"
```

Nhưng bạn thường dùng cách khác, ví dụ:

```
#define SVN_RANGE $WCRANGE$
#define SVN_REV $WCREV$
#define SVN_REVISION "$WCMIXED?SVN_RANGE:SVN_REV$"
```



## Mẹo

Một số các từ khóa này áp dụng đối với các tập tin đơn lẻ chứ không phải là toàn bộ một bản sao làm việc, vì vậy nó chỉ có nghĩa để sử dụng những cái này khi SubWCRev được gọi là để quét một tập tin đơn lẻ. Điều này áp dụng đối với \$WCINSVN\$, \$WCNEEDSLOCK\$, \$WCISLOCKED\$, \$WCLOCKDATE\$, \$WCLOCKOWNER\$ và \$WCLOCKCOMMENT\$ .

## 5.3. Ví dụ Từ khoá

Ví dụ dưới đây cho thấy làm thế nào các từ khoá trong một tập tin mẫu được thay thế trong các tập tin đầu ra.

```
// 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
```

Sau khi chạy SubWCRev.exe đường dẫn \ tới \ bản sao làm việc testfile.tmpl testfile.txt , tập tin đầu ra testfile.txt sẽ trông như thế này:

```
// Test file for SubWCRev

char *Revision      = "22837";
char *Revision16    = "53";
char *Revisionp100  = "22937";
char *Revisionm100  = "22737";
char *Modified       = "Modified";
char *Unversioned   = "no unversioned items";
char *Date           = "2012/04/26 18:47:57";
char *CustDate       = "Thu, 26 April 2012";
char *DateUTC        = "2012/04/26 16:47:57";
char *CustDateUTC    = "Thu, 26 April 2012";
char *TimeNow        = "2012/04/26 20:51:17";
char *TimeNowUTC     = "2012/04/26 18:51:17";
char *RevRange       = "22836:22837";
char *Mixed          = "Mixed revision WC";
char *ExtAllFixed    = "All externals fixed";
char *IsTagged       = "Not tagged";
char *URL            = "https://tortoisesvn.googlecode.com/svn/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
```



### Mẹo

Một tập tin như thế này sẽ được bao gồm trong bản xây dựng, do đó bạn sẽ mong đợi nó được phiên bản. Hãy chắc chắn tạo phiên bản tập tin mẫu, không phải là tập tin được tạo ra, nếu không mỗi khi bạn tạo lại các tập tin phiên bản bạn cần phải cam kết thay đổi, do đó có nghĩa là các tập tin phiên bản cần được cập nhật.

## 5.4. giao diện COM

Nếu bạn cần truy cập thông tin sửa đổi Subversion từ các chương trình khác, bạn có thể sử dụng giao diện COM SubWCRev. Các đối tượng để tạo ra là `SubWCRev.object` , và các phương pháp sau đây được hỗ trợ:

Phương pháp	Mô tả
<code>.GetWCInfo</code>	Phương pháp này duyệt qua các bản sao hoạt động thu thập thông tin sửa đổi. Đương nhiên bạn phải gọi việc này trước khi bạn có thể truy cập thông tin bằng cách sử dụng các phương pháp còn lại. Tham số đầu tiên là đường dẫn. Tham số thứ hai nên là true nếu bạn muốn bao gồm các phiên bản thư mục.

Phương pháp	Mô tả
	Tương đương với -f chuyển đổi dòng lệnh. Tham số thứ ba nên là true nếu bạn muốn bao gồm svn:externals. Tương đương với -e chuyển đổi dòng lệnh.

### Bảng 5.4. các phương pháp COM / tự động hóa được hỗ trợ

Ví dụ sau đây cho thấy giao diện có thể được sử dụng như thế nào.

```
// 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);
```

Các danh sách sau đây là một ví dụ về cách sử dụng các đối tượng COM SubWCRev từ 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");
}
```

---

# Chương 6. giao diện IBugtraqProvider

Để có được tích hợp chặt chẽ với bộ theo dõi vấn đề hơn đơn giản bằng cách sử dụng thuộc tính `bugtraq:`, TortoiseSVN có thể sử dụng các plugin COM. Với các plugin này, có thể lấy thông tin trực tiếp từ bộ theo dõi vấn đề, tương tác với người sử dụng và cung cấp thông tin trở lại TortoiseSVN về các vấn đề còn mở, kiểm tra các thông điệp tường trình được nhập vào bởi người sử dụng và thậm chí chạy các hành động sau khi một cam kết thành công để ví dụ như, đóng một vấn đề.

Chúng tôi không thể cung cấp thông tin và hướng dẫn về làm thế nào bạn có thể thực hiện một đối tượng COM trong ngôn ngữ lập trình ưa thích của bạn, nhưng chúng tôi đã bổ sung ví dụ trong C++ / ATL và C # trong kho của chúng tôi trong thư mục `contrib / issue-tracker-plugins`. Trong thư mục đó, bạn cũng có thể tìm thấy các yêu cầu bao gồm các tập tin bạn cần để xây dựng plugin của bạn. ( [Phần 3](#), “[Giấy phép](#)” giải thích làm thế nào để truy cập vào kho lưu trữ.)



## Quan trọng

Bạn nên cung cấp cả hai phiên bản 32-bit và 64-bit của plugin của bạn. Bởi vì phiên bản x64 của TortoiseSVN không thể sử dụng một plugin 32-bit và ngược lại.

### 6.1. Quy ước đặt tên

Nếu bạn phát hành một plugin theo dõi vấn đề cho TortoiseSVN, xin vui lòng *không* đặt tên là *Tortoise<Something>*. Chúng tôi muốn để dành tiền tố *Tortoise* cho một trình khách kiểm soát phiên bản tích hợp vào windows shell. Ví dụ: TortoiseCVS, TortoiseSVN, TortoiseHg, TortoiseGit và TortoiseBzr là tất cả các trình máy khách kiểm soát phiên bản.

Xin vui lòng đặt tên plugin của bạn cho một trình khách Tortoise là *Turtle<Something>*, nơi *<Something>* đề cập đến bộ theo dõi vấn đề mà bạn đang kết nối với. Hoặc chọn một tên nghe giống như *Turtle* nhưng có một chữ đầu khác nhau. Các ví dụ tốt là:

- Gurtle - Một plugin theo dõi vấn đề cho mã của Google
- TurtleMine - Một plugin theo dõi vấn đề cho Redmine
- VurtleOne - Một plugin theo dõi vấn đề cho VersionOne

### 6.2. Giao diện IBugtraqProvider

TortoiseSVN 1,5 và sau đó có thể sử dụng các plugin mà thực hiện giao diện IBugtraqProvider. Giao diện này cung cấp một vài phương pháp bổ sung có thể sử dụng để tương tác với bộ theo dõi vấn đề.

```
HRESULT ValidateParameters (  
    // Cửa sổ mẹ cho bất kỳ Giao Diện nào cần được  
    // trung bày trong lúc xác minh.  
    [in] HWND hParentWnd,  
  
    // Chuỗi tham số cần được xác minh  
    [in] BSTR parameters,  
  
    // Chuỗi có hợp lệ không?  
    [out, retval] VARIANT_BOOL *valid  
);
```

Phương pháp này được gọi là từ hộp thoại thiết lập mà người dùng có thể thêm vào và cấu hình các plugin. Các chuỗi thông số có thể được sử dụng bởi một plugin để có được thông tin bổ sung cần thiết, ví dụ, URL để theo dõi vấn đề, thông tin đăng nhập, vv Các plugin cần phải xác minh các chuỗi thông số và hiển

thị một hộp thoại báo lỗi nếu chuỗi không hợp lệ. Các tham số `hParentWnd` được sử dụng cho bất kỳ hộp thoại plugin được hiển thị như là cửa sổ cha mẹ. Các plugin phải trả về TRUE nếu việc xác nhận các thông số chuỗi thành công. Nếu plugin trả về FALSE, hộp thoại thiết lập sẽ không cho phép người sử dụng thêm các plugin vào một đường dẫn bản sao làm việc.

```
HRESULT GetLinkText (
    // Cửa sổ mẹ cho bất kỳ Giao Diện (lỗi) cần được trưng bày.
    [in] HWND hParentWnd,

    // Chuỗi tham số, phòng trường hợp bạn cần nói chuyện với
    // web service (ví dụ) để tìm hiểu văn bản đúng là gì.
    [in] BSTR parameters,

    // Bạn muốn trưng bày văn bản nào?
    // Sử dụng văn hóa địa phương thread hiện tại.
    [out, retval] BSTR *linkText
);
```

Các plugin có thể cung cấp một chuỗi ở đây mà được sử dụng trong hộp thoại cam kết TortoiseSVN cho nút gọi các plugin, ví dụ như, "Chọn vấn đề" hoặc "Chọn vé". Hãy chắc chắn rằng chuỗi không phải là quá dài, nếu không nó có thể không vừa với nút. Nếu phương thức trả về một lỗi (ví dụ như, `E_NOTIMPL`), một văn bản mặc định sẽ được sử dụng cho nút.

```
HRESULT GetCommitMessage (
    // Cửa sổ mẹ cho Giao Diện của nhà cung cấp của bạn.
    [in] HWND hParentWnd,

    // Các tham số cho nhà cung cấp của bạn.
    [in] BSTR parameters,
    [in] BSTR commonRoot,
    [in] SAFEARRAY(BSTR) pathList,

    // Văn bản đã có mặt trong thông điệp cam kết.
    // Nhà cung cấp của bạn nên bao gồm văn bản này trong thông điệp mới,
    // ở nơi phù hợp.
    [in] BSTR originalMessage,

    // Văn bản mới cho thông điệp cam kết.
    // Cái này sẽ thay thế thông điệp gốc.
    [out, retval] BSTR *newMessage
);
```

Đây là phương pháp chính của các plugin. Phương pháp này được gọi là từ hộp thoại TortoiseSVN cam kết khi người dùng nhấp vào nút plugin.

Các chuỗi `thông số` là một chuỗi người dùng phải nhập vào trong hộp thoại thiết lập khi cấu hình plugin. Thông thường, một plugin sẽ sử dụng cái này để tìm địa chỉ URL của bộ theo dõi vấn đề và / hoặc thông tin đăng nhập hoặc nhiều hơn.

Các chuỗi `commonRoot` chứa đường dẫn cha mẹ của tất cả các mục đã chọn để đưa lên hộp thoại cam kết. Lưu ý rằng đây *không* phải là đường dẫn thư mục gốc của tất cả các mục mà người dùng đã lựa chọn trong hộp thoại cam kết. Đối với hộp thoại chi nhánh/thẻ, đây là đường dẫn mà sẽ được sao chép.

Tham số `pathList` có chứa một loạt các đường dẫn (dưới dạng các chuỗi) mà người dùng đã lựa chọn cho cam kết.

Các tham số `originalMessage` có chứa các văn bản nhập vào hộp thông điệp tương trình trong hộp thoại cam kết. Nếu người dùng chưa nhập vào bất kỳ văn bản nào, chuỗi này sẽ trống rỗng.

Các `newMessage` chuỗi trả về được sao chép vào hộp chỉnh sửa thông điệp tương trình trong hộp thoại cam kết, thay thế bất cứ thứ gì đã có. Nếu một plugin không thay đổi chuỗi `originalMessage`, nó phải trả lại cùng một chuỗi một lần nữa ở đây, nếu không sẽ bị mất bất kỳ văn bản người dùng đã nhập vào.

### 6.3. Giao diện IBugtraqProvider2

Trong TortoiseSVN 1.6 giao diện mới đã được bổ sung cung cấp nhiều chức năng hơn cho các plugins. Giao diện IBugtraqProvider2 này thừa hưởng từ IBugtraqProvider.

```
HRESULT GetCommitMessage2 (  
    // Parent window for your provider's UI.  
    [in] HWND hWndParentWnd,  
  
    // Parameters for your provider.  
    [in] BSTR parameters,  
    // The common URL of the commit  
    [in] BSTR commonURL,  
    [in] BSTR commonRoot,  
    [in] SAFEARRAY(BSTR) pathList,  
  
    // The text already present in the commit message.  
    // Your provider should include this text in the new message,  
    // where appropriate.  
    [in] BSTR originalMessage,  
  
    // You can assign custom revision properties to a commit  
    // by setting the next two params.  
    // note: Both safearrays must be of the same length.  
    //      For every property name there must be a property value!  
  
    // The content of the bugID field (if shown)  
    [in] BSTR bugID,  
  
    // Modified content of the bugID field  
    [out] BSTR * bugIDOut,  
  
    // The list of revision property names.  
    [out] SAFEARRAY(BSTR) * revPropNames,  
  
    // The list of revision property values.  
    [out] SAFEARRAY(BSTR) * revPropValues,  
  
    // The new text for the commit message.  
    // This replaces the original message  
    [out, retval] BSTR * newMessage  
);
```

Phương pháp này được gọi là từ hộp thoại cam kết TortoiseSVN khi người dùng nhấp vào nút plugin. Phương pháp này được gọi thay vì `GetCommitMessage()`. Xin vui lòng tham khảo các tài liệu cho `GetCommitMessage` để biết các thông số cũng được sử dụng ở đó.

Tham số `commonURL` là URL cha mẹ của tất cả các mục được lựa chọn để đưa lên hộp thoại cam kết. Về cơ bản đây là URL của đường dẫn `commonRoot`.

Tham số `bugID` chứa nội dung của trường ID lỗi (nếu nó được hiển thị, cấu hình với thuộc tính `bugtraq:message`).

Tham số trả về `bugIDOut` được sử dụng để điền vào các trường ID lỗi khi phương pháp quay về.

Các thông số trả về `revPropNames` và `revPropValues` có thể chứa các cặp tên / giá trị cho các thuộc tính mà cam kết nên thiết lập. Một plugin phải đảm bảo rằng cả hai mảng có cùng kích thước lúc trả về! Mỗi tên thuộc tính trong `revPropNames` cũng phải có một giá trị tương ứng trong `revPropValues`. Nếu không có thuộc tính sửa đổi nào được thiết lập, các plugin phải trả lại mảng trống.

```
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
);
```

Phương pháp này được gọi là ngay trước khi hộp thoại cam kết được đóng và cam kết bắt đầu. Một plugin có thể sử dụng phương pháp này để xác nhận các tập tin / thư mục được lựa chọn cho các cam kết và / hoặc thông điệp cam kết được nhập vào bởi người sử dụng. Các thông số đều giống nhau như đối với `GetCommitMessage2()`, với sự khác biệt mà `commonURL` URL chung của tất cả các mục được *kiểm tra*, và `commonRoot` đường dẫn thư mục gốc của tất cả các mục được kiểm tra.

Đối với hộp thoại chi nhánh / thẻ, `commonURL` URL nguồn của bản sao chép, và `commonRoot` được đặt đến các URL mục tiêu của bản sao chép.

Tham số trả về `errorMessage` có phải có một thông báo lỗi mà TortoiseSVN cho người sử dụng thấy hoặc được để trống cho cam kết để bắt đầu. Nếu một thông báo lỗi được trả về, TortoiseSVN hiển thị chuỗi lỗi trong một hộp thoại và giữ hộp thoại cam kết mở để người dùng có thể sửa chữa bất cứ điều gì bị sai. Một plugin do đó cần trả về một chuỗi lỗi thông báo cho người sử dụng *những gì* là sai và làm thế nào để sửa nó.

```
HRESULT OnCommitFinished (
    // Parent window for any (error) UI that needs to be displayed.
    [in] HWND hParentWnd,

    // The common root of all paths that got committed.
    [in] BSTR commonRoot,

    // All the paths that got committed.
    [in] SAFEARRAY(BSTR) pathList,

    // The text already present in the commit message.
    [in] BSTR logMessage,

    // The revision of the commit.
    [in] ULONG revision,

    // An error to show to the user if this function
    // returns something else than S_OK
    [out, retval] BSTR * error
);
```

Phương pháp này được gọi sau khi một cam kết thành công. Một plugin có thể sử dụng phương pháp này ví dụ như, đóng vấn đề được lựa chọn hoặc thêm thông tin về cam kết vào vấn đề này. Các thông số đều giống nhau như đối với `GetCommitMessage2`.

```
HRESULT HasOptions(  
    // Whether the provider provides options  
    [out, retval] VARIANT_BOOL *ret  
);
```

Phương pháp này được gọi là từ hộp thoại thiết lập mà người dùng có thể cấu hình các plugin. Nếu một plugin cung cấp hộp thoại cấu hình riêng của mình với `ShowOptionsDialog`, nó phải trả về TRUE ở đây, nếu không nó phải trả về FALSE.

```
HRESULT ShowOptionsDialog(  
    // Parent window for the options dialog  
    [in] HWND hParentWnd,  
  
    // Parameters for your provider.  
    [in] BSTR parameters,  
  
    // The parameters string  
    [out, retval] BSTR * newparameters  
);
```

Phương pháp này được gọi là từ hộp thoại thiết lập khi người dùng nhấp vào nút "Tùy chọn" được hiển thị nếu `HasOptions` trả về TRUE. Một plugin có thể hiển thị một hộp thoại tùy chọn để làm cho nó dễ dàng hơn cho người sử dụng để cấu hình các plugin.

Các chuỗi thông số chứa các chuỗi thông số của plugin đã được thiết lập / nhập vào.

Thông số trả về `newparameters` phải chứa các chuỗi thông số mà các plugin xây dựng từ thông tin thu thập trong hộp thoại tùy chọn của nó. Chuỗi `paramameters` đó được chuyển tới tất cả các `IBugtraqProvider` khác và các phương pháp `IBugtraqProvider2`.

---

# Phụ lục A. Các Câu Hỏi Thường Gặp (FAQ)

Bởi vì TortoiseSVN đang được phát triển mọi lúc nên đôi khi khó để giữ cho các tài liệu được cập nhật hoàn toàn. Chúng tôi duy trì một [Hỏi Đáp trực tuyến](http://tortoisesvn.net/faq.html) [http://tortoisesvn.net/faq.html] trong đó có chứa những câu hỏi được chọn lọc mà chúng tôi được hỏi nhiều nhất trên danh sách gửi thư TortoiseSVN < dev@tortoisesvn.tigris.org > và < users@tortoisesvn.tigris.org >.

Chúng tôi cũng duy trì một dự án [Vấn đề Tracker](http://code.google.com/p/tortoisesvn/wiki/IssueTracker?tm=3) [http://code.google.com/p/tortoisesvn/wiki/IssueTracker?tm=3] mà cho bạn biết về một số trong những điều chúng ta có trong danh sách To-Do, và các lỗi đã được sửa. Nếu bạn nghĩ rằng bạn đã tìm thấy một lỗi, hoặc muốn yêu cầu một tính năng mới, hãy kiểm tra ở đây trước xem có thấy một người nào khác đã đến đó trước bạn.

Nếu bạn có một câu hỏi mà không trả lời bất cứ nơi nào khác, nơi tốt nhất để yêu cầu nó là một trong những danh sách gửi thư:

- < users@tortoisesvn.tigris.org > là một cái để sử dụng nếu bạn có câu hỏi về sử dụng TortoiseSVN.
- Nếu bạn muốn giúp đỡ phát triển TortoiseSVN, thì bạn nên tham gia vào các cuộc thảo luận trên < dev@tortoisesvn.tigris.org >.
- Nếu bạn muốn giúp đỡ với bản dịch của giao diện người dùng TortoiseSVN hoặc tài liệu hướng dẫn, gửi e-mail < translators@tortoisesvn.tigris.org >.

---

# Phụ lục B. Làm thế nào để tôi. ..

Phụ lục này chứa các giải pháp cho các vấn đề / câu hỏi mà bạn có thể có khi sử dụng TortoiseSVN.

## B.1. Di chuyển / sao chép nhiều tập tin cùng một lúc

Di chuyển / sao chép các tập tin đơn lẻ có thể được thực hiện bằng cách sử dụng TortoiseSVN → **Đổi tên** ... . Nhưng nếu bạn muốn di chuyển / sao chép nhiều tập tin, cách này là quá chậm và tốn quá nhiều công sức.

Cách đề nghị là kéo phải các tập tin đến vị trí mới. Đơn giản chỉ cần nhấp chuột phải trên các tập tin mà bạn muốn di chuyển / sao chép mà không phải thả nút chuột. Sau đó kéo các tập tin đến vị trí mới và thả chuột. Một trình đơn ngữ cảnh sẽ xuất hiện nơi bạn có thể chọn **Trình Đơn Ngữ Cảnh** → **SVN Sao chép tập tin được phiên bản ở đây** . hoặc **Trình Đơn Ngữ Cảnh** → **SVN Di chuyển các tập tin được phiên bản ở đây** .

## B.2. Buộc người dùng phải nhập một thông điệp tường trình

Có hai cách để ngăn chặn người dùng khỏi cam kết với một thông điệp tường trình trống rỗng. Một cách là cụ thể với TortoiseSVN, còn cách kia làm việc được với tất cả các máy khách Subversion, nhưng đòi hỏi phải truy cập trực tiếp vào máy chủ.

### B.2.1. Kích bản Hook trên máy chủ

Nếu bạn có quyền truy cập trực tiếp đến máy chủ kho lưu trữ, bạn có thể cài đặt một kịch bản móc tiền-cam kết mà từ chối tất cả các cam kết với một thông điệp tường trình trống rỗng hoặc quá ngắn.

Trong thư mục kho lưu trữ trên máy chủ, có một thư mục con `hooks` trong đó có một số kịch bản móc ví dụ bạn có thể sử dụng. Tập tin `pre-commit.tmpl` chứa một kịch bản mẫu để từ chối cam kết nếu không có thông điệp tường trình được cung cấp, hoặc tin thông điệp là quá ngắn. Các tập tin cũng chứa bình luận về việc làm thế nào để cài đặt / sử dụng kịch bản này. Chỉ cần làm theo các hướng dẫn trong tập tin đó.

Phương pháp này là cách được khuyến khích dùng nếu người dùng của bạn cũng sử dụng máy khách Subversion khác hơn so với TortoiseSVN. Nhược điểm là cam kết bị từ chối bởi máy chủ và do đó người dùng sẽ nhận được một thông báo lỗi. Các máy khách không thể biết trước khi cam kết rằng nó sẽ bị từ chối. Nếu bạn muốn làm cho TortoiseSVN có nút **OK** bị vô hiệu hóa cho đến khi thông điệp tường trình là đủ dài thì xin vui lòng sử dụng phương pháp mô tả dưới đây.

### B.2.2. Các thuộc tính dự án

TortoiseSVN sử dụng các thuộc tính để kiểm soát một số tính năng của nó. Một trong những thuộc tính là thuộc tính `tsvn:logminsize` .

Nếu bạn thiết lập thuộc tính trên một thư mục, thì TortoiseSVN sẽ vô hiệu hóa nút **OK** trong tất cả các hộp thoại cam kết cho đến khi người sử dụng đã nhập vào một thông điệp tường trình với chiều dài ít nhất được quy định trong thuộc tính.

Để biết thêm thông tin chi tiết về những thuộc tính dự án, xin vui lòng tham khảo [Phần 4.17, “Thiết lập Dự án”](#) .

## B.3. Cập nhật các tập tin được lựa chọn từ kho

Thông thường bạn cập nhật bản sao của bạn làm việc bằng cách sử dụng TortoiseSVN → **Cập nhật** . Nhưng nếu bạn chỉ muốn chọn một số tập tin mới mà một đồng nghiệp đã thêm vào mà không cần sáp nhập bất kỳ thay đổi trên các tập tin khác tại cùng một thời gian, bạn cần một cách tiếp cận khác nhau.

Sử dụng TortoiseSVN → **Kiểm tra cho sửa đổi** . và bấm vào **Kiểm tra kho lưu trữ** để xem những gì đã thay đổi trong kho. Chọn các tập tin bạn muốn cập nhật tại địa phương, sau đó sử dụng trình đơn ngữ cảnh để cập nhật các tập tin.



## B.4. Phục hồi (Hoàn tác) các sửa đổi trong kho

### B.4.1. Sử dụng hộp thoại tường trình sửa đổi

Là cách dễ dàng nhất để hoàn nguyên các thay đổi từ một hoặc nhiều bản chỉnh sửa, là để dùng hộp thoại tường trình chỉnh sửa.

1. Chọn tập tin hoặc thư mục mà bạn cần để phục hồi các thay đổi. Nếu bạn muốn hoàn tác tất cả thay đổi, đây sẽ là thư mục cấp cao nhất.
2. Chọn TortoiseSVN → Hiện Tường Trình để hiển thị một danh sách các phiên bản. Bạn có thể cần phải sử dụng Hiện tất cả hoặc 100 Tiếp theo để hiển thị các sửa đổi bạn đang quan tâm.
3. Chọn bản chỉnh sửa mà bạn muốn hoàn nguyên. Nếu bạn muốn hoàn tác một dãy các bản chỉnh sửa, chọn bản đầu tiên và giữ phím **Shift** trong khi chọn bản cuối. Nếu bạn muốn chọn các bản chỉnh sửa riêng lẻ và theo dãy, dùng phím **Ctrl** khi chọn các bản chỉnh sửa. Nhấp phải trên các bản được chọn, sau đó chọn Trình Đơn Ngữ Cảnh → Hoàn nguyên các thay đổi từ bản chỉnh sửa này.
4. Hoặc nếu bạn muốn thực hiện một bản sửa đổi trước đó thành bản CHÍNH mới, nhấp chuột phải vào bản chỉnh sửa được lựa chọn, sau đó chọn Trình Đơn Ngữ Cảnh → Hoàn nguyên về phiên bản này. Điều này sẽ loại bỏ *tất cả* thay đổi sau khi sửa đổi được lựa chọn.

Bạn đã hoàn nguyên các thay đổi trong bản sao làm việc của bạn. Kiểm tra kết quả, sau đó cam kết những thay đổi.

### B.4.2. Sử dụng hộp thoại hợp nhất

Nếu bạn muốn nhập vào các số chỉnh sửa như là một danh sách, bạn có thể dùng hộp thoại Hợp Nhất. Phương pháp trước sử dụng hợp nhất ngầm ẩn; phương pháp này sử dụng nó một cách rõ ràng.

1. Trong bản sao làm việc của bạn chọn TortoiseSVN → Hợp Nhất.
2. Trong hộp thoại Kiểu Hợp Nhất chọn Hợp nhất một dãy các bản chỉnh sửa.
3. Trong trường Từ: nhập vào URL đầy đủ tới kho của thư mục bản làm việc của bạn. Nó nên được dùng làm URL mặc định.
4. Trong trường Dãy chỉnh sửa để hợp nhất nhập vào danh sách các chỉnh sửa để quay lại (hoặc dùng hộp thoại tường trình để chọn chúng như đã mô tả phía trên).
5. Hãy chắc chắn rằng hộp kiểm Quay lui hợp nhất được chọn.
6. Trong hộp thoại Các tùy chọn cho hợp nhất chấp nhận các giá trị mặc định.
7. Nhấn chuột vào Hợp Nhất để hoàn tất hợp nhất.

Bạn đã hoàn nguyên các thay đổi trong bản làm việc của bạn. Kiểm tra các kết quả có như mong muốn, sau đó cam kết các thay đổi.

### B.4.3. Sử dụng `svndumpfilter`

Bởi vì TortoiseSVN không bao giờ mất dữ liệu của, bạn “ cuộn lại ” các sửa đổi vẫn còn tồn tại như các bản sửa đổi trung gian trong kho. Chỉ có phiên bản CHÍNH đã được thay đổi tới trạng thái trước đó. Nếu bạn muốn làm chỉnh sửa biến mất hoàn toàn từ kho lưu trữ của bạn, tẩy xóa tất cả các dấu vết mà chúng đã từng tồn tại, bạn phải sử dụng các biện pháp cực đoan hơn. Trừ khi có một lý do thực sự tốt để làm điều này, nó *không được khuyến khích*. Một lý do có thể sẽ là một người nào đó cam kết một tài liệu bí mật cho một kho lưu trữ công cộng.

The only way to remove data from the repository is to use the Subversion command line tool `svnadmin`. You can find a description of how this works in the [Repository Maintenance](http://svnbook.red-bean.com/en/1.8/svn.reposadmin.maint.html) [http://svnbook.red-bean.com/en/1.8/svn.reposadmin.maint.html].

## B.5. So sánh hai phiên bản của một tập tin hoặc thư mục

Nếu bạn muốn so sánh hai phiên bản trong lịch sử của một mục, ví dụ như sửa đổi 100 và 200 của cùng một tập tin, chỉ cần sử dụng TortoiseSVN → Hiện Tường Trình vào danh sách lịch sử sửa đổi cho tập tin đó. Chọn hai phiên bản bạn muốn so sánh sau đó sử dụng Trình Đơn Ngữ Cảnh → So Sánh Các Sửa Đổi .

Nếu bạn muốn so sánh các mục tương tự trong hai cây khác nhau, ví dụ như các thân cây và một chi nhánh, bạn có thể sử dụng trình duyệt kho lưu trữ để mở cả hai cây, chọn tập tin trong cả hai nơi, sau đó sử dụng Trình Đơn Ngữ Cảnh → So Sánh Các Sửa Đổi .

Nếu bạn muốn so sánh hai cây để xem những gì đã thay đổi, ví dụ như các thân cây và phát hành một được gắn thẻ, bạn có thể sử dụng TortoiseSVN → Đồ Thị Sửa Đổi Chọn hai nút để so sánh, sau đó sử dụng Trình Đơn Ngữ Cảnh → So sánh các sửa đổi CHÍNH . Điều này sẽ hiển thị một danh sách các tập tin thay đổi, và sau đó bạn có thể chọn các tập tin cá nhân để xem các thay đổi chi tiết. Bạn cũng có thể xuất khẩu một cấu trúc cây có chứa tất cả các tập tin thay đổi, hoặc chỉ đơn giản là một danh sách của tất cả các tập tin đã thay đổi. Đọc [Phần 4.10.3, “So sánh các thư mục”](#) để biết thêm thông tin. Ngoài ra sử dụng Trình Đơn Ngữ Cảnh → Diff Thống nhất của Bản Sửa Đổi CHÍNH để xem một bản tóm tắt của tất cả các sự khác biệt, với bối cảnh tối thiểu.

## B.6. Bao gồm các tiểu dự án chung

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. Sử dụng svn:externals

Thiết lập thuộc tính `svn:externals` cho một thư mục trong dự án của bạn. Thuộc tính này bao gồm một hoặc nhiều dòng, mỗi dòng có tên của một thư mục con mà bạn muốn sử dụng như là các thư mục cho kiểm xuất cho mã chung, và URL kho lưu trữ mà bạn muốn được kiểm xuất ở đó. Để có đầy đủ chi tiết tham khảo [Phần 4.18, “Mục bên ngoài”](#) .

Cam kết các thư mục mới. Bây giờ khi bạn cập nhật, Subversion sẽ kéo một bản sao của dự án mà từ kho lưu trữ của nó vào bản sao làm việc của bạn. Các thư mục con sẽ được tạo ra tự động nếu có yêu cầu. Mỗi khi bạn cập nhật bản làm việc chính của bạn, bạn cũng sẽ nhận được phiên bản mới nhất của tất cả các dự án bên ngoài.

Nếu dự án bên ngoài là trong cùng một kho lưu trữ, bất kỳ thay đổi bạn thực hiện sẽ được bao gồm trong danh sách cam kết khi bạn cam kết dự án chính của bạn.

Nếu dự án bên ngoài nằm trong một kho lưu trữ khác, bất kỳ thay đổi bạn thực hiện đến các dự án bên ngoài sẽ được thông báo khi bạn cam kết dự án chính, nhưng bạn phải cam kết những thay đổi bên ngoài một cách riêng biệt.

Trong ba phương pháp mô tả, đây là cách duy nhất mà không cần cài đặt ở phía máy khách. Một khi các thành phần bên ngoài được quy định trong các thuộc tính thư mục, tất cả các máy khách sẽ nhận được các thư mục làm đầy khi chúng cập nhật.

### B.6.2. Sử dụng một bản sao làm việc lồng nhau

Tạo một thư mục mới trong dự án của bạn để chứa các mã thông thường, nhưng không thêm nó vào Subversion.

Chọn TortoiseSVN → Kiểm xuất cho thư mục mới và kiểm xuất một bản sao của mã thông thường vào nó. Bây giờ bạn có một bản sao hoạt động riêng biệt lồng nhau trong bản sao làm việc chính của bạn.

Hai bản sao làm việc là độc lập. Khi bạn thực hiện thay đổi vào bản cha, các thay đổi đến WC lồng nhau được bỏ qua. Tương tự như vậy khi bạn cập nhật các bản cha, WC lồng nhau không được cập nhật.

### B.6.3. Sử dụng một vị trí tương đối

Nếu bạn sử dụng cùng một mã lõi phổ biến ở một số dự án, và bạn không muốn giữ nhiều bản sao làm việc của nó cho mỗi dự án có sử dụng nó, bạn có thể kiểm xuất nó đến một vị trí riêng biệt có liên quan đến tất cả các dự án khác sử dụng nó. Ví dụ:

```
C:\Projects\Proj1
C:\Projects\Proj2
C:\Projects\Proj3
C:\Projects\Common
```

và tham khảo các mã chung bằng cách sử dụng một đường dẫn tương đối, ví dụ như `..\..\Common\DSPcore`.

Nếu các dự án của bạn nằm rải rác tại các địa điểm không liên quan, bạn có thể sử dụng một biến thể của nó, đó là đặt các mã chung ở một vị trí và dùng thay thế ký tự ổ đĩa để ánh xạ vị trí đó vào cái gì bạn có thể mã cứng vào dự án của bạn, ví dụ như Kiểm xuất mã chung vào `D:\Documents\Framework` hoặc `C:\Documents and Settings\{login}\My Documents\framework` sau đó sử dụng

```
SUBST X: "D:\Documents\framework"
```

để tạo ra các ánh xạ ổ đĩa được sử dụng trong mã nguồn của bạn. Mã của bạn sau đó có thể sử dụng địa điểm tuyệt đối.

```
#include "X:\superio\superio.h"
```

Phương pháp này sẽ chỉ làm việc tốt trong tất cả các môi trường máy tính, và bạn sẽ cần phải lập tài liệu ánh xạ ổ đĩa cần thiết để nhóm của bạn biết nơi có các tập tin bí ẩn. Phương pháp này là nghiêm ngặt cho sử dụng trong môi trường phát triển khép kín, và không được khuyến khích sử dụng chung.

#### 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. Tạo một phím tắt đến một kho lưu trữ

Nếu bạn thường xuyên cần phải mở trình duyệt kho lưu trữ tại một địa điểm cụ thể, bạn có thể tạo một shortcut trên desktop bằng cách sử dụng các giao diện tự động hóa đến TortoiseProc. Chỉ cần tạo ra một shortcut mới và đặt mục tiêu:

```
TortoiseProc.exe /command:repobrowser /path:"url/to/repository"
```

Tất nhiên bạn cần phải bao gồm các URL kho thực sự.

### B.8. Bỏ qua các tập tin được đã được phiên bản

Nếu bạn vô tình thêm một số tập tin mà cần phải có được bỏ qua, làm thế nào để bạn lấy chúng ra khỏi kiểm soát phiên bản mà không bị mất chúng? Có lẽ bạn có tập tin cấu hình IDE của riêng bạn mà không phải là một phần của dự án của bạn, nhưng làm bạn mất một thời gian dài để thiết lập theo cách của bạn.

Nếu bạn chưa cam kết việc bỏ sung vào, sau đó tất cả những gì bạn phải làm là sử dụng TortoiseSVN → Hoàn Tác Việc Bỏ Sung... để hoàn tác việc ổ sung. Sau đó, bạn nên thêm các tập tin vào danh sách bỏ qua để chúng không được thêm vào một lần nữa sau đó do nhầm lẫn.

Nếu các tập tin đã có trong kho, chúng phải bị xóa khỏi kho lưu trữ và được thêm vào danh sách bỏ qua. May mắn thay TortoiseSVN có một phím tắt thuận tiện để làm điều này. TortoiseSVN → Bỏ phiên bản và thêm vào

danhsach bỏ qua đầu tiên sẽ đánh dấu các tập tin / thư mục để xóa từ kho, lưu giữ các bản sao địa phương. Nó cũng thêm mục này vào danhsach bỏ qua để nó không được thêm vào Subversion lần nữa do nhầm lẫn. Một khi điều này được thực hiện bạn chỉ cần cam kết thư mục mẹ.

## B.9. Bỏ phiên bản một bản sao hoạt động

Nếu bạn có một bản sao làm việc mà bạn muốn chuyển đổi trở lại thành một cây thư mục đơn giản mà không có thư mục `.svn`, bạn chỉ đơn giản có thể xuất khẩu vào chính nó. Đọc [Phần 4.26.1](#), “[Loại bỏ một bản sao làm việc từ kiểm soát phiên bản](#)” để tìm hiểu cách thức.

## B.10. Loại bỏ một bản sao hoạt động

Nếu bạn có một bản sao hoạt động mà bạn không còn cần, làm thế nào để bạn dẹp bỏ nó một cách sạch sẽ? Dễ dàng - chỉ cần xóa nó trong Windows Explorer! Bản sao làm việc là các đơn vị riêng tư địa phương, và chúng là tự khép kín. Xóa một bản sao làm việc trong Windows Explorer không ảnh hưởng đến các dữ liệu trong kho lưu trữ chút nào.

---

# Phụ lục C. Các Lời Khuyên Hữu Ích Cho Các Quản Trị

Phụ lục này chứa các giải pháp cho các vấn đề / câu hỏi bạn có thể có khi bạn đang chịu trách nhiệm cho việc triển khai TortoiseSVN cho nhiều máy tính khác.

## C.1. Triển khai TortoiseSVN thông qua các chính sách nhóm

Bộ cài đặt TortoiseSVN đến dưới dạng một tập tin MSI, có nghĩa là bạn sẽ không có vấn đề thêm tập tin MSI đó vào các chính sách nhóm của bộ điều khiển tên miền của bạn.

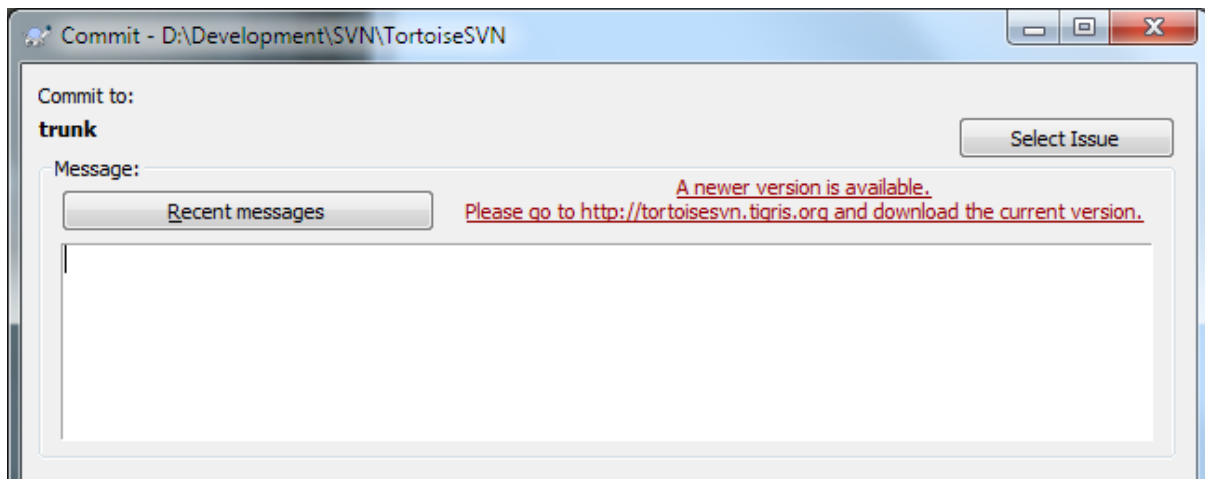
Một cuộc dạo qua về làm thế nào để làm điều đó có thể được tìm thấy ở bài viết trong cơ sở kiến thức 314934 từ Microsoft: <http://support.microsoft.com/?kbid=314934> [<http://support.microsoft.com/?kbid=314934>].

TortoiseSVN must be installed under *Computer Configuration* and not under *User Configuration*. This is because TortoiseSVN needs the CRT and MFC DLLs, which can only be deployed *per computer* and not *per user*. If you really must install TortoiseSVN on a per user basis, then you must first install the MFC and CRT package version 11 from Microsoft on each computer you want to install TortoiseSVN as per user.

Bạn có thể tùy chỉnh các tập tin MSI nếu bạn muốn rằng tất cả người dùng của bạn kết thúc với các thiết lập tương tự. Các thiết lập của TSVN được lưu trữ trong registry dưới HKEY\_CURRENT\_USER\Software\TortoiseSVNfilename> v%APPDATA%\Subversionfilename>. N chuyb>.

## C.2. Chuyển hướng kiểm tra nâng cấp

TortoiseSVN kiểm tra nếu có một phiên bản mới có sẵn vài ngày một lần. Nếu có một phiên bản mới hơn, một thông báo được hiển thị trong hộp thoại cam kết.



**Hình C.1. Hộp thoại cam kết, hiển thị thông báo nâng cấp**

Nếu bạn đang chịu trách nhiệm cho rất nhiều người dùng trong tên miền của bạn, bạn có thể muốn người dùng chỉ sử dụng các phiên bản được bạn phê duyệt và họ không phải lúc nào cũng cài đặt phiên bản mới nhất. Bạn có thể không muốn thông báo nâng cấp đó hiển thị để người dùng của bạn không đi tới đó và nâng cấp ngay lập tức.

Phiên bản 1.4.0 và sau này của TortoiseSVN cho phép bạn chuyển hướng kiểm tra nâng cấp đến máy chủ mạng nội bộ của bạn. Bạn có thể thiết lập các khóa registry HKCU\Software\TortoiseSVN\UpdateCheckURL (giá trị chuỗi) đến một URL trỏ đến một tập tin văn bản trong mạng nội bộ của bạn. Tập tin văn bản đó phải có định dạng sau đây:

1.4.1.6000

Một phiên bản mới của TortoiseSVN hiện có sẵn để bạn có thể tải về!  
<http://192.168.2.1/downloads/TortoiseSVN-1.4.1.6000-svn-1.4.0.msi>

Dòng đầu tiên trong tập tin đó là một chuỗi phiên bản. Bạn phải chắc chắn rằng nó phù hợp với chuỗi phiên bản chính xác của gói cài đặt TortoiseSVN. Dòng thứ hai là một văn bản tùy chỉnh, được hiển thị trong hộp thoại cam kết. Bạn có thể viết bất cứ điều gì bạn muốn. Chỉ cần lưu ý rằng không gian trong hộp thoại cam kết là hạn chế. Thông điệp quá dài sẽ được cắt ngắn! Dòng thứ ba là URL đến gói cài đặt mới. URL này được mở ra khi người dùng nhấp vào nhận thông báo tùy chỉnh trong hộp thoại cam kết. Bạn cũng có thể chỉ người dùng đến một trang web thay vì các tập tin MSI trực tiếp. URL được mở ra với trình duyệt web mặc định, vì vậy nếu bạn chỉ định một trang web, trang đó được mở ra và hiển thị cho người dùng. Nếu bạn chỉ định các gói MSI, trình duyệt sẽ yêu cầu người dùng lưu các tập tin MSI tại địa phương.

### C.3. Thiết lập biến môi trường SVN\_ASP\_DOT\_NET\_HACK

Kể từ phiên bản 1.4.0 và sau đó, trình cài đặt TortoiseSVN không cung cấp cho người sử dụng các tùy chọn để thiết lập biến môi trường SVN\_ASP\_DOT\_NET\_HACK nữa, bởi vì nó gây ra nhiều vấn đề và gây nhầm lẫn cho những người dùng mà luôn luôn cài đặt *tất cả mọi thứ* bất kể cho dù họ có biết nó là cho cái gì hay không.

Nhưng tùy chọn đó là ẩn cho người sử dụng. Bạn vẫn có thể ép buộc trình cài đặt TortoiseSVN thiết lập biến môi trường đó bằng cách thiết lập thuộc tính ASPDOTNETHACK thành TRUE. Ví dụ, bạn có thể bắt đầu trình cài đặt như thế này:

```
msiexec /i TortoiseSVN-1.4.0.msi ASPDOTNETHACK=TRUE
```



#### Quan trọng

Xin lưu ý rằng mảnh này là chỉ cần thiết nếu bạn vẫn còn sử dụng VS.NET2002. Tất cả các phiên bản sau của Visual Studio *không* yêu cầu mảnh này để được kích hoạt! Vì vậy, trừ khi bạn đang sử dụng công cụ cổ xưa đó, **ĐỪNG SỬ DỤNG CÁI NÀY!**

### C.4. Vô hiệu hoá các mục trình đơn ngữ cảnh

Phiên bản 1.5.0 và sau đó, TortoiseSVN cho phép bạn vô hiệu hóa (trên thực tế, **ẩn**) các mục trình đơn ngữ cảnh. Vì đây là một tính năng mà không nên được sử dụng nhẹ nhàng chỉ nếu có một lý do thuyết phục, không có giao diện đồ họa cho nó và nó phải được thực hiện trực tiếp trong registry. Điều này có thể được sử dụng để vô hiệu hóa các lệnh nhất định cho người dùng không nên sử dụng chúng. Nhưng xin lưu ý rằng chỉ các mục trình đơn ngữ cảnh trong *explorer* được ẩn, và các lệnh vẫn dùng được thông qua các phương tiện khác, ví dụ như dòng lệnh hoặc thậm chí các hộp thoại khác trong bản thân TortoiseSVN!

Các khóa registry giữ thông tin mà các trình đơn ngữ cảnh được hiển thị là HKEY\_CURRENT\_USER\Software\TortoiseSVN\ContextMenuEntriesMaskLow và HKEY\_CURRENT\_USER\Software\TortoiseSVN\ContextMenuEntriesMaskHigh .

Mỗi mục trong các mục registry này là một giá trị DWORD , với mỗi bit tương ứng với một mục trình đơn cụ thể. Một bit được bật có nghĩa là mục trình đơn tương ứng bị vô hiệu hoá.

Giá trị	Mục trình đơn
0x0000000000000001	Kiểm xuất

#### Bảng C.1. Các mục trong trình đơn và các giá trị của chúng

Ví dụ: để vô hiệu hóa “Di dời” các mục trình đơn “Xóa các mục không phiên bản” và “Các thiết lập”, thêm các giá trị được gán cho các mục như thế này:

```
0x00000000000080000
+ 0x0000000080000000
+ 0x2000000000000000
= 0x2000000080080000
```

Giá trị DWORD thấp hơn (0x80080000) phải được lưu trong HKEY\_CURRENT\_USER\Software\TortoiseSVN\ContextMenuEntriesMaskLow, giá trị DWORD cao hơn (0x20000000) in HKEY\_CURRENT\_USER\Software\TortoiseSVN\ContextMenuEntriesMaskHigh.

Để kích hoạt các mục trình đơn một lần nữa, chỉ đơn giản là xóa các khóa registry.

# Phụ lục D. Tự động hoá TortoiseSVN

Vì tất cả các lệnh cho TortoiseSVN được kiểm soát thông qua các tham số dòng lệnh, bạn có thể tự động hoá nó với kịch bản hàng loạt hoặc bắt đầu các lệnh và hộp thoại cụ thể từ các chương trình khác (ví dụ như trình soạn thảo văn bản yêu thích của bạn).



## Quan trọng

Hãy nhớ rằng TortoiseSVN là một trình máy khách giao diện đồ họa, và hướng dẫn tự động này cho bạn thấy làm thế nào để làm cho các hộp thoại TortoiseSVN xuất hiện để thu thập đầu vào của người sử dụng. Nếu bạn muốn viết một kịch bản không yêu cầu đầu vào, bạn nên sử dụng dòng lệnh máy khách Subversion chính thức thay thế.

## D.1. Các lệnh TortoiseSVN

Chương trình TortoiseSVN giao diện đồ họa được gọi là `TortoiseProc.exe`. Tất cả các lệnh được quy định cụ thể với tham số `/command:abcd` nơi `abcd` là tên lệnh được yêu cầu. Hầu hết các lệnh này cần ít nhất một đối số đường dẫn, được đưa ra với `/path: "some\path"`. Trong bảng sau lệnh đề cập đến tham số `/command:abcd` và đường dẫn liên quan đến tham số `/path: "some\path"`.

Bởi vì một số lệnh có thể lấy một danh sách các đường dẫn mục tiêu (ví dụ như cam kết một số tập tin cụ thể) tham số `/path` có thể lấy một vài đường dẫn, cách nhau bởi ký tự `*`.

Bạn cũng có thể chỉ định một tập tin, trong đó có một danh sách các đường dẫn, cách nhau bằng ký tự dòng mới. Các tập tin phải được định dạng UTF-16, mà không có một *BOM* [[http://en.wikipedia.org/wiki/Byte-order\\_mark](http://en.wikipedia.org/wiki/Byte-order_mark)]. Nếu bạn truyền một tập tin như vậy, sử dụng `/pathfile` thay vì `/path`. Để TortoiseProc xóa các tập tin sau khi lệnh này kết thúc, bạn có thể truyền vào tham số `/deletepathfile`.

Hộp thoại tiến triển được sử dụng cho các cam kết, cập nhật và các lệnh thường vẫn mở sau khi lệnh đã hoàn tất cho đến khi người dùng nhấn nút OK. Điều này có thể được thay đổi bằng cách kiểm tra các tùy chọn tương ứng trong hộp thoại thiết lập. Nhưng sử dụng thiết lập đó sẽ đóng hộp thoại tiến triển, bất kể nếu bạn bắt đầu lệnh từ tập tin hàng loạt của bạn hoặc từ trình đơn ngữ cảnh TortoiseSVN.

Để chỉ định một vị trí khác nhau của tập tin cấu hình, sử dụng các tham số `/configdir: "path\to\config\directory"`. Việc này sẽ ghi đè lên đường dẫn mặc định, bao gồm cả bất kỳ thiết lập registry.

Để đóng hộp thoại tiến triển ở cuối một lệnh một cách tự động mà không cần sử dụng các thiết lập vĩnh viễn, bạn có thể đưa vào tham số `/closeonend`.

- `/closeonend:0` không tự động đóng hộp thoại
- `/closeonend:1` tự động đóng nếu không có lỗi
- `/closeonend:2` tự động đóng nếu không có lỗi và xung đột
- `/closeonend:3` tự động đóng nếu không có lỗi, xung đột và sáp nhập

Để đóng hộp thoại tiến triển cho các hoạt động địa phương nếu không có lỗi hoặc xung đột, đưa vào tham số `/closeforlocal`.

Bảng dưới đây liệt kê tất cả các lệnh có thể được truy cập bằng cách sử dụng dòng lệnh TortoiseProc.exe. Như đã mô tả ở trên, những lệnh này nên được sử dụng dưới hình thức `/command:abcd`. Trong bảng, tiền tố `/command` được bỏ qua để tiết kiệm không gian.

Lệnh	Mô tả
<code>:about</code>	Hiển thị hộp thoại nói về. Cái này cũng được hiển thị nếu lệnh không được đưa ra.

### Bảng D.1. Danh sách các lệnh có thể dùng và các tùy chọn



Các ví dụ (nên được nhập vào trên một dòng):

```
TortoiseProc.exe /command:commit  
                /path:"c:\svn_wc\file1.txt*c:\svn_wc\file2.txt"  
                /logmsg:"test log message" /closeonend:0
```

```
TortoiseProc.exe /command:update /path:"c:\svn_wc\" /closeonend:0
```

```
TortoiseProc.exe /command:log /path:"c:\svn_wc\file1.txt"  
                /startrev:50 /endrev:60 /closeonend:0
```

## D.2. xử lý Tsvncmd URL

Sử dụng URL đặc biệt, nó cũng có thể gọi TortoiseProc từ một trang web.

TortoiseSVN đăng ký một giao thức mới `tsvncmd:` mà có thể được sử dụng để tạo ra các siêu liên kết thực hiện lệnh TortoiseSVN. Các lệnh và các thông số là tương tự như khi tự động hóa TortoiseSVN từ dòng lệnh.

Định dạng của URL `tsvncmd:` là như thế này:

```
tsvncmd:command:cmd?parameter:paramvalue?parameter:paramvalue
```

với `cmd` là một trong các lệnh được cho phép, tham số là tên của một tham số như `path` hoặc `revision`, và `paramvalue` là giá trị được sử dụng cho tham số đó. Danh sách các thông số được cho phép phụ thuộc vào các lệnh được sử dụng.

Các lệnh sau đây được cho phép với `tsvncmd:` URL:

- `:update`
- `:commit`
- `:diff`
- `:repobrowser`
- `:checkout`
- `:export`
- `:blame`
- `:repostatus`
- `:revisiongraph`
- `:showcompare`
- `:log`

Một URL ví dụ đơn giản có thể trông như thế này:

```
<a href="tsvncmd:command:update?path:c:\svn_wc?rev:1234">Update</a>
```

hoặc trong một trường hợp phức tạp hơn:

```
<a href="tsvncmd:command:showcompare?  
url1:https://stexbar.googlecode.com/svn/trunk/StExBar/src/setup/Setup.wxs?  
url2:https://stexbar.googlecode.com/svn/trunk/StExBar/src/setup/Setup.wxs?
```

```
revision1:188?revision2:189">compare</a>
```

### D.3. Các lệnh TortoiseIDiff

Công cụ tìm khác biệt hình ảnh có một vài tùy chọn dòng lệnh mà bạn có thể sử dụng để kiểm soát cách mà công cụ này được bắt đầu. Chương trình này được gọi là `TortoiseIDiff.exe`.

Bảng dưới đây liệt kê tất cả các tùy chọn có thể được truyền đến công cụ khác biệt hình ảnh trên dòng lệnh.

Tùy chọn	Mô tả
<code>/left</code>	Đường dẫn đến tập tin hiển thị bên trái.

#### Bảng D.2. Danh sách các tùy chọn có sẵn

Ví dụ (nên được nhập vào trên một dòng):

```
TortoiseIDiff.exe /left:"c:\images\img1.jpg" /lefttitle:"image 1"  
                  /right:"c:\images\img2.jpg" /righttitle:"image 2"  
                  /fit /overlay
```

---

# Phụ lục E. Tham Chiếu Chéo Giao Diện Dòng Lệnh

Đôi khi hướng dẫn này đề cập các tài liệu Subversion chính, trong đó mô tả Subversion về Giao Diện Dòng Lệnh (CLI). Để giúp bạn hiểu những gì TortoiseSVN đang làm đằng sau hậu trường, chúng tôi đã biên soạn một danh sách hiển thị các lệnh CLI tương đương cho mỗi tác vụ của Giao Diện Người Dùng Đồ Họa TortoiseSVN.

## Ghi chú

Mặc dù có các CLI tương đương với những gì TortoiseSVN làm, hãy nhớ rằng TortoiseSVN *không* gọi CLI nhưng sử dụng thư viện Subversion trực tiếp.

Nếu bạn nghĩ rằng bạn đã tìm thấy một lỗi trong TortoiseSVN, chúng tôi có thể yêu cầu bạn cố gắng tái tạo nó bằng cách sử dụng CLI, để chúng tôi có thể phân biệt các vấn đề TortoiseSVN với các vấn đề Subversion. Tham chiếu này sẽ cho bạn biết lệnh nào để thử.

## E.1. Các quy ước và quy tắc cơ bản

Trong các mô tả theo sau đây, URL cho một vị trí kho được thể hiện đơn giản như URL, và một ví dụ có thể là `http://tortoisesvn.googlecode.com/svn/trunk/`. Đường dẫn bản sao làm việc được hiển thị đơn giản là PATH, và một ví dụ có thể là `C:\TortoiseSVN\trunk`.



## Quan trọng

Bởi vì TortoiseSVN là một Windows Shell mở rộng, không thể sử dụng khái niệm về một thư mục làm việc hiện tại. Tất cả các đường dẫn bản sao làm việc phải được cung cấp bằng cách sử dụng đường dẫn tuyệt đối, chứ không phải là một đường dẫn tương đối.

Một số mục là tùy chọn, và những cái này thường được kiểm soát bởi các hộp kiểm tra hoặc nút radio trong TortoiseSVN. Các tùy chọn này được hiển thị trong [dấu ngoặc vuông] trong định nghĩa dòng lệnh.

## E.2. Các lệnh TortoiseSVN

### E.2.1. Kiểm xuất

```
svn checkout [-depth ARG] [--ignore-externals] [-r rev] URL PATH
```

Các mục của hộp combo cho chiều sâu liên quan đến thông số `-depth`.

Nếu bỏ qua các phần bên ngoài được chọn, sử dụng chuyển đổi `--ignore-externals`.

Nếu bạn đang kiểm xuất một phiên bản cụ thể, xác định nó sau các URL bằng cách sử dụng chuyển đổi `-r`.

### E.2.2. Cập nhật

```
svn info URL_of_WC
svn update [-r rev] PATH
```

Việc cập nhật nhiều mục hiện nay không phải là một hoạt động nguyên tử trong Subversion. Vì vậy, TortoiseSVN đầu tiên tìm bản sửa đổi CHÍNH của kho, và sau đó cập nhật tất cả các mục với số sửa đổi cụ thể để tránh tạo ra một bản sao làm việc với hỗn hợp sửa đổi.

Nếu chỉ một mục được lựa chọn để cập nhật hoặc các mục đã chọn không phải tất cả từ cùng một kho lưu trữ, TortoiseSVN chỉ cập nhật vào CHÍNH.

Không có tùy chọn dòng lệnh được sử dụng ở đây. Cập nhật đến bản sửa đổi cũng thực hiện các lệnh cập nhật, nhưng cung cấp nhiều lựa chọn hơn.

### E.2.3. Cập nhật đến bản sửa đổi

```
svn info URL_of_WC
svn update [-r rev] [-depth ARG] [--ignore-externals] PATH
```

Các mục của hộp combo cho chiều sâu liên quan đến thông số `-depth`.

Nếu Bỏ qua các phần bên ngoài được chọn, sử dụng chuyển đổi `--ignore-externals`.

### E.2.4. Cam kết

Trong TortoiseSVN, hộp thoại cam kết sử dụng nhiều lệnh Subversion. Giai đoạn đầu tiên là kiểm tra tình trạng mà quyết định các mục trong bản sao làm việc của bạn có khả năng có thể được cam kết. Bạn có thể xem danh sách, tìm khác biệt các tập tin đối với CƠ SỞ và chọn các mục bạn muốn được bao gồm trong cam kết.

```
svn status -v PATH
```

Nếu Hiển thị các tập tin không phiên bản được chọn, TortoiseSVN cũng sẽ hiển thị tất cả các tập tin và thư mục không phiên bản trong hệ thống phân cấp bản sao làm việc, có tính đến các quy tắc bỏ qua. Tính năng này đặc biệt không có tương đương trực tiếp trong Subversion, vì lệnh tình trạng `svn` không đi xuống vào các thư mục không phiên bản.

Nếu bạn chọn bất kỳ tập tin và thư mục không phiên bản, những mục đó trước hết sẽ được thêm vào bản sao làm việc của bạn.

```
svn add PATH...
```

Khi bạn click vào OK, cam kết Subversion diễn ra. Nếu bạn để tất cả các hộp kiểm tra lựa chọn tập tin trong trạng thái mặc định của chúng, TortoiseSVN sử dụng một cam kết đệ quy của bản sao làm việc. Nếu bạn bỏ chọn một số tập tin, thì một cam kết không đệ quy ( `-N` ) phải được sử dụng, và mỗi đường dẫn phải được xác định một cách riêng lẻ trên dòng lệnh cam kết.

```
svn commit -m "LogMessage" [-depth ARG] [--no-unlock] PATH...
```

`LogMessage` ở đây đại diện cho nội dung của hộp biên tập thông điệp tương trình. Cái này có thể được để trống.

Nếu Giữ khóa được duyệt, sử dụng chuyển đổi `--no-unlock`.

### E.2.5. Tìm Khác Biệt

```
svn diff PATH
```

Nếu bạn sử dụng Tìm Khác Biệt từ trình đơn ngữ cảnh chính, bạn đang tìm khác biệt một tập tin thay đổi so với sửa đổi CƠ SỞ của nó. Kết quả của lệnh CLI ở trên cũng thực hiện điều này và tạo ra kết quả trong định dạng khác biệt-thống nhất. Tuy nhiên, đây không phải là những gì TortoiseSVN đang sử dụng. TortoiseSVN sử dụng TortoiseMerge (hoặc một chương trình khác theo lựa chọn của bạn) để hiển thị sự khác biệt trực quan giữa các tập tin văn bản đầy đủ, vì vậy không có CLI tương đương trực tiếp.

Bạn cũng có thể tìm khác biệt 2 tập tin bất kỳ bằng cách sử dụng TortoiseSVN, dù có hoặc không có phiên bản. TortoiseSVN chỉ cấp dữ liệu hai tập tin vào chương trình tìm khác biệt được lựa chọn và để nó tìm ra các khác biệt.

## E.2.6. Hiện Tường Trình

```
svn log -v -r 0:N --limit 100 [--stop-on-copy] PATH
or
svn log -v -r M:N [--stop-on-copy] PATH
```

Theo mặc định, TortoiseSVN cố gắng để lấy 100 thông điệp tường trình bằng cách sử dụng phương pháp `--limit`. Nếu các thiết lập hướng dẫn sử dụng các API cũ, thì hình thức thứ hai được sử dụng để nạp các thông điệp tường trình cho 100 sửa đổi kho lưu trữ.

Nếu Dừng trên bản sao/đổi tên được duyệt, sử dụng chuyển đổi `--stop-on-copy` .

## E.2.7. Kiểm tra xem có sửa đổi

```
svn status -v PATH
or
svn status -u -v PATH
```

Kiểm tra tình trạng ban đầu chỉ nhìn vào bản sao làm việc của bạn. Nếu bạn bấm vào **Kiểm tra kho lưu trữ** thì kho lưu trữ cũng được kiểm tra để xem tập tin sẽ được thay đổi bằng một lệnh cập nhật, trong đó yêu cầu các chuyển đổi `-u` .

Nếu **Hiện thị các tập tin không phiên bản** được chọn, TortoiseSVN cũng sẽ hiện thị tất cả các tập tin và thư mục không phiên bản trong hệ thống phân cấp bản sao làm việc, có tính đến các quy tắc bỏ qua. Tính năng này đặc biệt không có tương đương trực tiếp trong Subversion, vì lệnh tình trạng `svn` không đi xuống vào các thư mục không phiên bản.

## E.2.8. Đồ Thị Sửa Đổi

Đồ thị sửa đổi là một tính năng chỉ của TortoiseSVN. Không có tương đương trong máy khách dòng lệnh.

Những gì TortoiseSVN làm là một

```
svn info URL_of_WC
svn log -v URL
```

noi URL là kho lưu trữ *root* và phân tích dữ liệu được trả về.

## E.2.9. Trình Duyệt Kho Lưu Trữ

```
svn info URL_of_WC
svn list [-r rev] -v URL
```

Bạn có thể sử dụng thông tin `svn` để xác định gốc kho, đó là cấp cao nhất được hiển thị trong trình duyệt kho lưu trữ. Bạn không thể định hướng lên Trên quá mức này. Ngoài ra, lệnh này trả về tất cả các thông tin khóa được hiển thị trong trình duyệt kho.

Việc gọi danh sách `svn` sẽ liệt kê các nội dung của một thư mục, đưa ra một URL và sửa đổi.

## E.2.10. Chinh sửa xung đột

Lệnh này không có tương đương trong CLI. Nó gọi TortoiseMerge hoặc một công cụ tìm khác biệt/hợp nhất theo 3-cách bên ngoài để nhìn vào các tập tin có liên quan trong cuộc xung đột và phân loại các dòng để sử dụng.

## E.2.11. Đã giải quyết

```
svn resolved PATH
```

## E.2.12. Đổi tên

```
svn rename CURR_PATH NEW_PATH
```

## E.2.13. Xóa

```
svn delete PATH
```

## E.2.14. Hoàn nguyên

```
svn status -v PATH
```

Giai đoạn đầu tiên là kiểm tra tình trạng mà quyết định các mục trong bản sao làm việc của bạn có khả năng có thể được hoàn nguyên. Bạn có thể xem lại danh sách, tìm khác biệt các tập tin đối với CỐ SỞ và chọn các mục bạn muốn được bao gồm trong việc hoàn nguyên.

Khi bạn click vào OK, Subversion hoàn nguyên diễn ra. Nếu bạn để tất cả các hộp kiểm lựa chọn tập tin trong trạng thái mặc định của chúng, TortoiseSVN sử dụng một lệnh hoàn nguyên đệ quy duy nhất ( `-R` ) trên bản sao làm việc. Nếu bạn bỏ chọn một số tập tin, thì mọi đường dẫn phải được xác định một cách riêng lẻ trên dòng lệnh hoàn nguyên.

```
svn revert [-R] PATH...
```

## E.2.15. Dọn sạch

```
svn cleanup PATH
```

## E.2.16. Nhận Khóa

```
svn status -v PATH
```

Giai đoạn đầu tiên là kiểm tra tình trạng mà quyết định các tập tin trong bản sao làm việc của bạn có khả năng có thể bị khóa. Bạn có thể chọn các mục bạn muốn được khóa.

```
svn lock -m "LockMessage" [--force] PATH...
```

`LockMessage` ở đây đại diện cho các nội dung của hộp biên tập khóa thông điệp. Điều này có thể được để trống.

Nếu `Ẩn cấp các ổ khóa` được duyệt, sử dụng chuyển đổi `--force`.

## E.2.17. Thả Khóa

```
svn unlock PATH
```

## E.2.18. Nhánh/Thẻ

```
svn copy -m "LogMessage" URL URL
or
svn copy -m "LogMessage" URL@rev URL@rev
or
```

```
svn copy -m "LogMessage" PATH URL
```

Hộp thoại Chi Nhánh/Thẻ thực hiện một sao chép đến kho lưu trữ. Có 3 tùy chọn nút radio:

- bản sửa đổi CHÍNH trong kho
- Sửa đổi cụ thể trong kho
- Bản sao làm việc

tương ứng với 3 biến thể dòng lệnh phía trên.

LogMessage ở đây đại diện cho nội dung của hộp biên tập thông điệp tương trình. Cái này có thể được để trống.

## E.2.19. Chuyển

```
svn info URL_of_WC
svn switch [-r rev] URL PATH
```

## E.2.20. Hợp nhất

```
svn merge [--dry-run] --force From_URL@revN To_URL@revM PATH
```

Các Kiểm tra Hợp nhất thực hiện việc hợp nhất cùng với chuyển đổi `--dry-run` .

```
svn diff From_URL@revN To_URL@revM
```

Các Khác biệt thống nhất cho thấy các tác vụ tìm khác biệt sẽ được sử dụng để thực hiện hợp nhất.

## E.2.21. Xuất khẩu

```
svn export [-r rev] [--ignore-externals] URL Export_PATH
```

Hình thức này được sử dụng khi truy cập từ một thư mục không phiên bản, và thư mục được sử dụng làm điểm đến.

Xuất khẩu một bản sao làm việc đến một vị trí khác được thực hiện mà không cần sử dụng thư viện Subversion, do đó, không có dòng lệnh tương đương được khớp.

Những gì TortoiseSVN làm là sao chép tất cả tập tin đến vị trí mới trong khi hiển thị cho bạn tiến triển của hoạt động này. Tập tin /thư mục không phiên bản theo tùy chọn cũng có thể được xuất khẩu.

Trong cả hai trường hợp, nếu Bỏ qua thành phần ngoài được chọn, sử dụng chuyển đổi `-ignore-externals` .

## E.2.22. Di dời

```
svn switch --relocate From_URL To_URL
```

## E.2.23. Tạo Kho Chứa Ở Đây

```
svnadmin create --fs-type fsfs PATH
```

## E.2.24. Thêm

```
svn add PATH...
```

Nếu bạn chọn một thư mục, TortoiseSVN đầu tiên quét nó theo cách đệ quy cho các hạng mục có thể được thêm vào.

### E.2.25. Nhập khẩu

```
svn import -m LogMessage PATH URL
```

LogMessage ở đây đại diện cho nội dung của hộp biên tập thông điệp tường trình. Cái này có thể được để trống.

### E.2.26. Đồ lỗi

```
svn blame -r N:M -v PATH  
svn log -r N:M PATH
```

Nếu bạn sử dụng TortoiseBlame để xem các thông tin đồ lỗi, tập tin tường trình cũng là cần thiết để hiển thị các thông báo tường trình vào một tooltip. Nếu bạn xem đồ lỗi như một tập tin văn bản, thông tin này là không cần thiết.

### E.2.27. Thêm vào Danh sách Bỏ qua

```
svn propget svn:ignore PATH > tempfile  
{edit new ignore item into tempfile}  
svn propset svn:ignore -F tempfile PATH
```

Bởi vì thuộc tính `svn:ignore` thường là một giá trị nhiều dòng, nó được hiển thị ở đây như là được thay đổi thông qua một tập tin văn bản hơn là trực tiếp trên dòng lệnh.

### E.2.28. Tạo Bản Vá

```
svn diff PATH > patch-file
```

TortoiseSVN tạo ra một tập tin vá lỗi trong định dạng khác biệt thống nhất bằng cách so sánh bản sao làm việc với phiên bản **CƠ SỞ** của nó.

### E.2.29. Áp dụng Bản vá

Áp dụng các bản vá lỗi là một việc khó khăn, trừ khi các bản vá và bản sao làm việc ở trong cùng một sửa đổi. May mắn cho bạn, bạn có thể sử dụng TortoiseMerge, mà không có tương đương trực tiếp trong Subversion.



---

# Phụ lục F. Thông tin chi tiết Thực hiện

Phụ lục này chứa một thảo luận chi tiết hơn về việc thực hiện một số tính năng TortoiseSVN.

## F.1. Biểu tượng Lớp phủ

Mỗi tập tin và thư mục có một giá trị trạng thái Subversion như theo báo cáo của các thư viện Subversion. Trong các máy khách dòng lệnh, những cái này đại diện bởi các mã chữ cái, nhưng trong TortoiseSVN chúng được hiển thị bằng đồ họa bằng cách sử dụng lớp phủ biểu tượng. Bởi vì số lượng của lớp phủ là rất hạn chế, mỗi lớp phủ có thể đại diện cho một trong nhiều giá trị trạng thái.



Các *Lớp phủ mâu thuẫn* được sử dụng để đại diện cho trạng thái *mâu thuẫn*, nơi một cập nhật hoặc chuyển đổi cho kết quả là các cuộc xung đột giữa các thay đổi địa phương và thay đổi tải về từ kho. Nó cũng được sử dụng để chỉ ra tình trạng *tắc nghẽn*, mà có thể xảy ra khi một thao tác không thể hoàn thành được.



Lớp phủ *Thay đổi* đại diện cho trạng thái *sửa đổi*, nơi bạn đã thực hiện các thay đổi địa phương, trạng thái *sáp nhập*, nơi thay đổi từ kho lưu trữ đã được sáp nhập với những thay đổi địa phương, và trạng thái *thay thế*, nơi một tập tin đã bị xóa và được thay thế bởi một tập tin khác cùng tên.



Lớp phủ *Đã bị xóa* đại diện cho trạng thái *xóa*, nơi một mục được lên kế hoạch để xóa, hoặc trạng thái *thiếu mất*, nơi một mục không có mặt. Đương nhiên một mục thiếu mất không thể có một lớp phủ riêng của mình, nhưng các thư mục mẹ có thể được đánh dấu nếu một trong các mục con của nó bị thiếu mất.



Lớp phủ *Thêm* chỉ đơn giản là được sử dụng để đại diện cho trạng thái *thêm* khi một mục đã 0được thêm vào kiểm soát phiên bản.



Lớp phủ *Trong Subversion* được sử dụng để đại diện cho một mục trong trạng thái *bình thường*, hoặc một mục có phiên bản mà trạng thái vẫn chưa được biết. Bởi vì TortoiseSVN sử dụng một quá trình ngầm lưu bộ nhớ đệm để thu thập trạng thái, nó có thể mất một vài giây trước khi các lớp phủ được cập nhật.



Lớp phủ *Cần Khóa* được sử dụng để chỉ ra khi một tập tin có thuộc tính `svn:needs-lock` được thiết lập.



Lớp phủ *Đã bị khoá* được sử dụng khi bản sao làm việc địa phương giữ một khóa cho tập tin đó.



Lớp phủ *Bỏ qua* được sử dụng để đại diện cho một mục trong trạng thái *bỏ qua*, hoặc là do một mô hình bỏ qua toàn cục, hoặc do thuộc tính `svn:ignore` của thư mục mẹ. Lớp phủ này là tùy chọn.



Lớp phủ *Không phiên bản* được sử dụng để đại diện cho một mục trong trạng thái không phiên bản. Đây là một mục trong một thư mục được phiên bản, nhưng bản thân nó không thuộc kiểm soát phiên bản. Lớp phủ này là tùy chọn.

Nếu một mục có trạng thái Subversion là `khôngliteral> (mBnhKhh)`

Một mục chỉ có thể có một giá trị tình trạng Subversion. Ví dụ một tập tin có thể được thay đổi tại địa phương và nó có thể được đánh dấu để xóa cùng một lúc. Subversion trả về một giá trị tình trạng đơn lẻ - trong trường hợp này là được xóa. Những ưu tiên được xác định trong bản thân Subversion.

Khi TortoiseSVN hiển thị trạng thái một cách đệ quy (thiết lập mặc định), mỗi thư mục sẽ hiển thị một lớp phủ phản ánh trạng thái của mình và tình trạng của tất cả các con của nó. Để hiển thị một lớp phủ *tóm tắt*, chúng tôi sử dụng thứ tự ưu tiên hiển thị ở trên để xác định lớp phủ để sử dụng, với lớp phủ *Mâu thuẫn* được ưu tiên cao nhất.

Trong thực tế, bạn có thể thấy rằng không phải tất cả các biểu tượng này được sử dụng trên hệ thống của bạn. Điều này là do số lượng các lớp phủ cho phép Windows được giới hạn đến 15. Windows sử dụng 4 cái, và còn lại 11 có thể được sử dụng bởi các ứng dụng khác. Nếu không có lô lớp phủ sẵn đủ, TortoiseSVN cố gắng là một *Công dân tốt (TM)* và giới hạn sử dụng lớp phủ để dành cơ hội cho các ứng dụng khác.

Kể từ khi có các trình khách Tortoise có sẵn cho các hệ thống kiểm soát phiên bản khác, chúng tôi đã tạo ra một thành phần chia sẻ chịu trách nhiệm cho việc hiển thị các biểu tượng lớp phủ. Các chi tiết kỹ thuật không quan trọng ở đây, tất cả những gì bạn cần phải biết là thành phần chia sẻ này cho phép tất cả các trình khách Tortoise sử dụng các lớp phủ tương tự và do đó giới hạn của 11 lô có sẵn không được sử dụng hết bằng cách cài đặt nhiều hơn một trình khách Tortoise. Tất nhiên có một nhược điểm nhỏ: tất cả các trình khách Tortoise sử dụng cùng các biểu tượng che phủ, vì vậy bạn không thể nhìn biểu tượng che phủ mà suy ra hệ thống điều khiển phiên bản nào mà một bản sao làm việc đang sử dụng.

- *Bình thường*, *Được thay đổi* và *Mâu thuẫn* luôn luôn được nạp và có thể nhìn thấy.
- *Đã bị xóa* được tải nếu có thể, nhưng rơi trở lại *Thay đổi* nếu không có đủ lô.
- *Chi-Đọc* được tải nếu có thể, nhưng rơi trở lại *Bình thường* nếu không đủ lô.
- *Đã bị khoá* được tải nếu có thể, nhưng rơi trở lại *Bình thường* nếu không có đủ lô.
- *Được Thêm* được tải nếu có thể, nhưng rơi trở lại *Được thay đổi* nếu không có đủ lô.

---

# Phụ lục G. Các gói ngôn ngữ và bộ kiểm tra chính tả

Bộ cài đặt tiêu chuẩn chỉ có hỗ trợ dành cho tiếng Anh, nhưng bạn có thể tải về gói ngôn ngữ riêng biệt và các từ điển kiểm tra đánh vần riêng rẽ sau khi cài đặt.

## G.1. Gói Ngôn Ngữ

Giao diện người TortoiseSVN đã được dịch ra nhiều ngôn ngữ khác nhau, vì vậy bạn có thể tải về gói ngôn ngữ phù hợp với nhu cầu của bạn. Bạn có thể tìm thấy các gói ngôn ngữ của chúng tôi [trang tình trạng phiên dịch](http://tortoisesvn.net/translation_status) [http://tortoisesvn.net/translation\_status]. Và nếu không có gói ngôn ngữ có sẵn, tại sao không tham gia nhóm nghiên cứu và nộp bản dịch của bạn ;-)

Mỗi gói ngôn ngữ được đóng gói như là một bộ cài đặt `.msi`. Chỉ cần chạy chương trình cài đặt và làm theo hướng dẫn. Sau khi kết thúc cài đặt, bản dịch sẽ sẵn sàng.

Tài liệu cũng đã được dịch sang nhiều ngôn ngữ khác nhau. Bạn có thể tải về các hướng dẫn sử dụng dịch từ [Trang hỗ trợ](http://tortoisesvn.net/support) [http://tortoisesvn.net/support] trên trang web của chúng tôi.

## G.2. Bộ kiểm tra chính tả

TortoiseSVN bao gồm bộ kiểm tra chính tả mà cho phép bạn kiểm tra các thông điệp cam kết của bạn. Điều này đặc biệt hữu ích nếu ngôn ngữ của dự án không phải ngôn ngữ mẹ đẻ của bạn. Kiểm tra chính tả sử dụng các tập tin từ điển giống như [OpenOffice](http://openoffice.org) [http://openoffice.org] và [Mozilla](http://mozilla.org) [http://mozilla.org].

Trình cài đặt tự động thêm các từ điển tiếng Anh (Mỹ) và Anh (Anh). Nếu bạn muốn các ngôn ngữ khác, tùy chọn dễ dàng nhất chỉ đơn giản là cài đặt một gói ngôn ngữ của TortoiseSVN. Việc này sẽ cài đặt các tập tin từ điển thích hợp cũng như giao diện người sử dụng TortoiseSVN địa phương. Sau khi kết thúc cài đặt, từ điển cũng sẽ sẵn sàng.

Hoặc bạn có thể cài đặt các bộ từ điển chính mình. Nếu bạn có cài đặt OpenOffice hoặc Mozilla, bạn có thể sao chép các từ điển này, được đặt trong thư mục cài đặt cho những ứng dụng đó. Nếu không, bạn cần phải tải về tập tin từ điển cần thiết từ <http://wiki.services.openoffice.org/wiki/Dictionaries> [http://wiki.services.openoffice.org/wiki/Dictionaries].

Một khi bạn đã có các tập tin từ điển, bạn có thể cần phải đổi tên của chúng để tên tập tin chỉ có các ký tự địa phương trong đó. Ví dụ:

- en\_US.aff
- en\_US.dic

Sau đó chỉ cần sao chép chúng vào thư mục `con bin` của thư mục cài đặt TortoiseSVN. Thông thường đây này sẽ được `C:\Program Files\TortoiseSVN\bin`. Nếu bạn không muốn xả rác thư mục `con bin`, thay vào đó bạn có thể đặt các tập tin kiểm tra chính tả của bạn trong `C:\Program Files\TortoiseSVN\Languages`. Nếu thư mục đó không có ở đó, bạn phải tạo ra nó trước. Lần tiếp theo bạn bắt đầu TortoiseSVN, bộ kiểm tra chính tả sẽ sẵn sàng.

Nếu bạn cài đặt nhiều từ điển, TortoiseSVN sử dụng các quy tắc này để lựa chọn cái nào để sử dụng.

1. Kiểm tra thiết lập `tsvn:projectlanguage`. Tham khảo [Phần 4.17, “Thiết lập Dự án”](#) cho thông tin về thiết lập các thuộc tính dự án.
2. Nếu không có ngôn ngữ dự án được thiết lập, hoặc ngôn ngữ đó không được cài đặt, hãy thử ngôn ngữ tương ứng với văn hóa địa phương Windows.
3. Nếu văn hóa địa phương Windows chính xác không làm việc, thử ngôn ngữ “Cơ sở”, ví dụ như `de_CH` (tiếng Đức-Thụy Sĩ) rơi trở lại `de_DE` (tiếng Đức).

4. Nếu không có cái nào bên trên chạy được, thì ngôn ngữ mặc định là tiếng Anh, được bao gồm trong cài đặt tiêu chuẩn.

---

# Thuật ngữ

Bản sao làm việc	Đây là “ sandbox ” địa phương của bạn, khu vực nơi bạn làm việc trên các tập tin có phiên bản, và nó thường cư trú trên đĩa cứng địa phương của bạn. Bạn tạo một bản sao làm việc bằng cách làm một “ Kiểm xuất ” từ một kho lưu trữ, và bạn nạp thay đổi của bạn trở lại vào kho lưu trữ bằng cách sử dụng “ Cam kết ” .
Bản sửa đổi CƠ SỞ	Bản sửa đổi cơ bản hiện hành của một tập tin hoặc thư mục trong <i>bản sao làm việc</i> của bạn. Đây là bản sửa đổi mà tập tin hoặc thư mục đã nằm bên trong, khi kiểm xuất, cập nhật hoặc cam kết mới nhất đã được chạy. Bản sửa đổi CƠ SỞ thường không bằng với bản sửa đổi CHÍNH.
Bản vá	Nếu một bản sao làm việc chỉ có các thay đổi đến các tập tin văn bản, nó có thể sử dụng lệnh Diff Subversion để tạo ra một bản tóm tắt tập tin duy nhất của những thay đổi trong định dạng Khác biệt Thống nhất. Một tập tin loại này thường được nhắc đến như là một “ Bản vá ” , và nó có thể được gửi qua email cho người khác (hoặc một danh sách gửi thư) và áp dụng cho một bản sao làm việc. Một người nào đó không có truy cập cam kết có thể tạo thay đổi và gửi tập tin vá cho một người cam kết có thẩm quyền để áp dụng. Hoặc nếu bạn không chắc chắn về sự thay đổi, bạn có thể gửi một bản vá cho những người khác để xem xét.
BDB	Berkeley DB. Một cơ sở dữ liệu đằng sau được thử nghiệm kỹ cho các kho lưu trữ, không có thể được sử dụng trên mạng chia sẻ. Mặc định cho các kho trước 1.2.
Cam kết	Lệnh Subversion được sử dụng để chuyển những thay đổi trong bản sao làm việc địa phương của bạn trở lại vào kho lưu trữ, tạo ra một phiên bản kho lưu trữ mới.
Cập nhật	Lệnh Subversion này kéo xuống những thay đổi mới nhất từ kho lưu trữ vào bản sao làm việc của bạn, sáp nhập bất kỳ thay đổi được thực hiện bởi những người khác với những thay đổi trong bản sao làm việc.
Chi nhánh	Một thuật ngữ thường được sử dụng trong các hệ thống kiểm soát sửa đổi để mô tả những gì xảy ra khi các phát triển phân nhánh tại một điểm cụ thể và đi theo 2 đường riêng biệt. Bạn có thể tạo ra một chi nhánh từ dòng phát triển chính để phát triển một tính năng mới mà không làm cho các dòng chính mất ổn định. Hoặc bạn có thể phân nhánh một bản phát hành ổn định mà bạn chỉ thực hiện sửa lỗi, trong khi phát triển mới diễn ra trên thân cây không ổn định. Trong Subversion một chi nhánh được thực hiện như một “ bản sao giá rẻ ” .
Chuyển	Cũng như “ Cập nhật-đến-sửa đổi ” thay đổi cửa sổ thời gian của một bản sao làm việc để nhìn vào một điểm khác nhau trong lịch sử, vì vậy “ Chuyển ” thay đổi cửa sổ không gian của một bản sao làm việc để nó trở tới một phần khác nhau của kho. Nó đặc biệt hữu ích khi làm việc trên thân cây và các nhánh chỉ có một vài tập tin khác nhau. Bạn có thể chuyển bản sao làm việc của bạn giữa hai phần này và chỉ thay đổi các tập tin sẽ được chuyển giao.
Di dời	Nếu di chuyển kho lưu trữ của bạn, có lẽ bởi vì bạn đã di chuyển nó vào một thư mục khác trên máy chủ của bạn, hoặc tên miền máy chủ đã thay đổi, bạn cần phải “ di dời ” bản sao của bạn làm việc để các URL kho lưu trữ chỉ đến vị trí mới.  Lưu ý: bạn chỉ nên sử dụng lệnh này nếu bản sao làm việc của bạn là đề cập đến cùng một vị trí trong cùng một kho lưu trữ, nhưng kho lưu trữ chính nó đã di chuyển. Trong bất kỳ hoàn cảnh khác, bạn có thể cần lệnh “ Chuyển ” thay vào đó.

Đồ lỗi	Lệnh này chỉ được dành cho các tập tin văn bản, và chú giải tất cả các dòng để hiển thị sửa đổi kho lưu trữ trong lần cuối nó được thay đổi, và tác giả thực hiện thay đổi đó. Giao Diện Người Dùng Đồ Họa của chúng tôi là được gọi là TortoiseBlame và nó cũng cho thấy ngày / thời gian cam kết và thông điệp trường trình khi bạn di chuyển chuột trên số sửa đổi.
Dọn sạch	Trích dẫn từ cuốn sách Subversion: “Đệ quy làm sạch bản sao làm việc, loại bỏ các khóa và nói lại các tác vụ chưa hoàn thành. Nếu bạn đã từng có được một <i>bản sao làm việc bị khóa</i> lỗi, chạy lệnh này để loại bỏ ổ khóa cũ và nhận bản sao của bạn làm việc vào một trạng thái có thể sử dụng lại.” Lưu ý rằng trong bối cảnh này <i>khóa</i> có nghĩa là khóa hệ thống tập tin địa phương, chứ không phải là khóa kho lưu trữ.
FSFS	Một hệ thống tập tin phụ trợ Subversion độc quyền cho các kho. Có thể được sử dụng trên mạng chia sẻ. Mặc định cho 1.2 và các kho mới hơn.
Giải quyết	Khi các tập tin trong một bản sao làm việc được giữ lại trong trạng thái mâu thuẫn sau khi hợp nhất, những xung đột đó phải được sắp xếp ra bởi một con người bằng cách sử dụng một trình soạn thảo (hoặc có lẽ TortoiseMerge). Quá trình này được gọi là “Giải quyết mâu thuẫn”. Khi điều này hoàn tất, bạn có thể đánh dấu các tập tin mâu thuẫn là đã được giải quyết, cho phép chúng được cam kết.
GPO	đổi tượng chính sách nhóm.
Hoàn nguyên	Subversion giữ một bản sao địa phương “nguyên sơ” của mỗi tập tin khi bạn cập nhật bản sao làm việc của bạn lần cuối. Nếu bạn đã thực hiện thay đổi và quyết định bạn muốn phục hồi lại chúng, bạn có thể sử dụng lệnh “hoàn nguyên” để quay trở lại bản sao nguyên sơ.
Hợp nhất	<p>Quá trình mà các thay đổi từ kho được bổ sung vào bản sao làm việc của bạn mà không làm gián đoạn bất kỳ thay đổi mà bạn đã thực hiện tại địa phương. Đôi khi những thay đổi này có thể không được dàn xếp tự động và bản sao làm việc được gọi là đang có xung đột.</p> <p>Kết hợp sẽ xảy ra tự động khi bạn cập nhật bản sao làm việc của bạn. Bạn cũng có thể sáp nhập các thay đổi cụ thể từ một chi nhánh khác bằng cách sử dụng lệnh Merge TortoiseSVN.</p>
Kho lưu trữ	Kho lưu trữ là một địa điểm trung tâm nơi dữ liệu được lưu trữ và duy trì. Một kho có thể là một nơi nhiều cơ sở dữ liệu hoặc các tập tin được đặt để phân phối qua mạng, hoặc một kho lưu trữ có thể là một địa điểm được truy cập trực tiếp cho người sử dụng mà không cần phải đi trên mạng.
Khóa	Khi bạn lấy ra khóa trên một mục có phiên bản, bạn đánh dấu nó trong kho lưu trữ như là không thể cam kết, ngoại trừ từ bản sao làm việc nơi khóa đã được lấy ra.
Kiểm xuất	Một lệnh Subversion tạo ra một bản sao hoạt động địa phương trong một thư mục rỗng bằng cách tải về các tập tin có phiên bản từ kho lưu trữ.
Lịch sử	Hiển thị lịch sử sửa đổi của một tập tin hoặc thư mục. Còn được gọi là “Tuồng Trình”.
Nhập khẩu	Lệnh Subversion để nhập khẩu toàn bộ hệ thống phân cấp thư mục vào kho lưu trữ trong một bản sửa đổi duy nhất.
Sao chép	Trong một kho lưu trữ Subversion, bạn có thể tạo một bản sao của một tập tin hoặc toàn bộ một cây. Những cái này được thực hiện như “bản sao giá rẻ” hành động giống như một liên kết tới bản gốc ở chỗ chung gần như không

	<p>chiếm không gian nào. Làm một bản sao bảo tồn lịch sử của các mục trong bản sao, vì vậy bạn có thể theo dõi những thay đổi được thực hiện trước khi bản sao chép được tạo ra.</p>
Sửa đổi	<p>Mỗi lần bạn thực hiện một bộ thay đổi, bạn tạo mới “ bản sửa đổi ” trong kho. Mỗi bản sửa đổi thể hiện trạng thái của cây kho lưu trữ tại một điểm nhất định trong lịch sử của nó. Nếu bạn muốn quay trở lại trong thời gian bạn có thể kiểm tra các kho lưu trữ tại bản sửa đổi N.</p> <p>Trong một nghĩa khác, một bản sửa đổi có thể tham khảo các thiết lập của những thay đổi đã được thực hiện khi bản sửa đổi đã được tạo ra.</p>
Sửa đổi CHÍNH	Các phiên bản mới nhất của một tập tin hoặc thư mục trong <i>kho</i> .
SVN	Viết tắt thường được sử dụng cho Subversion.
	Tên của giao thức tùy chỉnh Subversion được sử dụng bởi kho lưu trữ máy chủ “ svnservice ”.
Thêm	Một lệnh Subversion được sử dụng để thêm một tập tin hoặc thư mục vào bản sao làm việc của bạn. Các mục mới được thêm vào kho lưu trữ khi bạn cam kết.
Thuộc tính	Ngoài việc phiên bản các thư mục và tập tin của bạn, Subversion cho phép bạn thêm siêu dữ liệu có phiên bản - được gọi là “ thuộc tính ” cho mỗi thư mục và tập tin có phiên bản của bạn. Mỗi thuộc tính có một cái tên và một giá trị, giống như một khóa registry. Subversion có một số tính chất đặc biệt mà nó sử dụng nội bộ, chẳng hạn như <code>svn:eol-style</code> . TortoiseSVN cũng có một số, chẳng hạn như <code>tsvn:logminsize</code> . Bạn có thể thêm các thuộc tính của riêng bạn với bất kỳ tên và giá trị mà bạn chọn.
Thuộc Tính Sửa Đổi (revprop)	Cũng như các tập tin có thể có các thuộc tính, mỗi bản sửa đổi trong kho cũng có thể. Một số revprops đặc biệt được thêm vào tự động khi bản sửa đổi được tạo ra, cụ thể là: <code>svn:date</code> <code>svn:author</code> <code>svn:log</code> đại diện cho ngày / thời gian cam kết, người cam kết và thông điệp tường trình tương ứng. Các thuộc tính này có thể được chỉnh sửa, nhưng họ không phải được phiên bản, vì vậy thay đổi là vĩnh viễn và không thể được hoàn tác.
Tìm Khác Biệt	Viết tắt cho “ Hiển thị khác biệt ” . Rất hữu ích khi bạn muốn xem chính xác những thay đổi gì đã được thực hiện.
Tường Trình	Hiển thị lịch sử sửa đổi của một tập tin hoặc thư mục. Còn được gọi là “ Lịch sử ” .
Xóa	Khi bạn xóa một mục có phiên bản (và cam kết các thay đổi) mục đó không còn tồn tại trong kho lưu trữ sau khi sửa đổi cam kết. Nhưng tất nhiên nó vẫn còn tồn tại trong các phiên bản kho lưu trữ trước đó, vì vậy bạn vẫn có thể truy cập vào nó. Nếu cần thiết, bạn có thể sao chép một mục đã bị xóa và “ hồi sinh ” nó hoàn thành với lịch sử.
Xuất khẩu	Lệnh này tạo ra một bản sao của một thư mục có phiên bản, giống như một bản sao làm việc, nhưng không có thư mục <code>.svn</code> địa phương.
Xung đột	Khi các thay đổi từ kho được sáp nhập với những thay đổi của địa phương, đôi khi những thay đổi đó xảy ra trên cùng một dòng. Trong trường hợp này Subversion không thể tự động quyết định phiên bản nào để sử dụng và tập tin được cho là nằm trong cuộc xung đột. Bạn có thể chỉnh sửa các tập tin bằng tay và giải quyết cuộc xung đột trước khi bạn có thể cam kết bất kỳ thay đổi tiếp theo.

---

# Chỉ mục

## A

âm thanh, 130

## B

bản dịch, 196  
bản sao làm việc , 11  
bản sao làm việc chưa phiên b, 122  
bản vá, 109  
biểu tượng, 43  
bộ điều khiển tên miền, 182  
bộ kiểm tra chính tả, 196  
bộ lọc, 61  
bộ nhớ cache của xác thực, 25  
bỏ qua, 73  
bỏ qua toàn cục, 131  
bộ theo dõi lỗi, 124  
bộ theo dõi vấn đề , 124, 171  
bugtracker, 124

## C

các công cụ hợp nhất, 71  
Các dự án ASP, 183  
các dự án chung, 179  
các dự án của nhà cung cấp, 179  
các kho bên ngoài, 92  
các mục trình đơn ngữ cảnh, 183  
các phần bên ngoài, 92, 179  
các sửa đổi, 46  
các tập tin / thư mục không được phiên bản, 73  
các tập tin đặc biệt, 28  
các tập tin tạm thời, 26  
các thông điệp cam kết, 51  
các thông điệp tường trình, 51  
các thuộc tính sửa đổi, 60  
các xung đột hợp nhất, 104  
cài đặt, 1  
cam kết, 31  
cập nhật, 37, 177  
chỉ đọc, 106  
chi nhánh, 72, 94  
chính sách nhóm, 182, 183  
chỉnh sửa tường trình / tác giả, 60  
chú thích, 111  
chương trình xem kho lưu trữ, 129  
chuyển, 97  
CLI, 188  
COM, 166, 171  
công cụ khác biệt, 71  
Cột Explorer, 45  
Cuốn sách Subversion, 7

## D

đang tìm khác biệt, 48  
đánh dấu phát hành, 94

danh sách thay đổi, 49  
di chuyển, 77, 177  
di chuyển các tập tin, 72  
di dời, 123  
đồ lỗi, 111  
đồ thị, 117  
đồ thị sửa đổi, 117  
đổi tên, 77, 114, 177  
đổi tên tập tin, 72  
dọn sạch, 78, 80  
dòng lệnh, 185, 187  
Đường dẫn UNC, 17

## E

explorer, xi

## F

FAQ, 176

## G

giải quyết, 38  
Giao diện COM SubWCRev, 168  
gói ngôn ngữ, 196  
GPO, 182  
gửi những thay đổi, 31

## H

hoàn nguyên, 78, 178  
hoàn tác, 78  
hoàn tác cam kết, 178  
hoàn tác thay đổi, 178  
hook máy khách, 155  
hợp nhất, 98  
    hai cây, 101  
    phạm vi sửa đổi, 99

## I

IBugtraqProvider, 171

## K

kéo phải, 24  
kéo-và-thả, 24  
khác, 66, 109  
khen ngợi, 111  
kho, 7, 26  
khóa, 106  
không phiên bản, 123, 180  
khớp mẫu, 74  
khuôn mẫu loại trừ, 131  
kịch bản móc, 20, 155  
kịch bản móc phía máy chủ, 20  
kiểm nh, 31  
kiểm soát phiên bản, xi  
kiểm tra phiên bản mới, 182  
Kiểm xu, 29  
Kiểm xuất rời rạc, 29  
kiểm xuất từng phần, 29



## L

lấy thay đổi, 37  
 lịch sử, 51  
 liên kết, 20  
 liên kết kiểm xu, 20  
 liên kết TortoiseSVN, 20  
 loại bỏ, 76  
 loại bỏ việc phiên bản, 180  
 lớp phủ, 43, 194  
 lớp phủ ưu tiên, 194  
 lưu đệm tường trình, 152

## M

Mạng chia sẻ, 17  
 máy chủ di chuyển, 123  
 máy chủ đã được di chuyển, 123  
 máy chủ proxy, 145  
 máy khách dòng lệnh, 188  
 Microsoft Word, 71  
 mở rộng các từ khóa, 81  
 móc, 20  
 msi, 182

## N

nâng cấp kiểm tra, 182  
 nh chuột phải, 22  
 nhập khẩu, 26  
 Nhập khẩu tại chỗ, 28

## O

ổ đĩa SUBST, 144

## P

phiên bản, 182  
 phím tắt, 180  
 phục hồi, 178  
 plugin, 171  
 props-tự động, 83

## R

registry, 160  
 repo-browser, 114  
 revprops, 60

## S

sao chép, 94, 114  
 sao chép tập tin, 72  
 sao lưu, 19  
 server-side actions, 114  
 số liệu thống kê, 63  
 số phiên bản trong các tập tin, 166  
 so sánh, 66  
 so sánh các bản sửa đổi, 68  
 so sánh các tập tin, 179  
 so sánh các thư mục, 179  
 sửa đổi, 13, 117

SubWCRev, 166  
 SVN\_ASP\_DOT\_NET\_HACK, 183

## T

tách ra khỏi kho lưu trữ, 180  
 tái tích hợp sáp nhập, 105  
 Tạo, 16  
     TortoiseSVN, 16  
 tạo phiên bản các tập tin mới, 72  
 tạo ra bản sao làm việc, 29  
 tạo ra kho lưu trữ, 16  
 thay đổi, 179  
 thẻ, 72, 94  
 thêm, 72  
 thêm các tập tin vào kho lưu trữ, 26  
 theo dõi hợp nhất, 103  
 theo dõi lỗi, 124  
 thiết lập, 130  
 thông điệp cam kết, 177  
 thông điệp rỗng, 177  
 thông điệp tường trình, 177  
 thuộc tính cửa sổ, 44  
 thuộc tính dự án, 84  
 thuộc tính Subversion, 80  
 Thuộc tính TortoiseSVN, 84  
 tìm khác biệt hình ảnh, 70  
 tìm khác biệt thống nhất, 109  
 tình trạng, 43, 46  
 tình trạng bản sao làm việc, 43  
 tổ chức lại, 177  
 toàn cục hóa, 74  
 tối đa hóa, 26  
 TortoiseIDiff, 70  
 trang web, 20  
 trích xuất phiên bản, 166  
 triển khai, 182  
 trình đơn ngữ cảnh, 22  
 trình xem máy chủ, 114  
 Truy cập, 17  
 từ điển, 196  
 tự động hóa, 185, 186, 187  
 từ khóa, 81  
 tường trình, 51  
 tường trình theo dõi hợp nhất, 60

## U

URL b thay đổi, 123  
 URL kho lưu trữ bị thay đổi, 123

## V

ViewVC, 129  
 vô hiệu hóa các chức năng, 183  
 VS2003, 183

## W

WebSVN, 129  
 Windows shell, xi

## **X**

xác thực, 25  
xem các thay đổi, 43  
xem web, 129  
xóa, 76  
xử lý kéo, 24  
xử lý URL, 186  
xuất khẩu, 122  
xuất khẩu thay đổi, 68  
xung đột, 10, 38  
xung đột cây, 38