TortoiseSVN

Klient Subversion dla Windows

Version 1.11

Stefan Küng
Lübbe Onken
Simon Large

TortoiseSVN: Klient Subversion dla Windows: Version 1.11

autorstwa Stefan Küng, Lübbe Onken, i Simon Large
Tłumaczenie: Begina Felicysym (begina.felicysym@wp.eu)

data wydania 2018/09/22 18:28:22 (r28377)

Spis treści

Spis rysunków

Spis tabel

Wstęp

Kontrola wersji jest sztuką zarządzania zmianami informacji. Było to od dawna krytyczne narzędzie dla programistów, którzy zazwyczaj poświęcają czas na drobne zmiany oprogramowania, a następnie je wycofać lub sprawdzić niektóre z tych korekt następnego dnia. Wyobraźcie sobie zespół takich programistów pracujących jednocześnie - a może nawet jednocześnie na tym samym pliku! - teraz widzisz, dlaczego potrzebny jest dobry system do *zarządzania potencjałem chaosu*.

1. Czym jest TortoiseSVN?

TortoiseSVN jest Windowsowym klientem open-surce'owym dla systemu kontroli wersji *Apache™ Subversion®*. Oznacza to, że TortoiseSVN zarządza plikami i folderami w czasie. Pliki są przechowywane w centralnym *repozytorium*. Repozytorium przypomina najbardziej zwykły serwer plików, poza tym zapamiętuje on każdą zmianę dokonaną kiedykolwiek na plikach i katalogach. Pozwala to na przywrócenie starszych wersji plików i zweryfikować historię tego jakie dane i kiedy zostały zmienione, oraz kto je zmienił. Dlatego właśnie wielu ludzi uważa Subversion oraz ogólnie kontrolę wersji pewien rodzaj „wechikułu czasu”.

Niektóre systemy kontroli wersji są także systemami zarządzania konfiguracją oprogramowania (SCM). Systemy te są dostosowane specjalnie do zarządzania drzewem kodu źródłowego i posiadają wiele cech, które są specyficzne dla rozwoju oprogramowania - takie jak natywne rozumienie języków programowania lub dostarczenie narzędzi do tworzenia oprogramowania. Subversion nie jest jednak jednym z tych systemów. Jest ogólny system, który może być używany do zarządzania *każdej* kolekcji plików, w tym kodu źródłowego.

2. Cechy TortoiseSVN

Co sprawia, że TortoiseSVN jest takim dobrym klientem Subversion? Oto krótka lista cech.

Integracja z powłoką

> TortoiseSVN bezproblemowo integruje się z powłoką systemu Windows (np. eksploratorem). Oznacza to pracę z narzędziami, które są już znane. Nie trzeba przechodzić do innych aplikacji za każdym razem, gdy trzeba skorzystać z funkcji kontroli wersji.
>
> I nie jesteście ograniczeni do korzystania z eksploratora Windows, menu kontekstowe TortoiseSVN działa w wielu innych menedżerach plików, a także oknie dialogowym Plik/Otwórz, który jest wspólny dla większości standardowych aplikacji Windows. Należy jednak pamiętać, że TortoiseSVN jest opracowany docelowo jako rozszerzenie eksploratora Windows. Możliwe jest zatem, że w innych aplikacjach integracja nie jest tak pełna i np. nie są pokazywane nakładki ikon.

Nakładki ikon
: Status każdego z wersjonowanych plików i folderów jest wskazywany przez małe nakładki na ikony. W ten sposób można od razu sprawdzić, jaki jest status kopii roboczej.

Graficzny interfejs użytkownika
: Kiedy wyświetla się lista zmian pliku lub folderu, można kliknąć na wersję aby zobaczyć komentarze do tego zatwierdzenia. Można także zobaczyć listę zmienionych plików - wystarczy dwukrotnie kliknąć na plik, aby sprawdzić dokładnie co się zmieniło.

  Okno dialogowe zatwierdzenia zawiera wszystkie elementy, które zostaną dołaczone do zatwierdzenia, a każdy element posiada pole wyboru, dzięki któremu można wybrać, które elementy mają zostać uwzględnione. Niewersjonowane pliki mogą być również wymienione na wypadek, gdy zapomniało się dodać do kontroli wersji nowy plik.

Łatwy dostęp do poleceń Subversion
: Wszystkie polecenia Subversion są dostępne z menu kontekstowego eksploratora. TortoiseSVN dodaje swoje własne drzewo podmenu.

Ponieważ TortoiseSVN jest klientem Subversion, chcielibyśmy pokazać kilka cech samego Subversion:

Wersjonowanie katalogów
: CVS śledzi tylko historię pojedynczych plików, zaś Subversion realizuje „wirtualny” wersjonowany system plików, który śledzi zmiany w czasie całego drzewa katalogów. Wersjonowane są pliki *i* katalogi. W rezultacie mamy prawdziwe polecenia **move** i **copy** dotyczące plików i katalogów po stronie klienta.

Atomowe zatwierdzenia
: Zatwierdzenie albo trafia do repozytorium w całości, albo wcale. Pozwala to programistom na wprowadzanie i zatwierdzanie zmian w logicznych kawałkach.

Wersjonowane metadane
: Każdy plik i katalog posiada dołączony niewidoczny zestaw „atrybutów”. Możesz wymyślić i zapisać dowolną parę klucz/wartość. Atrybuty są wersjonowane w czasie, podobnie jak zawartość pliku.

Wybór warstwy sieci
: Subversion posiada abstrakcyjną koncepcję dostępu do repozytorium, co ułatwia ludziom realizację nowych mechanizmów sieci. „Zaawansowany” serwer sieciowy Subversion jest modułem serwera WWW Apache, który posługuje się wariantem HTTP o nazwie WebDAV/DeltaV. Daje to Subversion dużą przewagę w zakresie stabilności i interoperacyjności oraz oferuje gratis szereg kluczowych funkcji: uwierzytelnianie, autoryzacja, kompresji przesyłania i przeglądania repozytorium, itd.Dostępny jest również mniejszy, samodzielny serwer Subversion. Ten serwer używa niestandardowego protokołu, który może być łatwo tunelowany po ssh.

Spójne przetwarzanie danych
: Subversion określa różnice plików przy użyciu algorytmu binarnego różnicowania, który działa tak samo zarówno dla plików tekstowych (zrozumiały dla człowieka) i binarnych (niezrozumiałych). Oba typy plików są przechowywane w repozytorium w postaci skompresowanej, a różnice przesyłane są w sieci w obu kierunkach.

Efektywne odgałęzianie i etykietowanie
: Koszt odgałęziania i etykietowania nie musi być proporcjonalny do wielkości projektu. Subversion tworzy gałęzie i etykiety przez proste skopiowanie projektu, za pomocą mechanizmu podobnego do twardego dowiązania. Stąd te czynności zabierają jedynie niewielką, stałą ilość czasu i bardzo mało miejsca w repozytorium.

3. Licencja

TortoiseSVN jest projektem Open Source opracowanym na warunkach GNU General Public License (GPL). Można go pobrać za darmo i bezpłatnie używać, zarówno prywatnie, jak i komercyjnie, na dowolnej liczbie komputerów.

Mimo, że większość ludzi po prostu pobiera instalator, macie pełne prawa dostępu do kodu źródłowego tego programu. Możecie go przeglądać przez ten link https://sourceforge.net/p/tortoisesvn/code/HEAD/tree/. Bieżąca linia rozwoju znajduje się pod `/trunk/`, a wydane wersje znajdują się pod `/tags/`.

4. Rozwój

Zarówno TortoiseSVN jak i Subversion są rozwijane przez wspólnotę ludzi, którzy pracują nad tymi projektami. Pochodzą oni z różnych krajów na całym świecie i pracują razem aby stworzyć wspaniałe oprogramowanie.

4.1. Historia TortoiseSVN

W 2002 roku Tim Kemp stwierdził, że Subversion to bardzo dobry system kontroli wersji, ale brakuje mu dobrego klienta GUI. Pomysł na klienta Subversion jako integrację powłoki systemu Windows został zainspirowany

podobnym klientem CVS o nazwie TortoiseCVS. Tim przestudiował kod źródłowy TortoiseCVS i wykorzystał go jako bazę TortoiseSVN. Potem rozpoczął projekt, zarejestrował domenę `tortoisesvn.org` i umieścił w Internecie kod źródłowy.

W tym czasie Stefan Küng poszukiwał dobrego i bezpłatnego systemu kontroli wersji i znalazł Subversion oraz źródła TortoiseSVN. Ponieważ TortoiseSVN nie był jeszcze gotowy do użytku, przystąpił do projektu i zaczął programowanie. Wkrótce przepisał większość istniejącego kodu i dodał polecenia i funkcje, aż do punktu, w którym nie pozostało nic z oryginalnego kodu.

Podczas gdy Subversion stawał się bardziej stabilny, przyciągał coraz więcej użytkowników, którzy zaczęli również używać TortoiseSVN jako klienta Subversion. Baza użytkowników rosła szybko (i wciąż rośnie każdego dnia). Wtedy Lübbe Onken zaoferował pomoc dodając zestaw ładnych ikon i logo TortoiseSVN. Obecnie zajmuje się on witryną i zarządza wieloma tłumaczeniami.

Z czasem wszystkie te systemy kontroli wersji mając własne klienty Tortoise powodowały problemy z nakładkami ikon w Eksploratorze: liczba takich nakładek jest ograniczona i nawet jeden klient Tortoise może z łatwością przekroczyć to ograniczenie. To wtedy Stefan Küng zaimplementował komponent TortoiseOverlays pozwalający wszystkim klientom Tortoise używać tych samych nakładek ikon. Teraz wszystkie otwartoźródłowe klienty Tortoise a nawet niektóre klienty nie-Tortoise używają tego współdzielonego komponentu.

4.2. Podziękowania

Tim Kemp
za rozpoczęcie projektu TortoiseSVN

Stefan Küng
za ciężką pracę by TortoiseSVN stał się tym, czym jest obecnie i prowadzenie projektu

Lübbe Onken
za piękne ikony, logo, tłumaczenie i zarządzanie tłumaczeniami

Simon Large
za utrzymanie dokumentacji

Stefan Fuhrmann
za dziennik bufora i wykres wersji

The Subversion Book
za wspaniały wstęp do Subversion i jej rozdział 2, który tu skopiowaliśmy

The Tigris Style project
za niektóre style, które zostały ponownie użyte w tej dokumentacji

Nasi Współautorzy
za poprawki, raporty o błędach i nowe pomysły oraz pomaganie innym przez odpowiedzi na naszej liście mailingowej

Nasi Darczyńcy
za długie godziny radości z muzyki, którą nam przysłali

5. Przewodnik Czytelnika

Ta książka jest napisana dla ludzi obeznanych komputerowo, którzy chcą używać Subversion do zarządzania danymi, ale wolą korzystać z GUI klienta, a nie klienta linii poleceń. TortoiseSVN stanowi rozszerzenie powłoki Windows i zakłada, że użytkownik jest zaznajomiony z eksploratorem Windows i potrafi z niego korzystać.

To Wstęp wyjaśnia, czym jest TortoiseSVN, daje podstawowe informacje o projekcie TortoiseSVN, społeczności ludzi, którzy nad nim pracują, oraz warunkach licencji na użytkowania i dystrybucji.

Rozdział Rozdział 1, *Pierwsze kroki* wyjaśnia jak instalować TortoiseSVN na komputerze i jak zacząć z niego korzystać.

W Rozdział 2, *Podstawowe pojęcia kontroli wersji* dajemy krótki wstęp do systemu kontroli wersji *Subversion*, który stanowi podstawę TortoiseSVN. Został on zapożyczony z dokumentacji projektu Subversion i wyjaśnia różne podejścia do kontroli wersji oraz jak działa Subversion.

Rozdział Rozdział 3, *Repozytorium* wyjaśnia jak skonfigurować repozytorium lokalne, co jest przydatne do testowania Subversion i TortoiseSVN na pojedynczym PC. Wyjaśnia on trochę temat administracji repozytorium co jest również istotne dla repozytoriów umieszczonych na serwerze.

Sekcja Rozdział 4, *Instrukcja codziennej obsługi* jest najważniejsza gdyż wyjaśnia podstawowe cechy TortoiseSVN oraz sposób ich użycia. Jest ona w formie kursu, zaczynając od uaktualnienia kopii roboczej, poprzez jej modyfikację, zatwierdzenie zmian, itp. Następnie przechodzi do bardziej zaawansowanych zagadnień.

Rozdział 6, *Program SubWCRev* jest oddzielnym programem zawartym w TortoiseSVN który może odczytać informacje z kopii roboczej i zapisać je w pliku. Jest to przydatne do zawarcia informacji o budowie w projektach.

Sekcja Dodatek B, *Jak to zrobić...* odpowiada na podstawowe pytania o wykonaniu zadań nie opisanych gdzie indziej.

Sekcja Dodatek D, *Automatyzacja TortoiseSVN* pokazuje, jak dialogi graficznego interfejsu użytkownika TortoiseSVN mogą być wywoływane z linii poleceń. Jest to przydatne dla skryptów, które jednak wymagają interakcji z użytkownikiem.

Rozdział Dodatek E, *Odsyłacze interfejsu wiersza poleceń* pokazuje korelację między poleceniami TortoiseSVN oraz ich odpowiednikami w kliencie linii poleceniach Subversion `svn.exe`.

6. Terminologia używana w dokumencie

Aby łatwiej czytać dokumentację, nazwy wszystkich ekranów i menu z TortoiseSVN są zaznaczone inną czcionką.Dialog Dziennika na przykład.

Wybór menu jest wskazany strzałką. TortoiseSVN → Pokaż dziennik oznacza: wybierz *Pokaż dziennik* z menu kontekstowego *TortoiseSVN*.

Jeżeli lokalne menu kontekstowe pojawia się w jednym z okien dialogowych TortoiseSVN, pokaże się tak: Menu kontekstowe → Zapisz jako...

Przyciski interfejsu użytkownika są oznaczone w ten sposób: Należy nacisnąć OK, aby kontynuować.

Akcje użytkownika są oznaczone za pomocą pogrubionej czcionki. **Alt+A**: należy nacisnąć kalwisz **Alt** na klawiaturze i trzymając go nacisnąć jednocześnie **A**. Przeciągnięcie prawym przyciskiem myszy: wcisnąć prawy przycisk myszy i trzymając go w dół *przeciągnij* elementy do nowej lokalizacji.

Wyjście systemu i klawiatury oznaczone jest `inną` czcionką.

Ważne

Ważne informacje są oznaczone ikoną.

Podpowiedź

Porady, które ułatwią Wam życie.

Ostrzeżenie

Miejsca, gdzie trzeba uważać na to co robicie.

Ostrzeżenie

Gdzie konieczne jest zachowanie szczególnej ostrożności. Może nastąpić uszkodzenie danych lub inne nieprzyjemne rzeczy, jeśli te ostrzeżenia zostaną zignorowane.

Rozdział 1. Pierwsze kroki

Ta sekcja jest skierowana do ludzi, którzy chcieliby dowiedzieć się, co to jest TortoiseSVN i wypróbować program. Wyjaśnia, jak zainstalować TortoiseSVN i utworzyć lokalne repozytorium, a następnie prowadzi przez najczęściej używane operacje.

1.1. Instalacja TortoiseSVN

1.1.1. Wymagania systemowe

TortoiseSVN działa w systemie Windows Vista lub wyższym i jest dostępny w wersji 32-bitowej i 64-bitowej. Instalator dla systemu Windows 64-bitowego zawiera także 32-bitowe składniki rozszerzenia. Oznacza to, że nie trzeba instalować 32-bitowej wersji oddzielnie, aby uzyskać menu kontekstowego TortoiseSVN i nakładek w 32-bitowych aplikacjach.

Wsparcie dla Windows 98, Windows ME i Windows NT4 zostało zakończone w wersji 1.2.0, a Windows 2000 i XP do dodatku SP2 skończyło się w 1.7.0. Wsparcia dla Windows XP z SP3 zaprzestano w 1.9.0. Nadal można pobrać i instalować starsze wersje jeśli jest to konieczne.

1.1.2. Instalacja

TortoiseSVN jest wyposażony w łatwy w obsłudze instalator. Kliknij dwukrotnie na pliku instalatora i postępuj zgodnie z instrukcjami. Instalator zadba o resztę. Nie zapomnij ponownie uruchomić komputer po instalacji.

Ważne

Musicie mieć uprawnienia Administratora aby zainstalować TortoiseSVN. Instalator poprosi was o poświadczenia Administratora jeśli będzie to konieczne.

Są dostępne pakiety językowe, które tłumaczą interfejs użytkownika TortoiseSVN na wiele różnych języków. Proszę sprawdzić Dodatek G, *Pakiety językowe i sprawdzenia pisowni* by uzyskać więcej informacji na temat ich instalacji.

Jeśli napotkacie problemy podczas lub po instalacji TortoiseSVN, zapoznajcie się z naszym FAQ online na *https://tortoisesvn.net/faq.html* [http://tortoisesvn.net/faq.html].

1.2. Podstawowe pojęcia

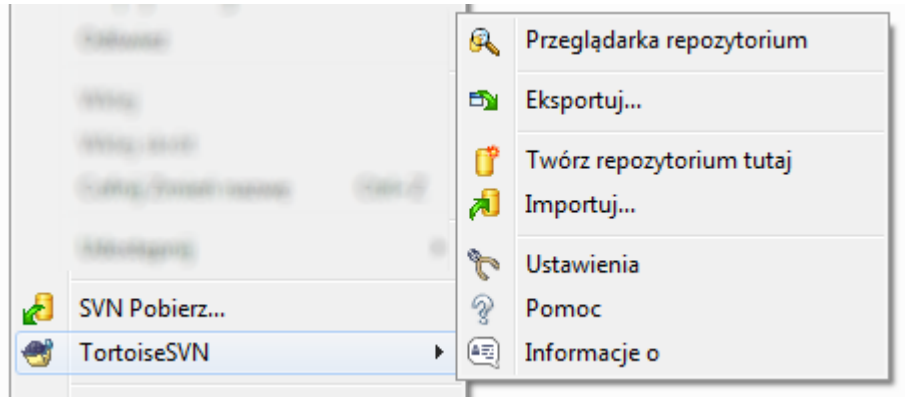
Zanim utkniemy podczas pracy z prawdziwymi plikami, ważne jest uzyskanie ogólnej wiedzy o tym jak działa Subversion i poznanie używanych terminów.

Repozytorium
: Subversion używa centralnej bazy danych, która zawiera wszystkie pliki poddane kontroli wersji z ich pełną historią. Baza ta jest określana jako *repozytorium*. Repozytorium trzymane jest zwykle na serwerze plików z działającym programem serwera Subversion, który dostarcza zawartość klientom Subversion (jak TortoiseSVN) na życzenie. Jeśli robicie kopię zapasową jednej rzeczy, wykonujcie ją na repozytorium, ponieważ jest to ostateczna i podstawowa kopia kopia wszystkich waszych danych.

Kopia robocza
: To tu wykonuje się prawdziwą pracę. Każdy projektant ma swoją własną kopię roboczą, czasami nazywaną piaskownicą, na swoim komputerze lokalnym. Możecie ściągnąć najnowszą wersję z repozytorium, pracować nad nią lokalnie bez wpływu na innych, a kiedy jesteście zadowoleni z wprowadzonych zmian, zatwierdźcie je z powrotem do repozytorium.

  Kopia robocza Subversion nie zawiera historii projektu, ale przechowuje kopie plików w stanie zapisanym w repozytorium przed rozpoczęciem wprowadzania zmian. Oznacza to, że łatwo jest sprawdzić, jakie dokładnie wprowadzono zmiany.

Należy również wiedzieć, gdzie szukać TortoiseSVN, bo niewiele znajduje się w menu Start. To dlatego, że TortoiseSVN jest rozszerzeniem powłoki, zatem na początek należy uruchomić Eksploratora Windows. Kliknięcie prawym przyciskiem myszy na folder w Eksploratorze i powinno ukazać nowe wpisy w menu kontekstowym:



Rysunek 1.1. Menu TortoiseSVN dla niewersjonowanych folderów

1.3. Przejście do jazdy próbnej

Ten rozdział pokazuje, jak wypróbować niektóre z najczęściej używanych funkcji na małym repozytorium testowym. Oczywiście nie wyjaśnia wszystkiego - to tylko ostatecznie Skrócony Przewodnik Startowy. Kiedy już go przejdziecie należy poświęcić trochę czasu i przeczytać resztę tej instrukcji obsługi, która objaśnia znacznie rzeczy bardziej szczegółowo. Wyjaśnia również dokładniej postawienie właściwego serwera Subversion.

1.3.1. Utworzenie repozytorium

Dla prawdziwego projektu należy ustawić repozytorium w bezpiecznym miejscu z kontrolującym go serwerem Subversion. Dla celów tego poradnika będziemy używać funkcji lokalnego repozytorium Subversion, które pozwala na bezpośredni dostęp do repozytorium tworzonego na dysku twardym bez żadnej konieczności korzystania z serwera.

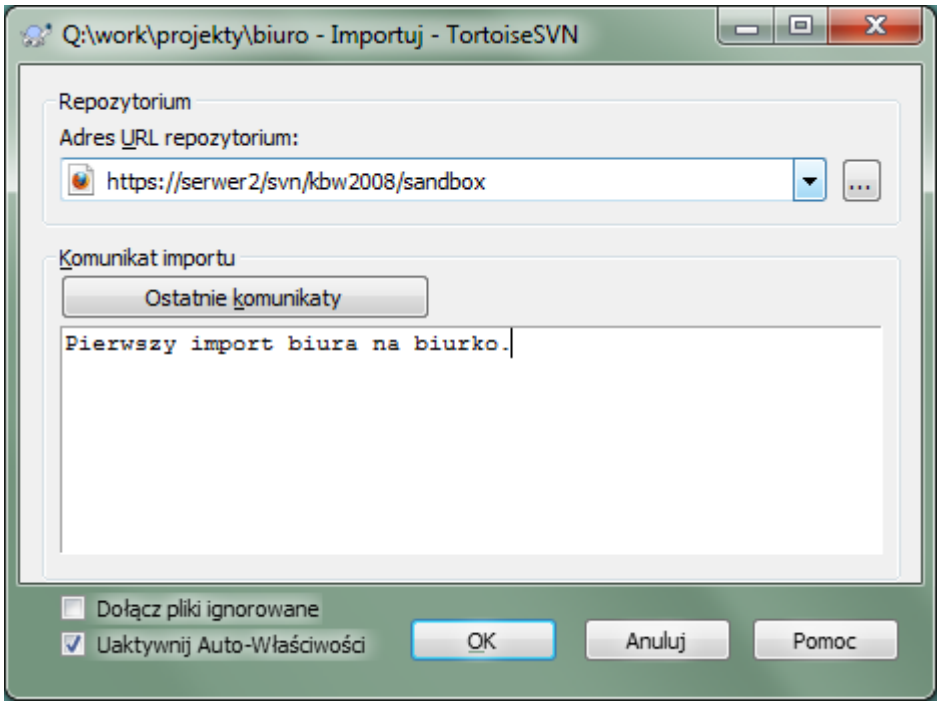
Na początku należy utworzyć nowy, pusty folder na komputerze lokalnym. Może powstać gdziekolwiek, ale na potrzeby naszego samouczka będziemy nazywać go `C:\svn_repos`. Teraz trzeba kliknąć prawym przyciskiem myszy na tym nowym folderze i w menu kontekstowym wybrać TortoiseSVN → Twórz repozytorium tutaj.... Repozytorium zostaje utworzone wewnątrz tego folderu i jest gotowe do pracy. Utworzymy również domyślną strukturę katalogów wciskając przycisk Utwórz strukturę folderów.

Ważne

Funkcja lokalnego repozytorium jest bardzo przydatna do badania i oceny, a o ile nie pracuje się jako jedyny programista na jednym komputerze PC, należy zawsze używać odpowiedniego serwera Subversion. Kuszące jest w małej firmie, by uniknąć pracy nad konfigurowaniem serwera i mieć prosty dostęp do repozytorium przez udział sieciowy. Nie róbcie tego. Spowoduje to utratę danych. Przeczytajcie Sekcja 3.1.4, „Dostęp do repozytorium na udziale sieciowym” by dowiedzieć się, dlaczego jest to zły pomysł i jak skonfigurować serwer.

1.3.2. Importowanie projektu

Posiadamy teraz repozytorium, ale póki co jest ono zupełnie puste. Załóżmy, że mam w `C:\Projects\Widget1` zestaw plików, które chcę dodać. Przejdźmy do katalogu `Widget1` w Eksploratorze i kliknijmy na nim prawym przyciskiem myszy. Wybierzmy teraz TortoiseSVN → Importuj..., co wyświetli okno dialogowe



Rysunek 1.2. Dialog importu

Repozytorium Subversion jest określone przez adres URL, który pozwala nam wskazać repozytorium z dowolnego miejsca w Internecie. W tym przypadku musimy wskazać nasze własne, lokalne repozytorium, które posiada adres URL `file:///c:/svn_repos/trunk` i do którego dodamy własny projekt o nazwie `Widget1`. Zauważcie, że mamy tu 3 ukośniki po `file:` i że w całej ścieżce używane są ukośniki a nie backslashe.

Inną ważną cechą tego okna jest pole Komunikat importu, które pozwala wprowadzić komunikat opisujący to, co robicie. Kiedy dochodzi do weryfikacji historii projektu, te opisy zmian stanowią cenną pomoc, jakie zmiany zostały dokonane i dlaczego. W tym przypadku możemy powiedzieć, coś prostego, jak „Import projektu Widget1”. Kliknijcie na OK i folder zostanie dodany do repozytorium.

1.3.3. Uaktualnienie kopii roboczej

Teraz gdy już mamy projekt w naszym repozytorium, potrzebujemy utworzyć kopię roboczą do codziennej pracy. Zauważcie, że sam import folderu nie zmienił go automatycznie w kopię roboczą. Terminem używanym przez Subversion dla utworzenia świeżej kopii roboczej jest `Pobranie`. Przejdźmy do pobrania folderu Widget1 z naszego repozytorium do folderu rozwojowego na PCcie, który nazwiemy `C:\Projects\Widget1-Dev`. Utwórzcie ten folder, po czym kliknijcie prawym przyciskiem myszy i wybierzcie TortoiseSVN → Pobierz.... Następnie wpiszcie adres URL do pobrania, w tym przypadku `file:///c:/svn_repos/trunk/Widget1` i wciśnijcie OK. Nasz folder rozwojowy zostaje wtedy zapełniony plikami z repozytorium.

Ważne

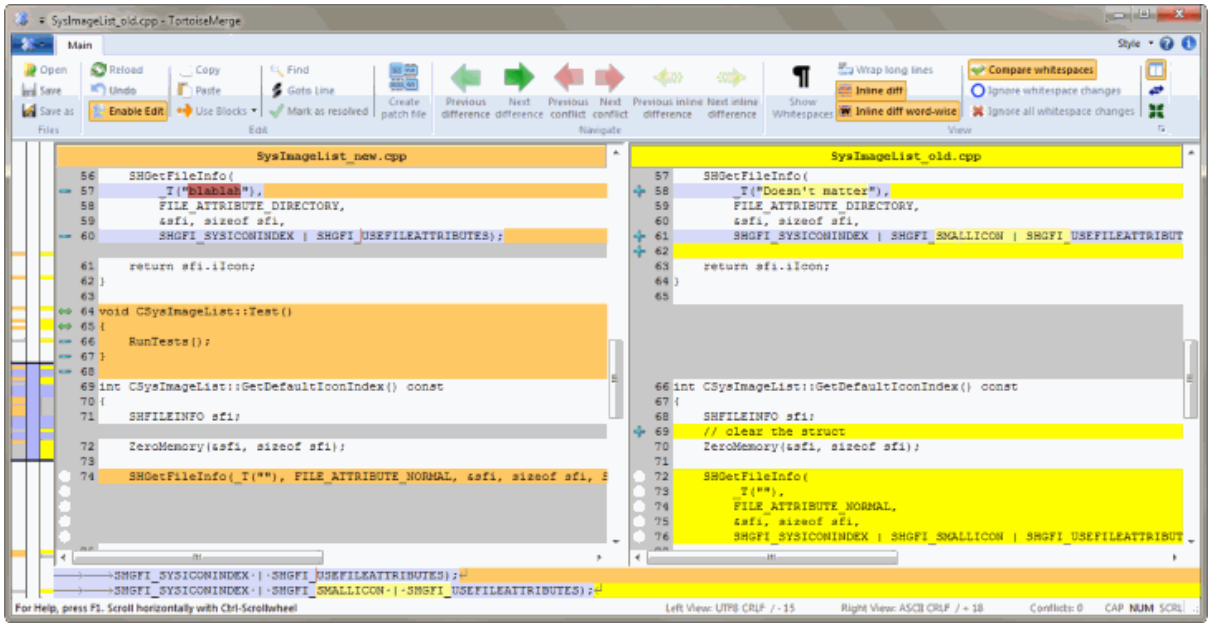
W ustawieniach domyślnych pozycja menu pobierz nie znajduje się w podmenu TortoiseSVN ale pokazywana jest w głównym menu eksploratora. Polecenia TortoiseSVN nie umieszczone w podmenu posiadają przedrostek `SVN`: SVN Pobierz...

Można zauważyć, że wygląd tego folderu różni się od naszego oryginalnego katalogu. Każdy plik ma zielony znacznik wyboru w lewym dolnym rogu. Są to ikony statusu TortoiseSVN, które są obecne tylko w kopii roboczej. Zielony stan wskazuje, że plik nie różni się od wersji w repozytorium.

1.3.4. Wprowadzanie zmian

Czas zabrać się do pracy. W `Widget1-Dev` zaczynamy edycję plików - powiedzmy, że wprowadzimy zmiany w `Widget1.c` i `ReadMe.txt`. Zauważmy, że nakładki ikon dla tych plików zmieniły się teraz na czerwone wskazując, że zmiany zostały wprowadzone lokalnie.

Ale jakie są zmiany? Należy kliknąć prawym przyciskiem myszy na jednym z zmienionych plików i wybrać TortoiseSVN → Porównaj. Uruchamia się porównywarka plików TortoiseSVN, pokazując, które dokładnie linie uległy zmianie.



Rysunek 1.3. Przeglądarka różnic pliku

OK, więc jesteśmy zadowoleni z tych zmian, zaktualizujmy repozytorium. Działanie to jest określane dalej jako`Zatwierdzenie` zmian. Trzeba kliknąć prawym przyciskiem myszy na folder `Widget1-Dev` i wybrać TortoiseSVN → Zatwierdź. Okno dialogowe zatwierdzenia zawiera listę zmienionych plików, z polem wyboru przy każdym z nich. Możecie wybrać tylko część z plików, ale w tym przypadku mamy zamiar zatwierdzić zmiany do obu plików. Wpiszcie komunikat opisujący, co się zmieniło i kliknijcie OK. Okno dialogowe postępu pokazuje, że pliki są przesyłane do repozytorium i gotowe.

1.3.5. Dodanie kolejnych plików

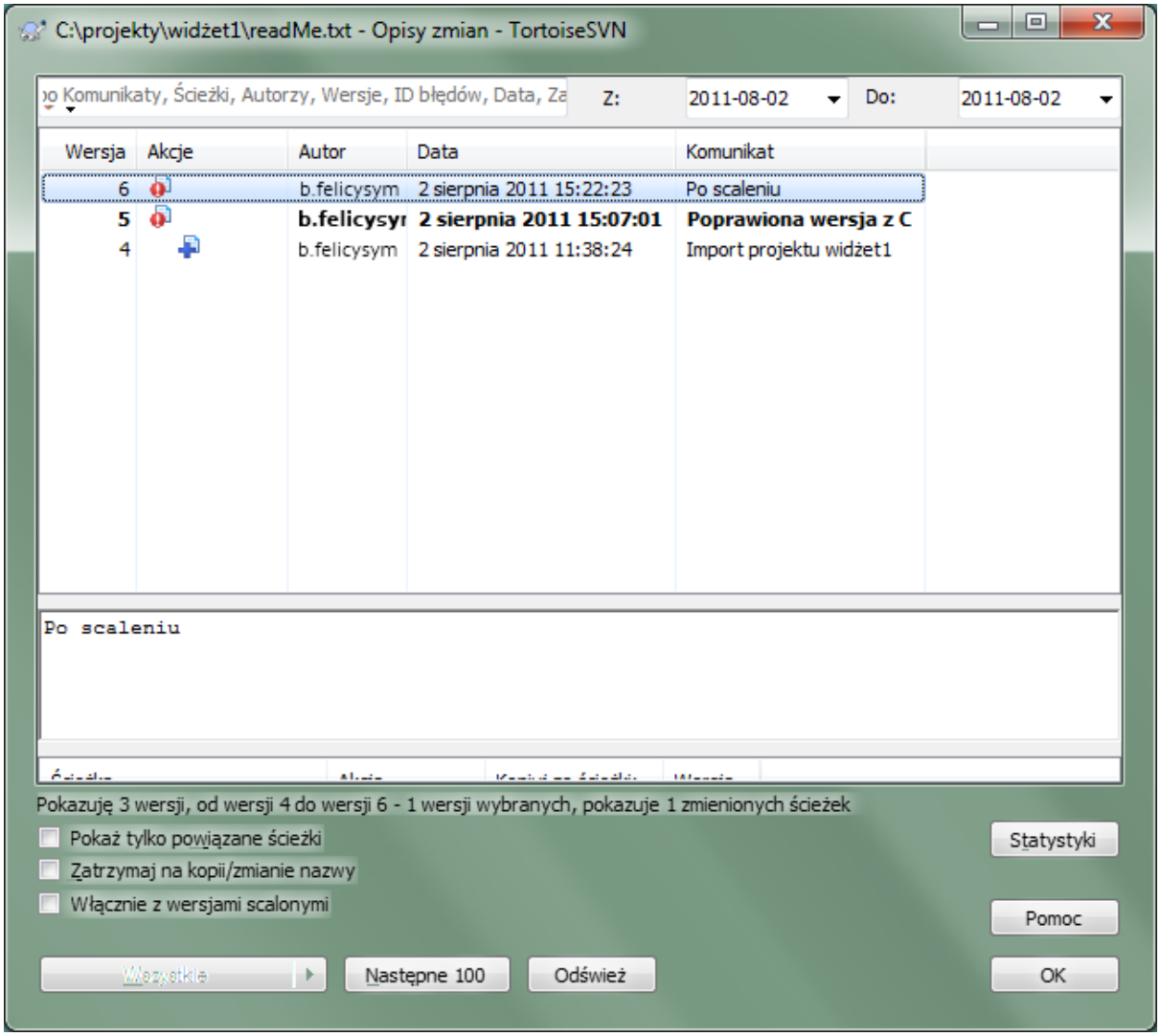
Wraz z rozwojem projektu trzeba będzie dodawać nowe pliki - powiedzmy, że dodacie kilka nowych funkcji w `Extras.c` i dodacie odwołanie w istniejącym `Makefile`. Trzeba kliknąć prawym przyciskiem myszy na folder i TortoiseSVN → Dodaj. Okno dialogowe dodania pokazuje teraz wszystkie pliki bez informacji o wersji i możecie wybrać te, które chcecie dodać. Innym sposobem dodawania plików jest kliknięcie prawym przyciskiem myszy na plik, i wybór TortoiseSVN → Dodaj.

Teraz po przejściu do zatwierdzenia folderu, nowy plik pokazuje się jako *Dodano* a istniejący plik jako *Zmodyfikowano*. Pamiętajcie, że możecie kliknąć dwukrotnie na zmodyfikowanym pliku, aby sprawdzić dokładnie, jakie zmiany zostały dokonane.

1.3.6. Przeglądanie historii projektu

Jedną z najbardziej przydatnych funkcji TortoiseSVN jest okno dziennika. Pokazuje ono listę wszystkich zatwierdzeń dokonanych na pliku lub folderze oraz wyświetla szczegółowe wiadomości odnośnie zatwierdzeń,

które wprowadziliście (*wpisaliście* komunikat zatwierdzenia zgodnie z sugestiami? Jeśli nie, teraz widzicie dlaczego to takie ważne).



Rysunek 1.4. Dialog dziennika

OK, oszukałem tu trochę i pokazałem zrzut ekranu z repozytorium TortoiseSVN.

W górnej części okna wyświetlana jest lista zatwierdzonych wersji wraz z początkiem opisu zatwierdzenia. Jeśli wybierze się jedną z tych wersji, środkowe okienko pokaże nam pełny opis zmian dla tej wersji a w dolnym panelu pojawi się lista zmienionych plików i folderów.

Każde z tych okienek ma własne menu kontekstowe, które oferuje wiele więcej sposobów wykorzystania tych informacji. W dolnym okienku można kliknąć dwukrotnie na pliku, aby sprawdzić dokładnie, jakie zmiany zostały dokonane w tej wersji. Przejdź do Sekcja 4.10, „Okno dialogowe dziennika wersji”, aby poznać całą historię.

1.3.7. Wycofanie zmian

Jedną z cech wszystkich systemów kontroli wersji jest to, że pozwala cofnąć zmiany wprowadzone wcześniej. Jak można się spodziewać, w TortoiseSVN działanie to jest łatwo dostępne.

Jeśli chcecie pozbyć się zmian, których jeszcze nie zatwierdziliście i przywrócić plik do stanu przed rozpoczęciem edycji TortoiseSVN → Wycofaj jest twoim przyjacielem. Polecenie odrzuca zmiany (do Kosza, na wszelki wypadek) i przywraca zatwierdzoną wersję z początku zmian. Jeśli chcecie pozbyć się tylko niektórych zmian, możecie użyć TortoiseMerge aby zobaczyć różnice i wybiórczo przywrócić zmienione wiersze.

Jeśli chcecie cofnąć skutki poszczególnych wersji, zacznijcie od okna dialogowego dziennika i odszukajcie niewłaściwą wersję. Teraz trzeba wybrać Menu kontekstowe → Wycofaj zmiany z tej wersji a zmiany te zostaną cofnięte.

1.4. Przejście dalej ...

Przewodnik ten dał wam bardzo szybką wycieczkę po niektórych najważniejszych i najbardziej przydatnych funkcjach TortoiseSVN, ale oczywiście znacznie więcej jest tych, które nie zostały jeszcze odkryte. Zalecamy spędzić trochę czasu, aby przeczytać resztę tego podręcznika, szczególnie Rozdział 4, *Instrukcja codziennej obsługi*, który rzuca o wiele więcej światła wykonywane codziennie operacje.

Zadaliśmy sobie wiele trudu, by upewnić się, że jest zarówno pouczający i łatwy do przeczytania, ale zdajemy sobie sprawę, że jest tego dużo! Nie spieszcie się i nie bójcie próbować rzeczy na repozytorium testowym w miarę czytania. Najlepszym sposobem nauczenia się czegoś jest przez jego używanie.

Rozdział 2. Podstawowe pojęcia kontroli wersji

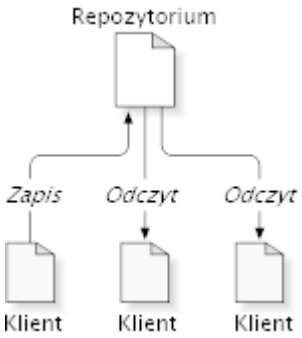
Ten rozdział to nieco zmieniona wersja takiego samego rozdziału w książce Subversion. Elektroniczna wersja książki Subversion dostępna jest tutaj: http://svnbook.red-bean.com/.

Ten rozdział jest krótkim, zdawkowym wprowadzeniem do Subversion. Jeśli jesteście świeży w kontroli wersji, rozdział ten jest zdecydowanie dla Was. Zaczniemy od omówienia ogólnej koncepcji kontroli wersji, rozpracujemy charakterystyczne idee kryjące się za Subversion i pokażemy kilka prostych przykładów z użycia systemu.

Choć przykłady w tym rozdziale pokazują ludzi współdzielących zbiory z kodem źródłowym programu, należy pamiętać, że Subversion może zarządzać jakąkolwiek kolekcją plików - narzędzie nie jest ograniczone do wspomagania programistów.

2.1. Repozytorium

Subversion to scentralizowany system wymiany informacji. U jego podstaw znajduje się *repozytorium*, które jest centralnym magazynem danych. Repozytorium przechowuje informacje w formie *drzewa plików* - typowej hierarchia plików i katalogów. Dowolna liczba *klientów* łączy się z repozytorium, a następnie odczytuje lub zapisuje te pliki. Zapisując dane, klient sprawia, że informacje stają się dostępne dla innych; przez odczyt danych klient otrzymuje informacje od innych.



Rysunek 2.1. Typowy system klient/serwer

Więc dlaczego ma to być interesujące? Na razie brzmi to jak definicja typowego serwera plików. I rzeczywiście, repozytorium *jest* swego rodzaju serwerem plików, ale nie takim, z jakim się zwykle stykasz. Tym co sprawia, że repozytorium SVN specjalne jest, że *pamięta każdą zmianę*, jaką kiedykolwiek w nim zapisano: każdą zmianę każdego pliku, a nawet zmiany w samym drzewie katalogów, takie jak dodanie, usunięcie i przegrupowanie plików i katalogów.

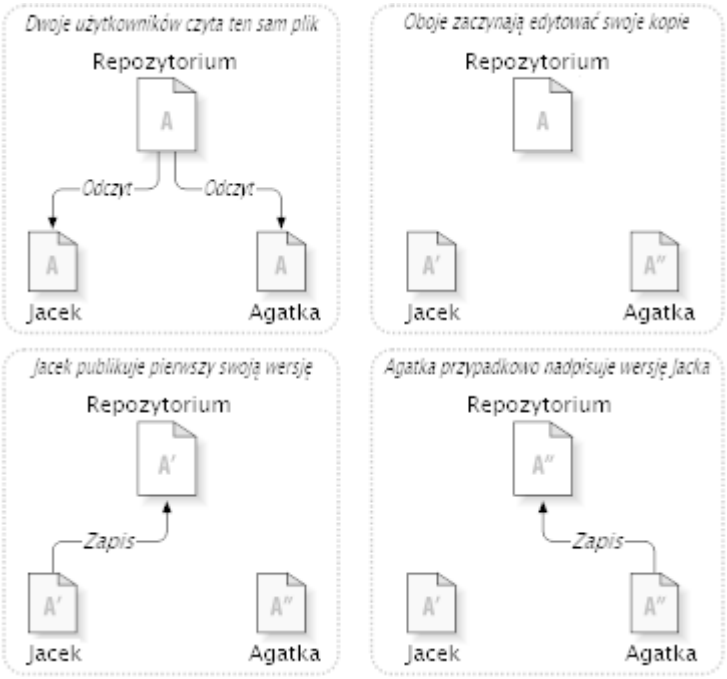
Gdy klient odczytuje dane z repozytorium, to zwykle widzi tylko ostatnią wersję drzewa katalogów. Ale klient ma również możliwość przeglądania *poprzednich* stanów systemu plików. Na przykład, klient może zadać historyczne pytania typu „ co zawierał ten katalog w ostatnią środę?” lub „ kto był ostatnią osobą, zmieniającą ten plik, i jakie zmiany zrobił? ”. To są rodzaje pytań, które znajdują się w centrum każdego *systemu kontroli wersji*: tj systemu, który jest przeznaczony do rejestracji i śledzenia zmian danych w czasie.

2.2. Modele wersjonowania

Wszystkie systemy kontroli wersji muszą rozwiązać ten sam podstawowy problem: w jaki sposób system umożliwia użytkownikom dzielenie się informacjami, a jednocześnie zapobiec przypadkowemu podstawianiu nogi jednym przez drugie? Zbyt łatwo o to, by jeden z użytkowników przypadkowo nadpisał zmiany innego w repozytorium.

2.2.1. Problem ze współdzieleniem plików

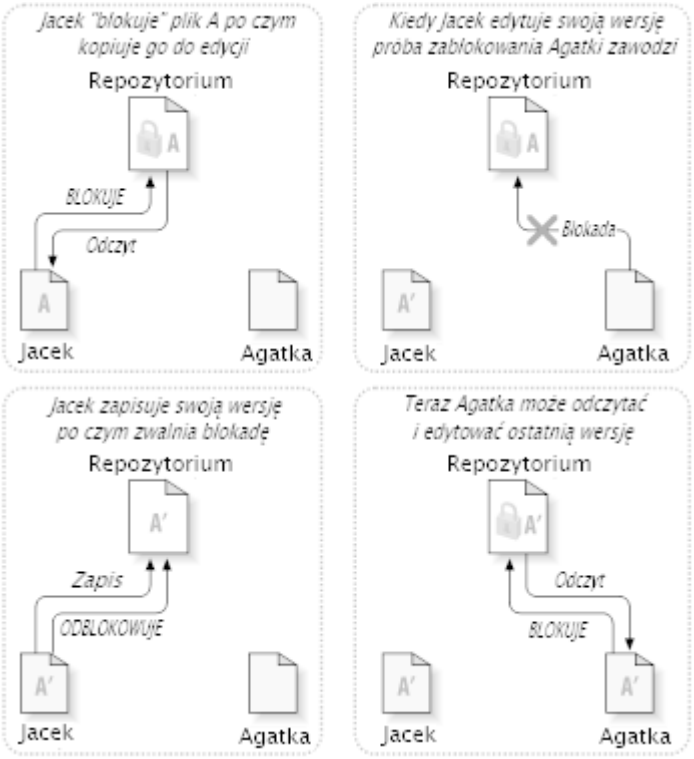
Rozważmy następujący scenariusz: załóżmy, że mamy dwoje współpracowników, Jacka i Agatkę. Każde z nich decyduje się na edycję tych samych plików w repozytorium w tym samym czasie. Jeśli Jacek zapisze jako pierwszy swoje zmiany do repozytorium, jest możliwe że potem (kilka chwil później) Agatka może przypadkowo nadpisać je własną nową wersją pliku. Chociaż wersja pliku Jacka nie została utracona na zawsze (ponieważ system pamięta każdą zmianę), wszelkie wprowadzone zmiany jakie wykonał *Jacek* nie są obecne w nowej wersji pliku Agatki, ponieważ nigdy nie widziała zmian Jacka by od nich zacząć. Praca Jacka jest nadal faktycznie stracona - a przynajmniej nie ma jej w najnowszej wersji pliku - i to prawdopodobnie przez przypadek. Jest to na pewno sytuacja, której chcemy uniknąć!



Rysunek 2.2. Problem, którego należy uniknąć

2.2.2. Rozwiązanie blokada-modyfikacja-odblokowanie

Wiele systemów kontroli wersji korzysta z modelu *blokada-modyfikacja-odblokowanie* by uporać się z problemem, co jest bardzo prostym rozwiązaniem. W takim systemie repozytorium pozwala tylko jednej osobie na raz zmieniać plik. Najpierw Jacek musi *zablokować* plik, zanim będzie mógł zacząć dokonywać zmian. Blokowanie pliku jest bardzo podobne do wypożyczenia książki z biblioteki, jeśli Jacek zablokował plik, Agatka nie może już dokonać żadnych zmian. Jeśli stara się ona zablokować plik, repozytorium odrzuci żądanie. Wszystko co może zrobić, to odczytać plik i czekać na Jacka, aż zakończy swoje zmiany i zwolni blokadę. Po odblokowaniu pliku przez Jacka, jego kolejka się kończy, a Agatka może z kolei zablokować i edytować.



Rysunek 2.3. Rozwiązanie blokada-modyfikacja-odblokowanie

Problemem z modelem blokada-modyfikacja-odblokowanie jest to, że jest on trochę zbyt restrykcyjny, co często staje się przeszkodą dla użytkowników:

- *Blokowanie może powodować problemy administracyjne.* Czasami Jacek może zablokować plik, a następnie o nim zapomnieć. Tymczasem Agatka, wciąż czeka by edytować plik i ma związane ręce. A potem Jacek idzie na urlop. Teraz Agatka musi znaleźć administratora by zwolnić blokadę Jacka. Sytuacja jest przyczyną długich niepotrzebnych opóźnień i straty czasu.

- *Blokowanie może to doprowadzić do zbędnej serializacji.* Co zrobić, jeśli Jacek edytuje początek pliku tekstowego, a Agatka chce tylko edytować koniec tego samego pliku? Zmiany te wcale się nie pokrywają. Mogliby łatwo edytować plik jednocześnie, nie czyniąc wielkiej szkody zakładając, że zmiany były właściwie scalone. Nie ma potrzeby, by czekać na swoją kolej w tej sytuacji.

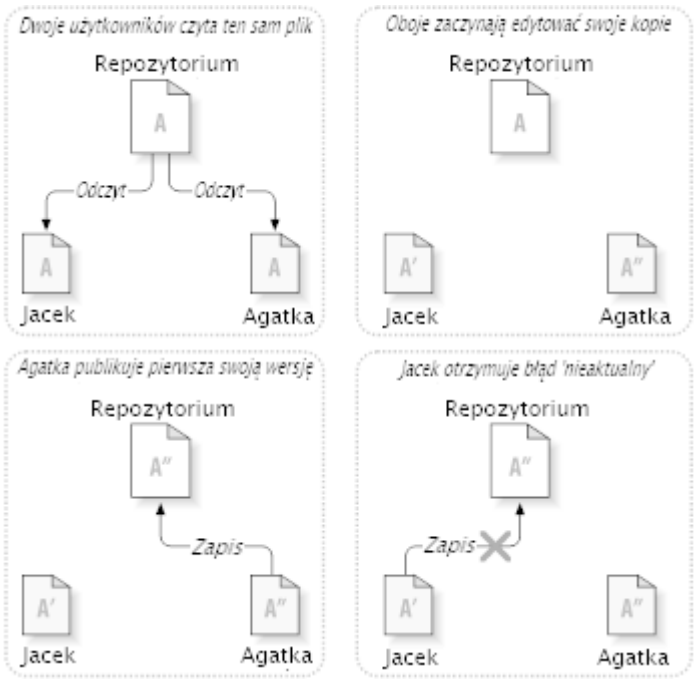
- *Blokowanie może stworzyć fałszywe poczucie bezpieczeństwa.* Załóżmy, że Jacek blokuje i edytuje plik A, a jednocześnie Agatka blokuje i edytuje plik B. Załóżmy też, że A i B zależą od siebie nawzajem, a zmiany dokonane w każdym są semantycznie sprzeczne. Nagle A i B nie są już zgodne ze sobą. System blokowania nie był w stanie zapobiec temu problemowi - a jednak w jakiś sposób daje złudne poczucie bezpieczeństwa. Łatwo Jackowi i Agatce wyobrazić sobie, że blokując pliki, każde zaczyna bezpieczne, odizolowane zadanie, co hamuje omawianie ich niezgodnych zmian na wczesnym etapie.

2.2.3. Rozwiązanie kopia-modyfikacja-scalanie

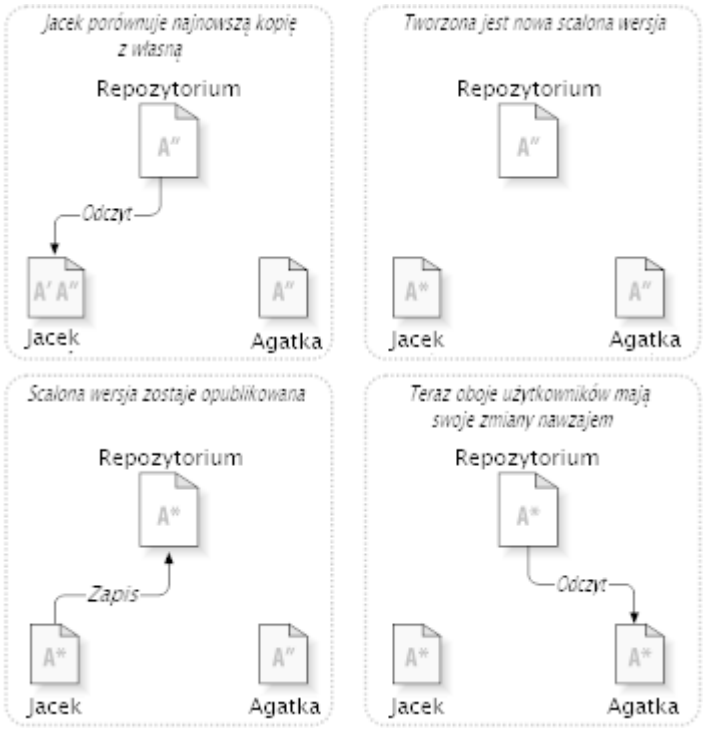
Subversion, CVS i inne systemy kontroli wersji używają modelu *kopia-modyfikacja-scalanie* jako alternatywy blokowania. W tym modelu klient każdego użytkownika czyta repozytorium i tworzy osobistą *kopię roboczą* pliku lub projektu. Użytkownicy mogą następnie pracować równolegle, zmieniając swoje prywatne kopie. Na koniec prywatne kopie są scalone do nowej, finalnej wersji. System kontroli wersji często pomaga przy scaleniu, ale na koniec to człowiek jest odpowiedzialny by wszystko przebiegło poprawnie.

Oto przykład. Powiedzmy, że oboje Jacek i Agatka tworzą kopie robocze tego samego projektu, pobrane z repozytorium. Pracują jednocześnie, i wprowadzają zmiany w tym samym pliku `A` w swoich kopiach. Agatka zapisuje swoje zmiany do repozytorium pierwsza. Kiedy Jacek później próbuje zapisać swoje zmiany, repozytorium informuje go, że jego plik A jest *nieaktualny*. Innymi słowy, że plik w repozytorium został w jakiś

sposób zmieniony odkąd ostatni Jacek skopiował go ostatnio. Tak więc Jacek prosi swojego klienta by *scalić* nowe zmiany z repozytorium do swojej kopii roboczej pliku A. Jest możliwe, że zmiany Agatki nie pokrywają się z jego własnymi i gdy już oba zestawy zmian zostaną zintegrowane, będzie mógł zapisać swoją kopię roboczą z powrotem do repozytorium.



Rysunek 2.4. Rozwiązanie kopia-modyfikacja-scalanie



Rysunek 2.5. ...Kopia-modyfikacja-scalanie ciąg dalszy

Ale co, jeśli zmiany Agatki *pokrywają się* ze zmianami Jacka? Co wtedy? Taka sytuacja nazywana jest *konfliktem* i zazwyczaj nie stanowi wielkiego problemu. Kiedy Jacek prosi swojego klienta o scalenie najnowszych zmian

w swojej kopii roboczej repozytorium, jego kopia pliku A jest w jakiś sposób oznaczona jako pozostająca w stanie konfliktu: będzie on w stanie zobaczyć oba zestawy zmian powodujących konflikt i ręcznie wybrać między nimi . Należy pamiętać, że oprogramowanie nie może automatycznie rozwiązać konfliktu, tylko ludzie są zdolni do zrozumienia i dokonania niezbędnych inteligentnych wyborów. A gdy już Jacek ręcznie rozwiąże nakładające się zmiany (być może omawiając konflikt z Agatką!), może spokojnie zapisać scalony plik z powrotem do repozytorium.

Model kopia-modyfikacja-scalanie może wydawać się nieco chaotyczny, ale w praktyce działa on bardzo sprawnie. Użytkownicy mogą pracować równolegle, nie czekając na siebie. Gdy pracują na tych samych plikach, okazuje się, że większość ich jednocześnie wprowadzonych zmian nie pokrywa się wcale, a konflikty są rzadkie. Przy tym czas potrzebny do rozwiązywania konfliktów jest znacznie krótszy niż stracony przez system blokowania.

W końcu wszystko sprowadza się do jednego zasadniczego czynnika: komunikacja między użytkownikami. Gdy użytkownicy komunikują się słabo, narastają konflikty zarówno składniowe jak i semantyczne. Żaden system nie może zmusić użytkowników do doskonałego komunikowania się, i żaden system nie jest w stanie wykryć konfliktów semantycznych. Nie warto usypiać czujności fałszywą obietnicą, że system blokowania będzie jakoś zapobiegać konfliktom. W praktyce blokada wydaje się hamować wydajność bardziej niż cokolwiek innego.

Istnieje jedna znana sytuacja, w której model blokada-modyfikacja-odblokowanie wypada lepiej - to tam, gdzie występują pliki niemożliwe do scalenia. Na przykład, jeśli repozytorium zawiera kilka obrazów graficznych, i dwóch ludzi zmienia obraz w tym samym czasie, nie ma sposobu aby te zmiany mogły zostać połączone razem. Albo Jacek albo Agatka straci swoje zmiany.

2.2.4. Co robi Subversion?

Subversion używa domyślnie rozwiązania kopia-modyfikacja-scalanie, i w wielu przypadkach jest to wszystko co będzie ci kiedykolwiek potrzebne. Jednak od wersji 1.2 Subversion obsługuje również blokowanie plików, więc jeśli pracujesz z plikami niemożliwymi do scalenia lub jeśli po prostu kierownictwo zmusza was do stosowania polityki blokowania, Subversion nadal dostarczy potrzebnych funkcji.

2.3. Subversion w akcji

2.3.1. Kopie robocze

Przeczytałeś już o kopiach roboczych, teraz pokażemy, w jaki sposób klient Subversion tworzy je i wykorzystuje.

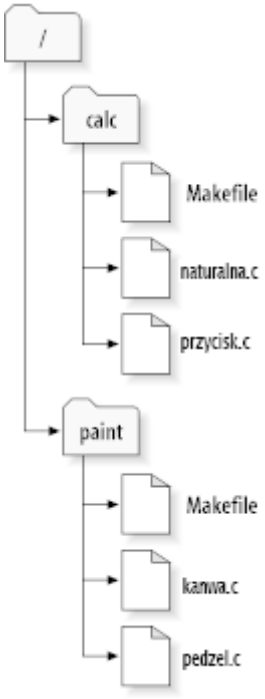
Kopia robocza Subversion jest zwykłym drzewem katalogów w systemie lokalnym, zawierającym zbiór plików. Możecie edytować te pliki do woli, a jeśli są to pliki z kodem źródłowym, można z nich skompilować program w zwykły sposób. Kopia robocza jest prywatnym miejscem pracy: Subversion nie wprowadza zmian innych ludzi, ani czyni własnych zmian dostępnych innym, dopóki wyraźnie nie dacie znać, by to uczynić.

Po dokonaniu pewnych zmian na plikach w kopii roboczej i sprawdzeniu, czy działają poprawnie, Subversion dostarcza ci poleceń *publikacji* zmian dla innych osób pracujących razem nad projektem (zapisując do repozytorium). Jeśli inni ludzie publikują własne zmiany, Subversion daje wam polecenia do scalenia tych zmian w katalogu roboczym (poprzez odczyt z repozytorium).

Kopia robocza zawiera również pewne dodatkowe pliki, tworzone i zarządzane przez Subversion, wspomagające realizację tych poleceń. W szczególności wasza kopia robocza zawiera podfolder o nazwie `.svn`, znany również jako *folder administracyjny* kopii roboczej. Pliki w tym folderze administracyjnym pomagają Subversion rozpoznać, które pliki zawierają nieopublikowane zmiany, a które są nieaktualne względem pracy innych osób. Przed wersją 1.7, Subversion utrzymywał foldery administracyjne `.svn` w każdym wersjonowanym folderze kopii roboczej. Subversion 1.7 przyjął zupełnie inne podejście i każda kopia robocza posiada teraz tylko jeden folder administracyjny znajdujący się w głównym folderze kopii roboczej.

Typowe repozytorium Subversion często zawiera pliki (lub kod źródłowy) kilku projektów, zazwyczaj każdy projekt jest podfolderem drzewa katalogów w systemie plików. W tym układzie kopia robocza użytkownika zazwyczaj odpowiada szczególnemu poddrzewu repozytorium.

Załóżmy dla przykładu, że macie repozytorium, zawierające dwa projekty programów.



Rysunek 2.6. Struktura plików repozytorium

Innymi słowy, korzeń drzewa katalogów repozytorium ma dwa podkatalogi: `paint` i `calc`.

Aby otrzymać kopię roboczą, musicie *pobrać* pewne poddrzewo z repozytorium. (Termin *pobrać* może brzmieć jakby miał coś wspólnego z blokadą lub rezerwacją zasobów, ale tak nie jest, polecenie po prostu tworzy prywatną kopię projektu dla Was.)

Załóżmy, że wprowadziliście zmiany do `przycisk.c`. Ponieważ katalog `.svn` pamięta datę modyfikacji pliku i oryginalną zawartość, Subversion może powiedzieć, że zmieniliście plik. Jednak Subversion nie uczyni zmian publicznymi aż wyraźnie to powiecie. Akt publikowania zmian jest bardziej znany jako *zatwierdzenie* (lub *zarezerwowanie*) zmiany do repozytorium.

Aby ujawnić zmiany innym, możesz użyć polecenia Subversion **commit** (zatwierdź).

Teraz zmiany pliku `przycisk.c` zostały zatwierdzone i przesłane do repozytorium, jeśli inny użytkownik pobiera kopię roboczą `/calc`, będzie widzieć zmiany z najnowszej wersji pliku.

Załóżmy, że macie współpracownika, Agatkę, która pobiera kopię roboczą `/calc` w tym samym czasie co wy. Kiedy zatwierdzacie swoje zmiany na `przycisk.c`, kopia robocza Agatki pozostaje bez zmian; Subversion modyfikuje kopie robocze tylko na życzenie użytkownika.

Aby doprowadzić swój projekt najnowszego stanu, Agatka może poprosić Subversion o *uaktualnienie* jej kopii roboczej, za pomocą polecenia Subversion **update** (uaktualnij). Pozwoli to uwzględnić wprowadzone przez Was zmiany do jej kopii roboczej, jak również wszelkie inne, które zostały zatwierdzone od czasu tego uaktualnienia.

Należy pamiętać, że Agatka nie musi określać, które pliki uaktualnić; Subversion używa informacji zawartych w katalogu `.svn`, i kolejnych informacji z repozytorium, aby zdecydować, które pliki muszą być uaktualnione.

2.3.2. Adresy URL repozytorium

Dostęp do repozytoriów Subversion można uzyskać za pomocą różnych metod - z dysku lokalnego lub za pośrednictwem różnych protokołów sieciowych. Lokalizacja repozytorium jednak zawsze jest adresem URL. Schemat URL wskazuje metodę dostępu:

Schemat	Metoda dostępu
`file://`	Bezpośredni dostęp do repozytorium na dysku lokalnym lub sieciowym.

Schemat	Metoda dostępu
`http://`	Dostęp przez protokół WebDAV do serwera Apache z zainstalowanym Subversion.
`https://`	To samo co `http://`, ale z szyfrowaniem SSL.
`svn://`	Nieuwierzytelniony dostęp TCP/IP za pośrednictwem niestandardowego protokołu do serwera `svnserve`.
`svn+ssh://`	uwierzytelniony, szyfrowany dostęp TCP/IP za pośrednictwem niestandardowego protokołu do serwera `svnserve`.

Tabela 2.1. URL dostępu do repozytorium

W większości adresy URL dla Subversion używają standardowej składni, dopuszczającej, by nazw serwerów i numerów portów były określone jako część adresu URL. Metoda dostępu `file://` jest zazwyczaj używana do dostępu lokalnego, chociaż może być używana ze ścieżką UNC do hosta w sieci. Dlatego URL przybiera formę `file://nazwaHosta/sciezka/do/repozytorium`. Na komputerze lokalnym, część`nazwaHosta` adresu URL musi być pusta lub `localhost`. Z tego powodu, w lokalnych ścieżkach zwykle pojawiają się trzy ukośniki, `file:///ścieżka/do/repozytorium`.

Ponadto użytkownicy schematu`file://` na platformie Windows będą musieli korzystać z nieoficjalnie „standardowej” składni dostępu do repozytoriów, które są na tej samej maszynie, ale na innym dysku niż bieżący dysk roboczy klienta. Obie z poniższych składni ścieżki URL działają, przy czym `X` jest dyskiem, na którym znajduje się repozytorium:

```
file:///X:/sciezka/do/repoz
...
file:///X|/sciezka/do/repoz
...
```

Należy pamiętać, że adres URL używa zwykłych ukośników pomimo że natywna (nie URL) forma ścieżki w Windows używa odwrotnych ukośników.

Możecie uzyskać dostęp do repozytorium FSFS przez udziały sieciowe, ale to *nie* jest zalecane z różnych powodów:

- Dajecie bezpośredni dostęp do zapisu wszystkim użytkownikom, zatem mogą oni przypadkowo usunąć lub uszkodzić system plików repozytorium.
- Nie wszystkie protokoły udostępniania plików w sieci wspierają blokady wymagane przez Subversion. Pewnego dnia możecie zastać wasze repozytoria z lekka *uszkodzone*.
- Musicie ustawić uprawnienia dostępu w odpowiedni sposób. SAMBA jest pod tym względem szczególnie trudna.
- Gdy jakaś osoba zainstaluje nowszą wersję klienta, która zaktualizuje format repozytorium, wszyscy inni nie będą mogli uzyskać dostępu do repozytorium, póki nie zaktualizują klienta do nowej wersji.

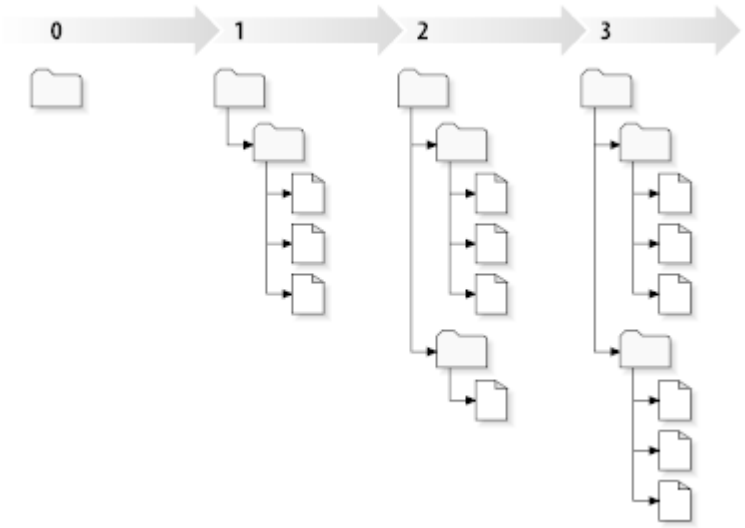
2.3.3. Wersje

Operacja **svn commit** pozwala opublikować zmiany dla dowolnej liczbie plików i katalogów jako pojedyncza transakcja atomowa. W swojej kopii roboczej możecie zmienić zawartość plików, tworzyć, usuwać, zmieniać nazwy i kopiować pliki i katalogi, a następnie zatwierdzić komplet zmian w pojedynczym działaniu.

W repozytorium każde zatwierdzenie jest traktowane jako transakcja atomowa: albo nastąpiło zatwierdzenie wszystkich zmian, albo żadna z nich nie dochodzi do skutku. Subversion zachowuje tą niepodzielność wobec awarii programów, awarii systemu, problemów z siecią i działań innych użytkowników.

Za każdym razem, gdy repozytorium akceptuje zatwierdzenie, tworzy nowy stan drzewa systemu plików, zwanego *wersją*. Każda wersja ma przypisaną unikalną liczbę naturalną, o jeden większą niż liczba wcześniejszej wersji. Początkowa wersja świeżo utworzonego repozytorium ma numer zero i składa się wyłącznie z pustego katalogu.

Dobrym sposobem, wizualizacji repozytorium jest szereg drzew. Wyobraźcie sobie szereg numerów wersji, zaczynając od 0, ciągnący się z lewej do prawej. Każdy numer wersji ma poniżej dołączone drzewo plików, a każde drzewo jest „migawką" wyglądu repozytorium po każdym zatwierdzeniu.



Rysunek 2.7. Repozytorium

Globalne numery wersji

W przeciwieństwie do wielu innych systemów kontroli wersji, w Subversion numery wersji odnoszą się do *całych drzew* , a nie pojedynczych plików. Każdy numer wersji obejmuje całe drzewo jako szczególny stan repozytorium po określonym zatwierdzeniu zmian. Inny sposób myślenia o tym zakłada, że wersja N reprezentuje stan systemu plików repozytorium z N-tego zatwierdzenia. Gdy użytkownik Subversion mówi o ``wersji 5 `foo.c`", tak naprawdę ma na myśli ``foo.c` taki, jak wyglądał w wersji 5". Zauważcie, że ogólnie wersje N i M pliku *nie muszą* się różnić!

Ważne jest, aby pamiętać, że kopie robocze nie zawsze odpowiadają pojedynczym wersjom w repozytorium, mogą zawierać pliki z kilku różnych wersji. Na przykład załóżmy, że pobierzecie kopię roboczą z repozytorium, którego najnowszą wersją jest 4:

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

W chwili obecnej ten katalog roboczy odpowiada dokładnie wersji 4 w repozytorium. Załóżmy jednak, że dokonujecie zmiany w `przycisk.c` i zatwierdzacie tą zmianę. Zakładając, że żadne inne zatwierdzenia nie miały miejsca, wasze zatwierdzenie stworzy wersję 5 repozytorium, a wasza kopia robocza będzie teraz wyglądać tak:

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

Załóżmy, że w tym momencie Agatka wprowadza zmianę w `naturalna.c`, tworząc wersję 6. Jeśli użyjecie **svn update**, aby zaktualizować swoją kopię roboczą, to będzie ona wyglądać tak:

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

Zmiany Agatki w `naturalna.c` pojawią się w twojej kopii roboczej, a wasze zmiany będą nadal obecne w `przycisk.c`. W tym przykładzie treść `Makefile` jest identyczna w wersjach 4, 5 i 6, ale Subversion zaznacza waszą kopię roboczą `Makefile` wersją 6 by wskazać, że jest nadal aktualna. Tak więc, po wykonaniu czystej aktualizacji na korzeniu kopii roboczej, na ogół będzie ona odpowiadała dokładnie jednej wersji w repozytorium.

2.3.4. Jak kopie robocze monitorują repozytorium

Dla każdego pliku w katalogu roboczym Subversion przechowuje dwie zasadnicze części informacji w przestrzeni administracyjnej `.svn/`:

- na jakiej wersji bazuje plik roboczy (jest to tzw *wersja robocza* pliku) oraz
- znacznik czasu zapisu kiedy lokalna kopia została zaktualizowana w repozytorium.

Uwzględniając te informacje, podczas komunikacji z repozytorium, Subversion może stwierdzić, w jakim z następujących czterech stanów znajduje się plik roboczy:

Bez zmian i aktualny
: Plik jest niezmieniony w katalogu roboczym i żadne zmiany w tym pliku nie zostały zapisane w repozytorium od jego wersji roboczej. Polecenie **commit** na tym pliku nic nie zmieni ani **update** na pliku nic nie zdziała.

Lokalnie zmieniony i aktualny
: Plik został zmieniony w folderze roboczym i żadne zmiany w tym pliku nie zostały zapisane w repozytorium od jego wersji roboczej. Istnieją lokalne zmiany, które nie zostały zatwierdzone do repozytorium, a tym samym polecenie **commit** na pliku powiedzie się opublikowaniem zmian a **update** pliku nic nie zdziała.

Bez zmian i nieaktualny
: Plik nie został zmieniony w folderze roboczym, ale został zmieniony w repozytorium. Plik powinien ostatecznie zostać zaktualizowany, aby był zgodny z wersją publiczną. Polecenie **commit** na pliku nie zrobi nic, a **update** pliku wprowadzi najnowsze zmiany do kopii roboczej.

Lokalnie zmieniony i nieaktualny
: Plik został zmieniony zarówno w folderze roboczym, jak i w repozytorium. Polecenie **commit** na pliku zakończy się niepowodzeniem z błędem *nieaktualny*. Plik powinien najpierw zostać zaktualizowany; polecenie **update** będzie próbowało scalić zmiany publiczne zmiany z lokalnymi. Jeśli Subversion nie uda się zakończyć automatycznego scalenia w sposób nadający się do przyjęcia, pozostawia rozwiązanie konfliktu przez połączenie zmian użytkownikowi.

2.4. Podsumowanie

Omówiliśmy w tym rozdziale wiele podstawowych pojęć Subversion:

- Wprowadziliśmy pojęcia centralnego repozytorium, klienckiej kopii roboczej i tablicy drzew wersji repozytorium.
- Obejrzeliśmy kilka prostych przykładów, jak dwaj współpracownicy mogą użyć Subversion do publikowania i odbierania zmiany od siebie nawzajem posługując się modelem 'kopia-modyfikacja-scalanie'.
- Omówiliśmy pobieżnie sposób, w jaki Subversion śledzi i zarządza informacjami w kopii roboczej.

Rozdział 3. Repozytorium

Bez względu na protokół użyty by uzyskać dostęp do repozytoriów, zawsze trzeba utworzyć co najmniej jedno repozytorium. Można to zrobić albo za pomocą klienta linii poleceń Subversion albo TortoiseSVN.

Jeśli jeszcze nie utworzyliście repozytorium Subversion, teraz nadszedł czas, aby to zrobić.

3.1. Tworzenie repozytorium

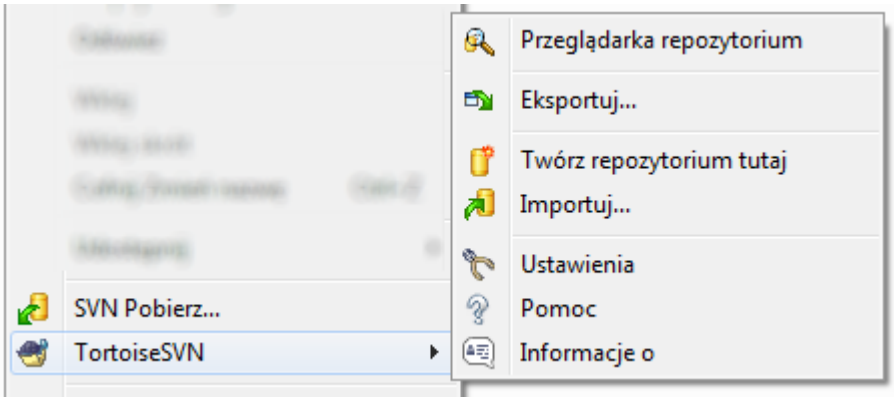
3.1.1. Tworzenie repozytorium przy użyciu klienta linii poleceń

1. Utwórzcie pusty folder o nazwie SVN (np. `D:\SVN\`), który będzie używany jest jako korzeń dla wszystkich repozytoriów.
2. Utwórzcie inny folder `MyNewRepository` wewnątrz `D:\SVN\`.
3. Otwórzcie okno wiersza poleceń (lub DOS-Box), przejdźcie do `D:\SVN\` i wpiszcie

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

Macie teraz nowe repozytorium znajdujące się w `D:\SVN\MyNewRepository`.

3.1.2. Tworzenie repozytorium przy użyciu TortoiseSVN



Rysunek 3.1. Menu TortoiseSVN dla niewersjonowanych folderów

1. Otwórzcie eksplorator windows
2. Utwórzcie nowy folder i nazwijcie go np. `SVNRepository`
3. Kliknijcie prawym przyciskiem myszy na nowo utworzony folder i wybierzcie TortoiseSVN → Twórz repozytorium tutaj.

   Repozytorium jest tworzone wewnątrz nowego folderu. *Nie wolno edytować tych plików samemu!!!*. W przypadku wystąpienia jakichkolwiek błędów upewnij się, że folder jest pusty i niezabezpieczony przed zapisem.

   Zostaniecie również zapytani, czy chcesz utworzyć strukturę katalogów w repozytorium. Dowiedzcie się więcej o opcjach układu katalogów przez link Sekcja 3.1.5, „Układ repozytorium”.

   Podczas tworzenia repozytorium TortoiseSVN ustawia niestandardową ikonę folderu, dzięki czemu można łatwo zidentyfikować lokalne repozytoria. Jeśli stworzysz repozytorium przy użyciu oficjalnego klienta linii poleceń, ikona nie zostanie przypisana do folderu.

Podpowiedź

Oczywiście nie polecamy również w ogóle używania dostępu przez `file://` za wyjątkiem lokalnego testowania. Korzystanie z serwera jest bardziej bezpieczne i niezawodne jeśli nie korzysta z niego jeden deweloper.

3.1.3. Lokalny dostęp do repozytorium

Aby uzyskać dostęp do lokalnego repozytorium potrzebna Ci ścieżka do jego folderu. Wystarczy pamiętać, że Subversion wymaga, by wszystkie ścieżki repozytorium były w postaci `file:///C:/SVNRepository/`. Zwróćcie uwagę na użyte wewnątrz ukośniki.

Aby uzyskać dostęp do repozytorium znajdującego się na udziale sieciowym można użyć mapowania dysku lub ścieżki UNC. Dla ścieżek UNC, użyć należy formy `file://NazwaSerwera/ścieżka/do/repoz/`. Należy pamiętać, że występują tu tylko 2 wiodące ukośniki.

Przed SVN 1.2, ścieżki UNC musiały być wprowadzane w bardziej niejasnej formie `file:///\NazwaSerwera/ścieżka/do/repoz`. Ta forma jest nadal obsługiwana, ale nie zalecana.

3.1.4. Dostęp do repozytorium na udziale sieciowym

Choć teoretycznie jest możliwe umieszczenie repozytorium FSFS na udziale sieciowym i dostęp wielu użytkowników za pomocą protokołu `file://`, jest to zdecydowanie *nie* zalecane. W rzeczywistości *mocno* odradzamy i nie wspieramy takiego jego wykorzystania z różnych powodów:

- Po pierwsze dajecie wszystkim użytkownikom bezpośredni dostęp do zapisu w repozytorium, więc każdy użytkownik może przypadkowo usunąć całe repozytorium lub uczynić je bezużytecznym w inny sposób.
- Po drugie, nie wszystkie protokoły udostępniania plików w sieci wspierają blokady, których wymaga Subversion, więc może się okazać, że repozytorium zostanie uszkodzone. Może nie stanie się to od razu, ale pewnego dnia dwóch użytkowników będzie próbowało uzyskać dostęp do repozytorium w tym samym czasie.
- Po trzecie, uprawnienia do plików muszą być ustawione dokładnie takie, jak trzeba. Być może ominiecie problem wykorzystując natywne współdzielenie z systemu Windows, ale z SAMBĄ jest to szczególnie trudne.
- Gdy jakaś osoba zainstaluje nowszą wersję klienta, która zaktualizuje format repozytorium, wszyscy inni nie będą mogli uzyskać dostępu do repozytorium, póki nie zaktualizują klienta do nowej wersji.

Dostęp przez `file://` jest przeznaczony dla lokalnego dostępu jednego użytkownika tylko, w szczególności podczas testowania i debugowania. Jeśli chcesz udostępnić repozytorium, *naprawdę* konieczne jest postawienie odpowiedniego serwera, a to nie jest tak trudne, jak mogłoby się wydawać. Czytajcie Sekcja 3.5, „Dostęp do repozytorium” by uzyskać wytyczne dotyczące wyboru i konfiguracji serwera.

3.1.5. Układ repozytorium

Przed zaimportowaniem informacji do repozytorium, należy najpierw zastanowić się, jak chcecie zorganizować swoje dane. Jeśli użyjecie jednego z zalecanych układów będzie Wam później znacznie łatwiej.

Istnieje kilka standardowych, zalecanych sposobów organizowania repozytorium. Większość ludzi tworzy katalog `trunk` trzymający „główną linię” rozwoju, folder `branches` zawierający odgałęziające się kopie i `tags` zawierający etykietowane kopie. Jeśli repozytorium posiada tylko jeden projekt, ludzie tworzą często katalogi najwyższego poziomu:

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

Ponieważ ten układ jest tak często używany podczas tworzenia nowego repozytorium za pomocą TortoiseSVN, program będzie również proponował Wam utworzenie struktury katalogów.

Jeśli repozytorium zawiera wiele projektów, ludzie często indeksują swój układ według gałęzi:

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

... lub według projektu:

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

Indeksowanie według projektu ma sens, jeśli projekty nie są ściśle powiązane i każdy jest sprawdzany indywidualnie. Dla projektów powiązanych, na których moglibyście chcieć pobrać wszystkie projekty na raz, lub gdy wszystkie projekty są ze sobą powiązane w jednym pakiecie dystrybucyjnym, często lepiej jest indeksować według gałęzi. W ten sposób masz tylko jedną linię główną do pobrania, a relacje między podprojektami są bardziej widoczne.

Jeśli przyjąć podejście z najwyższym poziomem `/trunk /tags /branches`, nie potrzeba nadmieniać, że należy skopiować cały folder trunk do osobnego folderu w branch lub tags, a pod pewnymi względami ta struktura oferuje wysoką elastyczność.

Dla niepowiązanych projektów wolicie może użyć oddzielnych repozytoriów. Kiedy zatwierdzacie zmiany, zmieniany jest numer wersji całego repozytorium, a nie numer wersji projektu. Trzymanie dwóch niepowiązanych projektów dzielących jedno repozytorium może oznaczać duże przeskoki w numerach wersji. Projekty w Subversion i TortoiseSVN wyświetlane są pod tym samym adresem hosta, ale użycie zupełnie niezależnych repozytoriów umożliwi samodzielny rozwój i brak nieładu przy numerach kompilacji.

Oczywiście, możecie swobodnie zignorować te powszechne układy. Możecie tworzyć dowolne mutacje, byleby najlepiej spełniały oczekiwania Wasze i Waszych zespołów. Pamiętajcie, że cokolwiek wybierzecie, nie stanowi to trwałego zobowiązania. Możecie zreorganizować repozytorium w dowolnym momencie. Ponieważ gałęzie i etykiety są zwykłymi folderami, TortoiseSVN może przenieść lub zmienić ich nazwy jednak chcecie.

Przełączenie z jednego układu do drugiego jest tylko kwestią opublikowania serii ruchów po stronie serwera; Jeśli nie lubicie sposobu, w jaki rzeczy uporządkowane są w repozytorium, po prostu przestawcie katalogi dookoła.

Więc jeśli nie utworzyliście jeszcze podstawowej struktury folderów wewnątrz repozytorium powinniście to zrobić teraz. Istnieją dwa sposoby osiągnięcia tego celu. Jeśli chcecie po prostu stworzyć strukturę `/trunk / tags /branches`, możecie skorzystać z przeglądarki repozytorium, aby utworzyć trzy foldery (w trzech odrębnych zatwierdzeniach). Jeśli chcecie stworzyć szerszą hierarchię, prostszym sposobem jest utworzenie najpierw struktury folderów na dysku i zaimportowanie jej jednym zatwierdzeniem, w taki sposób:

1. utwórzcie nowy pusty folder na dysku twardym
2. utwórzcie pożądaną strukturę folderów na najwyższego poziomu wewnątrz tego folderu - nie umieszczajcie w nim jeszcze plików!

3. wczytajcie tą struktury do repozytorium przez kliknij prawym klawiszem myszy na folderze, który zawiera strukturę katalogów i wybierzcie TortoiseSVN → Importuj.... W oknie dialogowym importu wpiszcie adres URL do repozytorium, a następnie kliknijcie OK. w ten sposób folder temp zostanie wczytany do katalogu głównego repozytorium, aby utworzyć podstawowy układ repozytorium.

Zauważcie, że nazwa folderu, z którego są importowane nie pojawi się w repozytorium, a tylko jego zawartość. Na przykład, utwórzcie następującą strukturę folderów:

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

Zaimportujcie `C:\Temp\New` do katalogu głównego repozytorium, a będzie on wyglądać tak:

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

3.2. Archiwizacja repozytorium

Niezależnie od użytego typu repozytorium, jest niezwykle ważne, aby wykonywać regularnie kopie zapasowe, i weryfikować każdą kopię zapasową. Jeśli serwer zawiedzie, będziecie mieli dostęp do najnowszych wersji własnych plików, ale bez repozytorium cała historia pracy zostaje utracona na zawsze.

Najprostszym (ale nie zalecanym) sposobem jest po prostu skopiowanie folderu repozytorium na nośnik kopii zapasowej. Jednak musicie mieć absolutną pewność, że żaden proces nie ma wtedy dostępu do danych. W tym kontekście, dostęp oznacza *absolutnie każde* połączenie. Jeśli łączono się z repozytorium podczas kopiowania, (pozostawiona otwarta przeglądarka internetowa, WebSVN itp.) kopia zapasowa będzie bezwartościowa.

Zalecaną metodą jest wykonanie

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

by utworzyć kopię repozytorium w bezpieczny sposób. A następnie backup kopii.

Narzędzie `svnadmin` tjest instalowane automatycznie podczas instalowania klienta linii poleceń Subversion. Najprostszym sposobem by to uzyskać jest zaznaczenie opcji dodania narzędzi linii poleceń podczas instalacji TortoiseSVN, jednak jeśli wolicie, możecie pobrać najnowszą wersję narzędzi linii poleceń z witryny *Subversion* [https://subversion.apache.org/packages.html#windows].

3.3. Skrypty przechwytujące po stronie serwera

Skrypt przechwytujący jest programem wywołanym przez jakieś zdarzenie repozytorium, jak tworzenie nowej wersji lub zmiany niewersjonowanego atrybutu. Każdemu przechwyceniu przekazywane jest tyle informacji, aby wiedziało, jakie to jest zdarzenie, na jakim elemencie(ach) działa oraz znało nazwę użytkownika osoby, która wywołała zdarzenie. W zależności od wyjścia lub stanu zwróconego przez przechwycenie, program przechwytujący może kontynuować działanie, zatrzymać lub zawiesić je w pewien sposób. Odnośny rozdział *Skrypty przechwytujące* [http://svnbook.red-bean.com/en/1.8/svn.reposadmin.create.html#svn.reposadmin.create.hooks] w Księdze Subversion pozwala uzyskać pełne informacje na temat przechwyceń, które są realizowane.

Te skrypty przechwytujące wykonywane są przez serwer, na którym znajduje się repozytorium. TortoiseSVN pozwala również skonfigurować skrypty przechwytujące po stronie klienta, które są wykonywane lokalnie przy pewnych zdarzenia. Zobaczcie Sekcja 4.31.8, „Skrypty przechwytujące po stronie klienta” by uzyskać więcej informacji.

Przykładowe skrypty przechwytujące znajdują się w folderze `hooks` repozytorium. Skrypty te są odpowiednie dla serwerów z systemem Unix/Linux, ale muszą zostać poprawione, jeżeli serwer jest oparty na systemie Windows.

Skrypt może być programem wsadowym lub wykonywalnym. Poniższy przykład pokazuje program wsadowy, który może być wykorzystany do wykonania do podmiany przechwycenia pre-revprop.

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

Pamiętajcie, że wszystko wysyłane do stdout zostaje utracone. Jeśli chcecie by wiadomość pojawiła się w oknie Zatwierdzenie Odrzucone, musicie ją wysłać do stderr. W pliku wsadowym wykonano to za pomocą `>&2`.

Nadpisywanie przechwyceń

Jeśli skrypt przechwytujący odrzuca Wasze zatwierdzenie, jego decyzja jest ostateczna. Ale można zbudować mechanizm nadpisania do skryptu za pomocą techniki *Magicznego słowa*. Jeśli skrypt chce odrzucić operację, najpierw skanuje wiadomość dziennika szukając specjalnego hasła, przy czym może to być stałe wyrażenie lub nazwa pliku z prefiksem. Jeśli odnajdzie magiczne słowo to pozwala zatwierdzeniu przejść dalej. Jeśli wyrażenie nie zostanie znalezione to może zablokować zatwierdzenie z komunikatem „Nie wpisałeś magicznego słowa”. :-)

3.4. Linki pobierania

Jeśli chcecie, aby Wasze repozytorium Subversion dostępne było dla innych, może chcecie umieścić link do niego na swojej stronie. Jednym ze sposobów uczynienia go bardziej dostępnym jest dołączenie *linku pobierania* dla innych użytkowników TortoiseSVN.

Po zainstalowaniu TortoiseSVN, rejestruje ono nowy protokół `tsvn:`. Gdy użytkownik TortoiseSVN kliknie na taki link, otworzy się automatycznie okno dialogowe pobierania z wypełnionym już adresem URL repozytorium.

Aby umieścić taki link na swojej stronie html, trzeba wprowadzić na niej kod, który wygląda tak:

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

Oczywiście będzie to wyglądać jeszcze lepiej, jeśli zawiera odpowiedni obraz. Możecie użyć *logo TortoiseSVN* [https://tortoisesvn.net/images/TortoiseCheckout.png] lub własnego rysunku.

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

Można też zrobić łącze bezpośrednio do określonej wersji, na przykład

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

3.5. Dostęp do repozytorium

Aby korzystać z TortoiseSVN (albo innego klienta SVN), potrzebujesz miejsca, w którym znajduje się repozytorium. Można przechowywać repozytoria lokalnie i mieć dostęp do nich za pomocą protokołu `file://`, można też umieścić je na serwerze, a dostęp do nich przez protokoły `http://` lub `svn://`. Te dwa protokoły dostępu do serwera mogą również być szyfrowane. Należy skorzystać z `https://` lub `svn+ssh://`, ewentualnie możecie użyć `svn://` z SASL.

Jeśli korzystacie z usług publicznych, takich jak hosting *Google Code* [https://sourceforge.net] albo serwer jest już przez kogoś skonfigurowany, to nie musicie już nic robić. Przejdźcie do Rozdział 4, *Instrukcja codziennej obsługi*.

Jeśli nie masz serwera i pracuje tylko jedna osoba, lub jeśli tylko sprawdzacie w zamkniętym środowisku Subversion i TortoiseSVN, lokalne repozytoria są prawdopodobnie najlepszym wyborem. Wystarczy utworzyć repozytorium na swoim komputerze, jak opisano wcześniej w Rozdział 3, *Repozytorium*. Można wtedy pominąć resztę tego rozdziału i przejść bezpośrednio do Rozdział 4, *Instrukcja codziennej obsługi*, aby dowiedzieć się jak zacząć z niego korzystać.

Jeśli myśleliście o utworzeniu repozytorium wielu użytkowników repozytorium w zasobie sieciowym, przemyślcie to jeszcze raz. Przeczytajcie Sekcja 3.1.4, „Dostęp do repozytorium na udziale sieciowym” by dowiedzieć się, dlaczego uważamy, to za zły pomysł. Konfiguracja serwera nie jest tak trudna, jak się wydaje, i daje lepszą niezawodność i oraz prędkość.

Więcej szczegółów o opcjach serwera Subversion oraz możliwościach wyboru najlepszej architektury w zależności od sytuacji znajduje się w księdze Subversion pod *Konfiguracja Serwera* [http://svnbook.red-bean.com/en/1.8/svn.serverconfig.html].

W czasach początków Subversion, postawienie i regulacja serwera wymagały dużego zrozumienia konfiguracji serwera stąd w poprzednich wersjach ta instrukcja zawierała szczegółowe opisy, jak skonfigurować serwer. Od tego czasu wszystko stało się łatwiejsze, ponieważ istnieje co najmniej kilka dostępnych paczek instalatorów serwerów, które poprowadzi Was przez proces budowy i konfiguracji. Podajemy tu linki do niektórych znanych nam instalatorów:

- *VisualSVN* [https://www.visualsvn.com/server/]
- *CollabNet* [https://www.collab.net/products/subversion]

Można zawsze znaleźć najnowsze linki na stronie *Subversion* [https://subversion.apache.org/packages.html].

Możecie znaleźć więcej przewodników "Jak To Zrobić" na stronie *TortoiseSVN* [https://tortoisesvn.net/usefultips.html].

Rozdział 4. Instrukcja codziennej obsługi

Dokument ten opisuje codzienną pracę użytkowania klienta TortoiseSVN. *Nie* jest wprowadzeniem do systemów kontroli wersji, *nie* jest też wprowadzeniem do Subversion (SVN). Jest to raczej miejsce, gdzie można zajrzeć, gdy wie się mniej więcej, co się chce zrobić, ale nie bardzo pamięta się jak.

Jeśli potrzebujecie wprowadzenia do kontroli wersji z Subversion, zalecamy zapoznać się z fantastyczną książką: *Kontrola wersji z Subversion* [http://svnbook.red-bean.com/].

Dokument ten jest również w trakcie tworzenia, jak samo TortoiseSVN i Subversion. Jeśli znajdziecie jakieś błędy, zgłoście je na listę mailingową, abyśmy mogli zaktualizować dokumentację. Niektóre zrzuty ekranu w Instrukcji Codziennej Obsługi (ICO) mogą nie odzwierciedlać aktualnego stanu oprogramowania. Proszę, wybaczcie nam. Pracujemy nad TortoiseSVN w wolnym czasie.

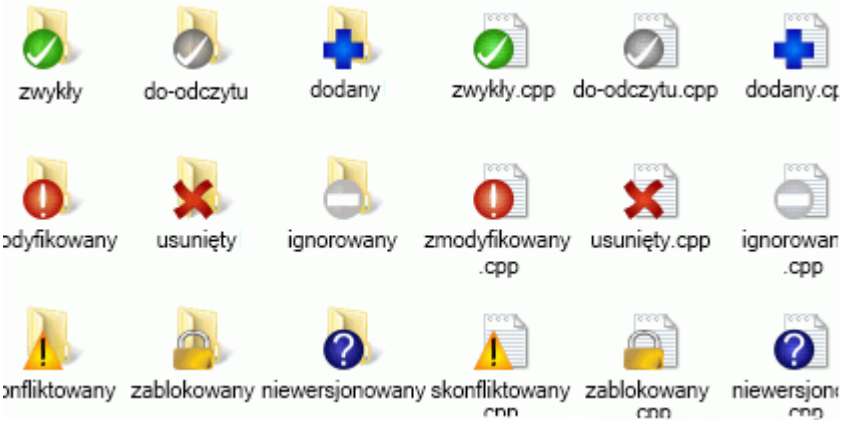
Aby uzyskać jak najwięcej z instrukcji codziennej obsługi:

- Powinniście mieć już zainstalowany TortoiseSVN.
- Powinniście być już zaznajomieni z systemami kontroli wersji.
- Powinniście znać podstawy Subversion.
- Trzeba skonfigurować serwer i/lub mieć dostęp do repozytorium Subversion.

4.1. Cechy podstawowe

Ta sekcja opisuje niektóre z cech TortoiseSVN, które mają zastosowanie do prawie całej zawartości instrukcji. Zauważcie, że wiele z tych funkcji ukazuje się tylko w kopii roboczej Subversion.

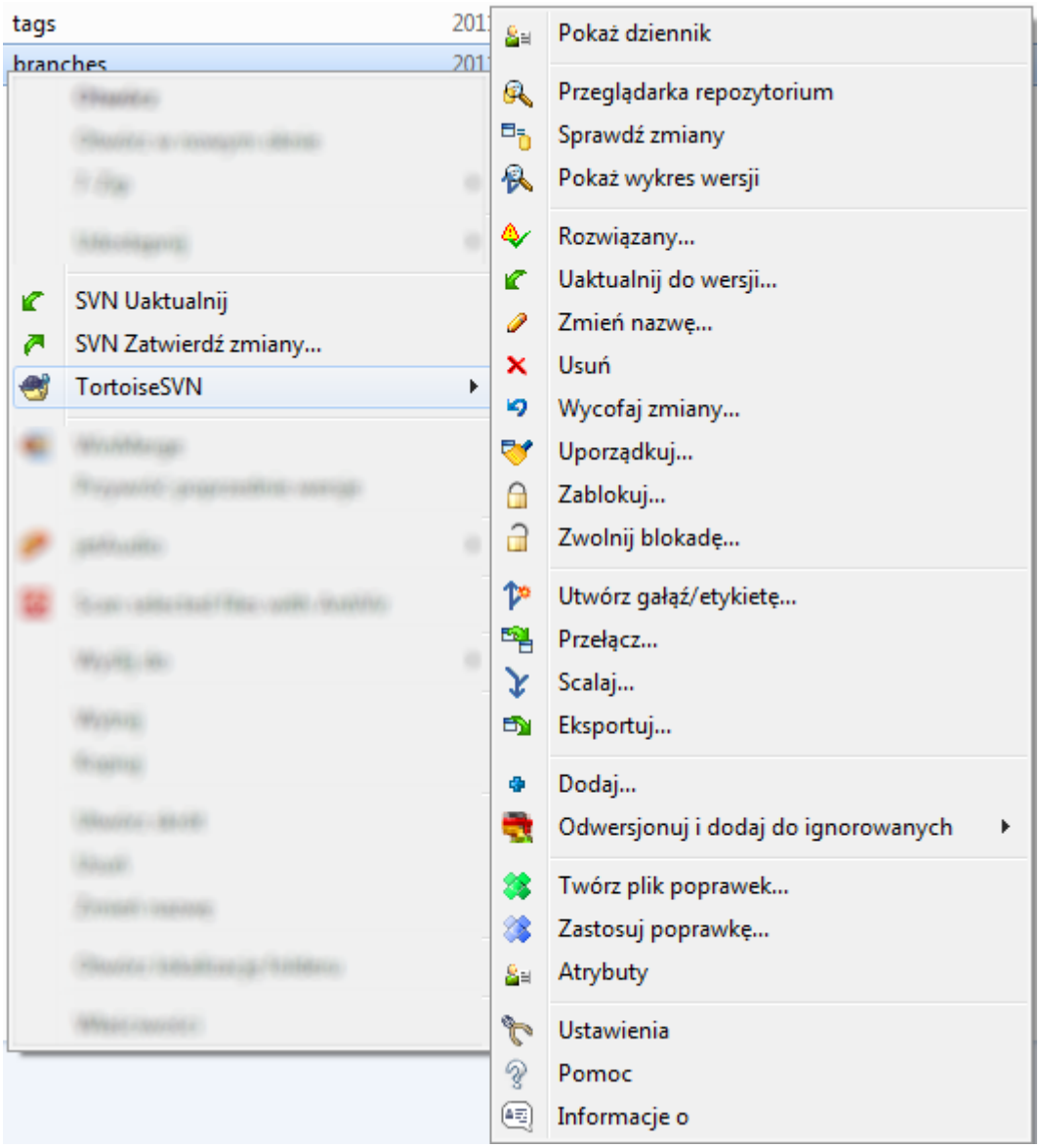
4.1.1. Ikony nakładkowe



Rysunek 4.1. Eksplorator pokazuje nakładki ikon

Jedną z najbardziej widocznych cech TortoiseSVN są nakładki ikon, które znajdują się na plikach w kopii roboczej. Uwidaczniają one na pierwszy rzut oka, które z Waszych plików zostały zmodyfikowane. Zapoznajcie się z Sekcja 4.7.1, „Ikony nakładkowe” by dowiedzieć się, co reprezentują różne nakładki.

4.1.2. Menu kontekstowe



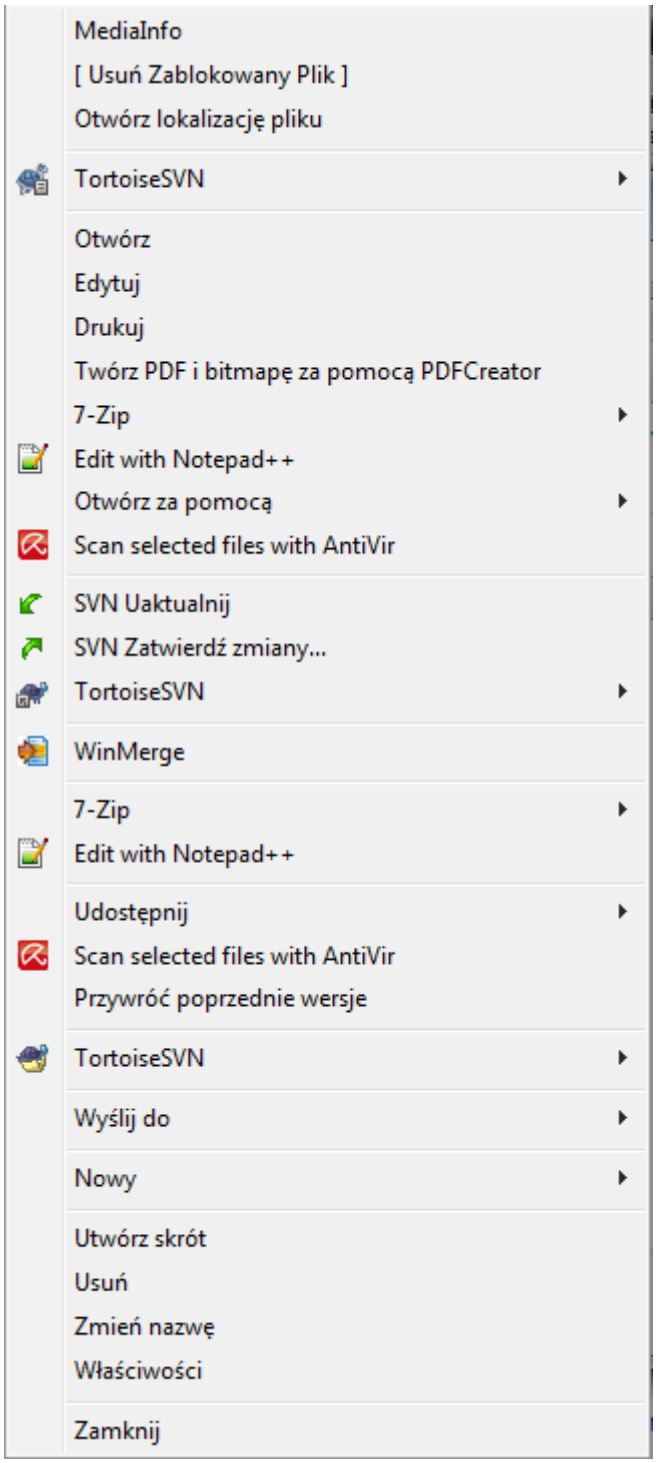
Rysunek 4.2. Menu kontekstowe dla katalogu pod kontrolą wersji

Wszystkie polecenia TortoiseSVN są wywoływane z menu kontekstowego eksploratora Windows. Większość z nich jest bezpośrednio widoczna, kiedy klikniesz prawym przyciskiem myszy na pliku lub folderze. Polecenia, które są dostępne, zależą od tego czy plik, folder lub folder nadrzędny jest pod kontrolą wersji, czy nie. Można także wyświetlić menu TortoiseSVN jako część menu górnego "Plik" eksploratora.

Podpowiedź

Niektóre polecenia, które są bardzo rzadko używane są dostępne tylko w rozszerzonym menu kontekstowym. Aby wywołać rozszerzone menu kontekstowe, należy przytrzymać klawisz **Shift**, podczas kliknięcia prawym przyciskiem myszy.

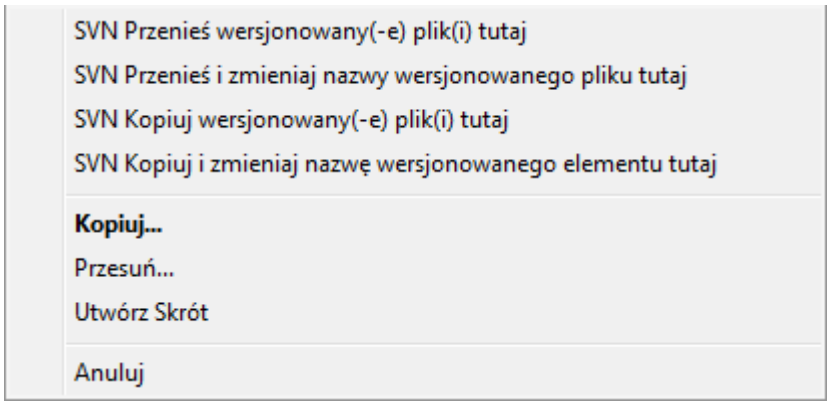
W niektórych przypadkach może pojawić się kilka wpisów TortoiseSVN. To nie jest błąd!



Rysunek 4.3. Menu "Plik" exploratora dla skrótu w wersjonowanym folderze

Ten przykład dotyczy niewersjonowanego skrótu w wersjonowanym folderze, gdy w menu Plik eksploratora są *trzy* wpisy dla TortoiseSVN. Jeden z nich do folderu, drugi dla samego skrótu, a trzeci dla obiektu, który skrót wskazuje. Aby pomóc Wam je rozróżnić, ikony mają wskaźnik w prawym dolnym rogu, aby pokazać, czy menu dotyczy pliku, folderu, skrótu lub wielu wybranych pozycji.

4.1.3. Mechanizm Przeciągnij i Upuść



Rysunek 4.4. Menu prawo-przeciągnięcia dla katalogu pod kontrolą wersji

Inne polecenia są dostępne przy obsłudze przeciągania, kiedy prawo-przeciąga się pliki lub foldery do nowej lokalizacji wewnątrz kopii roboczej lub po prawo-przeciągnięciu niewersjonowanego pliku lub folderu do katalogu, który jest pod kontrolą wersji.

4.1.4. Powszechne skróty

Niektóre powszechne działania mają doskonale znane skróty systemu Windows, ale nie pojawiają się na przyciskach ani w menu. Jeżeli nie możecie odnaleźć jak zrobić coś oczywistego, np. odświeżanie widoku, możecie sprawdzić tutaj.

F1
: Oczywiście pomoc.

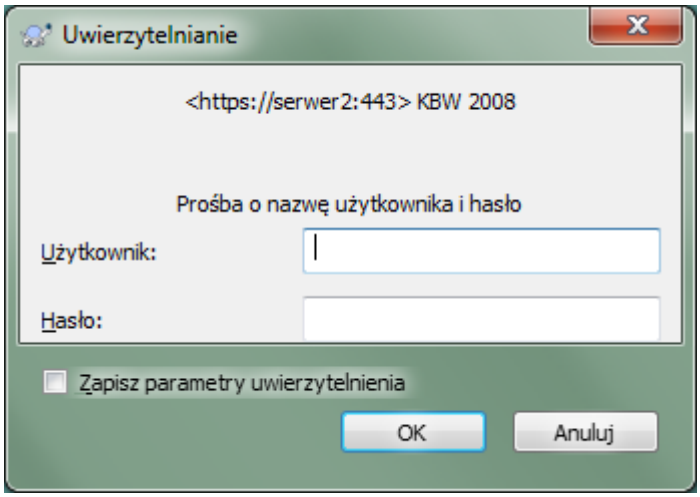
F5
: Odświeża bieżący widok. Być może jest to jedno z najbardziej przydatnych poleceń wykonywanych spod jednego klawisza. Na przykład ... W Eksploratorze będzie to odświeżenie nakładek ikon na kopii roboczej. W dialogu zatwierdzenia wykona ponowne skanowanie kopii roboczej, aby zobaczyć co koniecznie powinno zostać zatwierdzone. W oknie dialogowym dziennika wersji będzie łączyć się ponownie z repozytorium, aby sprawdzić najnowsze zmiany.

Ctrl-A
: Zaznaczenie wszystkiego. Może to być wykorzystywane, jeśli zostanie wyświetlony komunikat o błędzie i trzeba skopiować go i wkleić do e-mail. Użyjcie Ctrl-A, aby wybrać komunikat o błędzie, a następnie ...

Ctrl-C
: Kopiuje zaznaczony tekst. W przypadku gdy zaznaczono nie tekst, ale np. wpis na liście lub okno komunikatu, zawartość tej pozycji listy lub okna wiadomości jest kopiowana do schowka.

4.1.5. Autentykacja

Jeśli repozytorium, z którym próbujecie się połączyć, jest chronione hasłem, wyświetlony zostanie dialog uwierzytelniania.



Rysunek 4.5. Okno dialogowe autoryzacji

Wpiszcie nazwę użytkownika i hasło. zaznaczenie pola wyboru spowoduje przechowanie przez TortoiseSVN poświadczeń w katalogu domyślnym Subversion: `%APPDATA%\Subversion\auth` w trzech podkatalogach:

- `svn.simple` zawiera poświadczenia uwierzytelniania podstawowego (login/hasło). Należy pamiętać, że hasła są zapisywane za pomocą WinCrypt API, a nie w postaci otwartego tekstu.
- `svn.ssl.server` zawiera certyfikaty SSL serwera.
- `svn.username` zawiera poświadczenia wymagające tylko loginu użytkownika do uwierzytelniania (hasło niepotrzebne).

Jeśli chcecie wyczyścić pamięć podręczną uwierzytelniania, możecie to zrobić przez stronę Zapisane dane w oknie ustawień TortoiseSVN. Przycisk Czyść wszystko umożliwia usunięcie wszystkich danych w pamięci podręcznej uwierzytelniania z wszystkich repozytorium. Za to przycisk Czyść... wyświetli okno, gdzie można będzie wskazać, które buforowane dane uwierzytelnienia powinny zostać usunięte. Zapoznajcie się z Sekcja 4.31.6, „Ustawienia zapisanych danych”.

Niektórzy ludzie wolą mieć dane uwierzytelniające usunięte po wylogowaniu lub zamykaniu systemu Windows. Sposobem na to jest użycie skryptu zamknięcia, aby usunąć folder `%APPDATA%\Subversion\auth`, np.

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

Można znaleźć opis jak zainstalować taki skrypt na http://www.windows-help-central.com/windows-shutdown-script.html.

Dodatkowe informacje na temat sposobu konfiguracji serwera do uwierzytelniania i kontroli dostępu znajdują się na Sekcja 3.5, „Dostęp do repozytorium”.

4.1.6. Maksymalizacja Okien

Wiele okien dialogowych TortoiseSVN zawiera dużo informacji do wyświetlenia, ale często przydatna jest maksymalizacja tylko wysokości lub szerokości, a nie powiększenie tak, by wypełnić cały ekran. Dla wygody są skróty do nich na przycisku Maksymalizacja. Użyjcie środkowego przycisk myszy maksymalizuje w pionie, a prawego przycisk myszy maksymalizuje poziomo.

4.2. Importowanie danych do repozytorium

4.2.1. Import

Jeśli dane są importowane do istniejącego repozytorium, które zawiera już pewne projekty, struktura repozytorium została już ustalona. Jeśli importujecie dane do nowego repozytorium, warto poświęcić czas, aby zastanowić się, jak będzie ono zorganizowane. Przeczytajcie Sekcja 3.1.5, „Układ repozytorium” w celu zasięgnięcia porady.

W tej sekcji opisano polecenie Subversion import, którego przeznaczeniem jest importowanie hierarchii katalogów do repozytorium w jednym ruchu. Chociaż wykonuje tą pracę, posiada kilka niedociągnięć:

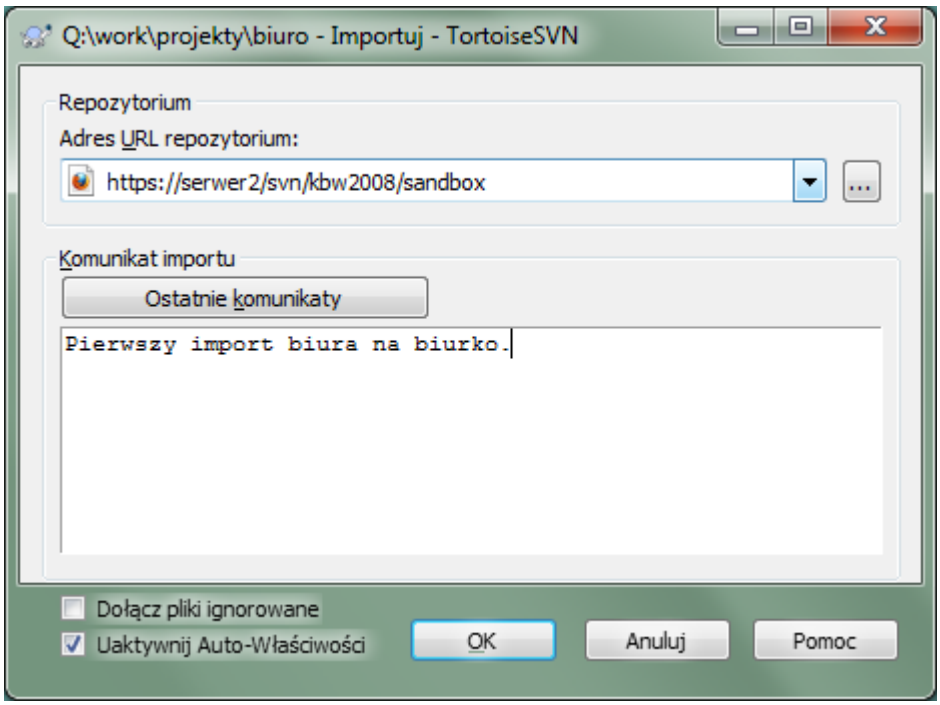
- Nie ma sposobu, by wybrać pliki i foldery, które dołączyć, oprócz wykorzystania z globalnych ustawień ignorowania.
- Importowany folder nie staje się kopią roboczą. W tym celu musicie pobrać na powrót pliki z serwera.
- Łatwo jest zaimportować do folderu na zły poziom w repozytorium.

Z tych powodów zaleca się nie korzystać w ogóle z polecenia import, ale raczej po dwuetapowej metody opisanej w Sekcja 4.2.2, „Import w miejscu”, chyba że wykonuje się prosty krok tworzenia wstępnej struktury `/trunk /tags /branches` w repozytorium. Ponieważ to czytacie, oto w jaki sposób działa podstawowe polecenie import ...

Przed importem projektu do repozytorium należy:

1. Usuńcie wszystkie pliki, które nie są potrzebne do budowy projektu (pliki tymczasowe, pliki, które są generowane przez np. kompilator *. obj, binarki, ...)
2. Zorganizujcie pliki w folderach i podfolderach. Choć można zmienić nazwę / przenieść pliki później, zaleca się, aby otrzymać projekt struktury już przed importem!

Teraz wybierzcie folder najwyższego poziomu w strukturze katalogów projektu w eksploratorze Windows i kliknąć prawym przyciskiem myszy, aby otworzyć menu kontekstowe. Wybierzcie polecenie TortoiseSVN → Importuj... co otworzy okno dialogowe:



Rysunek 4.6. Dialog importu

W tym oknie należy wprowadzić adres URL repozytorium, do którego chcecie zaimportować projekt. Bardzo ważne jest uświadomienie sobie, że sam lokalny folder importu nie pojawi się w repozytorium, a tylko jego zawartość. Na przykład, jeśli macie strukturę:

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

i importujecie `C:\Projects\Widget` do `http://mydomain.com/svn/trunk` to możecie być zaskoczeni, że podkatalogi trafiły prosto do folderu `trunk` zamiast pozostać w podkatalogu `Widget`. Musicie wskazać podkatalog jako część adresu URL, `http://mydomain.com/svn/trunk/Widget-X`. Zauważcie, że polecenie importu automatycznie utworzy podkatalogi w repozytorium, jeśli one nie istnieją.

Wiadomość importu jest wstawiana do dziennika wiadomości.

Domyślnie, pliki i foldery, które odpowiadają ogólnym wzorcom pomijania *nie* są importowane. Aby to zmienić należy zaznaczyć pole wyboru Dołącz pliki ignorowane. Zapoznajcie się z Sekcja 4.31.1, „Ustawienia ogólne” by uzyskać więcej informacji na temat ustawienia ogólnego wzorca pomijania.

Po naciśnięciu OK TortoiseSVN zaimportuje pełne drzewo katalogów, w tym wszystkie pliki, do repozytorium. Projekt jest teraz przechowywany w repozytorium pod kontrolą wersji. Należy pamiętać, że folder, który został zaimportowany *NIE* znajduje się pod kontrolą wersji! Aby uzyskać wersjonowaną *kopię roboczą* konieczne jest pobranie właśnie zaimportowanej wersji. Lub przeczytajcie dalej, aby dowiedzieć się, jak zaimportować folder w miejscu.

4.2.2. Import w miejscu

Zakładając, że macie już repozytorium, do którego chcecie dodać nową strukturę folderów, po prostu wykonajcie następujące kroki:

1. Skorzystajcie z przeglądarki repozytorium, aby utworzyć nowy folder projektu bezpośrednio w repozytorium. Jeśli używacie jednego ze standardowych układów będziecie prawdopodobnie chcieli utworzyć go jako podfolder linii głównej, a nie w katalogu głównym repozytorium. Przeglądarka repozytorium przedstawia strukturę repozytorium tak jak eksplorator Windows, dzięki czemu można zobaczyć, jak foldery są zorganizowane.

2. Pobieranie nowego folderu nadrzędnego do tego, który macie zamiar zaimportować. Pojawi się ostrzeżenie że folder lokalny jest niepusty. Należy je zignorować. Teraz macie wersjonowany folder najwyższego poziomu z niewersjonowaną zawartością.

3. Użyjcie TortoiseSVN → Dodaj... na tym wersjonowanym folderze, aby dodać część lub całą zawartość. Możecie dodawać i usuwać pliki, ustawić atrybut `svn:ignore` na folderach i dokonywać inne potrzebne zmiany.

4. Zatwierdźcie katalog najwyższego poziomu, i macie nowe wersjonowane drzewo oraz lokalną kopię roboczą, utworzoną z istniejącego folderu.

4.2.3. Pliki specjalne

Czasami trzeba trzymać pod kontrolą wersji plik, który zawiera dane specyficzne dla użytkownika. Oznacza to, że macie plik, który każdy programista/użytkownik musi zmodyfikować w celu dostosowania do swoich lokalnych ustawień. Ale wersjonowanie takiego pliku jest trudne, ponieważ każdy użytkownik chciałby za każdym razem zatwierdzić swoje zmiany do repozytorium.

W takich przypadkach zalecamy użycie *szablonu* plików. Utwórzcie plik, który zawiera wszystkie dane, jakich programiści będą potrzebować, dodajcie ten plik do kontroli wersji i niech programiści pobiorą ten plik. Następnie każdy programista powinien *wykonać kopię* z tego pliku i zmienić nazwę tej kopii. Po tym modyfikacja kopii nie stanowi już problemu.

Jako przykład można spojrzeć na skrypt wbudowany w TortoiseSVN. Wywołuje on plik o nazwie `default.build.user`, który nie istnieje w repozytorium. Za to jest tam plik `default.build.user.tmpl`. `default.build.user.tmpl` jest plikiem szablonu, z którego każdy

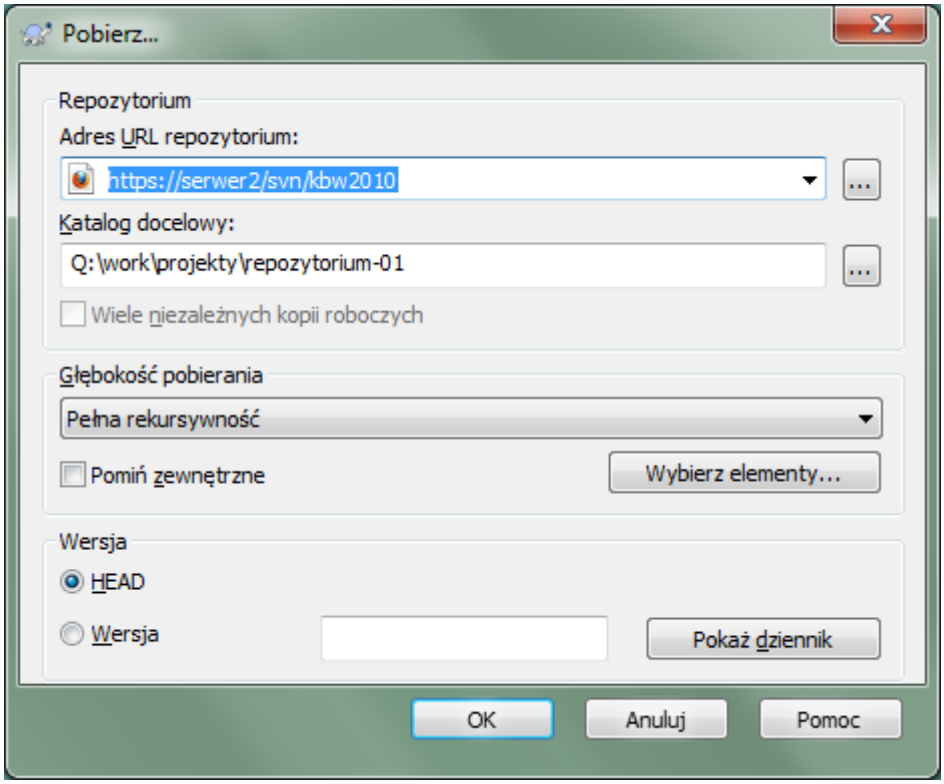
programista musi utworzyć kopię i zmienić jej nazwę na `default.build.user`. Wewnątrz tego pliku dodaliśmy komentarze wskazujące użytkownikom, które linie mają edytować i zmienić w zależności od lokalnej konfiguracji, aby móc go poprawnie stosować.

Jednocześnie, aby nie przeszkadzać użytkownikom, dodaliśmy także plik `default.build.user` do listy ignorowanych w jego folderze nadrzędnym, czyli mamy ustawić atrybut Subversion `svn:ignore` aby włączyć ten plik. W ten sposób nie będzie się pojawiać jako niewersjonowany przy każdym zatwierdzeniu.

4.3. Pobieranie kopii roboczej

Aby otrzymać kopię roboczą, musicie wykonać *pobranie* z repozytorium.

Wybierzcie w eksploratorze Windows katalog, w którym chcesz umieścić kopię roboczą. Teraz trzeba kliknąć prawym przyciskiem myszy aby pojawiło się menu kontekstowe po czym wybrać polecenie TortoiseSVN → Pobierz..., które przeniesie do następującego okna dialogowego:



Rysunek 4.7. Okno dialogowe pobierania

Jeśli wprowadzicie nazwę folderu, który jeszcze nie istnieje, to katalog o tej nazwie zostanie utworzony.

Ważne

W ustawieniach domyślnych pozycja menu pobierz nie znajduje się w podmenu TortoiseSVN ale pokazywana jest w głównym menu eksploratora. Polecenia TortoiseSVN nie umieszczone w podmenu posiadają przedrostek `SVN`: SVN Pobierz...

4.3.1. Głębokość pobierania

Można wybrać *głębokość* jaką chce się pobrać, co pozwala określić głębokość rekursji w folderach podrzędnych. Jeśli potrzeba tylko kilku sekcji z dużego drzewa, można pobrać tylko katalog najwyższego poziomu, a następnie aktualizować wybrane podkatalogi rekurencyjnie.

Pełna rekursywność
: Pobranie całego drzewa, w tym wszystkich folderów podrzędnych i dalszych podfolderów.

Bezpośrednie elementy podrzędne, w tym foldery
: Pobranie określonego katalogu, w tym wszystkich plików i foldery podrzędne, ale bez wypełnienia folderów podrzędnych.

Tylko pliki podrzędne
: Pobranie określonego katalogu, w tym wszystkich plików ale nie pobieranie żadnego z folderów podrzędnych.

Tylko ten element
: Pobranie tylko tego katalogu. Nie wypełnianie go ani plikami ani folderami podrzędnymi.

Kopia robocza
: Zachowaj głębokość określoną w kopii roboczej. Opcja ta nie jest używana w oknie pobrania, ale jest domyślna we wszystkich innych oknach dialogowych, które posiadają ustawienie głębokości.

Pomiń
: Stosowana w celu zmniejszenia głębokości kopii roboczej już po wypełnieniu folderu. Ta opcja jest dostępna tylko w oknie dialogowym Uaktualnij do wersji.

Aby łatwo wybrać tylko te elementy, które chcecie pobrać i wymusić by w wynikowej kopii roboczej utrzymać tylko te elementy, kliknijcie na przycisk Wybierz elementy.... Otworzy się nowe okno dialogowe, w którym można wybrać wszystkie elementy, jakie mają się znaleźć w kopii roboczej i usunąć zaznaczenie wszystkie niechciane elementy. Wynikowa kopia robocza jest nazywana `płytkim pobraniem`. Aktualizacja takiej kopii roboczej nie pobiera brakujących plików i folderów, a aktualizuje tylko te, które już są w waszej kopii roboczej.

Jeśli aktualizujecie rzadką kopięroboczą (tj, przez wybranie opcji głębokości aktualizacji innej niż `pełna rekurencja`), możecie w prosty sposób dodawaći usuwać podfoldery później przy użyciu następujących metod.

4.3.1.1. Rzadka Aktualizacja przy użyciu Aktualizacji do Wersji

Prawoklik na aktualizowanym folderze, następnie użycie TortoiseSVN → Aktualizuj do Wersji i wybór Wybierz elementy.... Otwiera to to samo okno, jakie było dostępne w zwykłej aktualizacji i pozwala wybrać i usunąć zaznaczenie elementów zawartych w aktualizacji. Metoda ta jest bardzo elastyczna ale może być powolna jako że każdy element folderu jest osobno aktualizowany.

4.3.1.2. Rzadka Aktualizacja przy użyciu Przeglądarki Repozytorium

Prawoklik na zaznaczonym folderze, po czym użycie TortoiseSVN → PrzeglądarkaRepozytorium by przywołaćprzeglądarkęrepozytorium. Odnalezienie podfolderu do dodania dokopii roboczej, następnie użycie Menu Kontekstowe → Aktualizuj element do wersji....

4.3.1.3. Rzadka Aktualizacja perzy wykorzystaniu Sprawdź Zmiany

Na oknie sprawdzenia zmian, najpierw z **shift** wciśnij przycisk Sprawdź repozytorium. Okno wyświetli wszystkie plikii foldery znajdujące sie w repozytorium ale nie zostały pobrane jako `dodane zdalnie`. Wykonajcie prawoklik na folderze(ach) które chcecie dodać do kopii roboczej, Po czym Menu kontekstowe → Uaktualnij.

Ta funkcja jest bardzo przydatna, gdy chcecie pobrać tylko części dużego drzewa z zachowaniem wygody aktualizowania pojedynczej kopii roboczej. Załóżmy, że macie duże drzewo, posiadające podfoldery `Project01` do `Project99` i chcecie pobrać tylko `Project03`, `Project25` i `Project76/SubProj`. Wykonajcie następujące kroki:

1. Pobierzcie folder macierzysty z głębokością „tylko ten element” Macie teraz pusty folder najwyższego poziomu.

2. Wybierzcie nowy folder i użyjcie TortoiseSVN → Przeglądarka repozytorium aby wyświetlić zawartość repozytorium.

3. Kliknijcie prawym przyciskiem myszy na `Project03` i menu kontekstowe → Uaktualnij element do wersji. Zachowajcie ustawienia domyślne i kliknijcie OK. Teraz macie ten folder w pełni wypełniony.

   Powtórzcie czynność dla `Project25`.

4. Przejdźcie do `Project76/SubProj` i zróbcie to samo. Tym razem zwróćcie uwagę, że folder `Project76` nie ma zawartości z wyjątkiem `SubProj`, który sam jest w pełni wypełniony. Subversion stworzył pośrednie foldery za Was bez wypełniania ich.

Zmiana głębokości roboczej kopii

Po pobraniu kopii roboczej do konkretnej głębokości możecie zmienić głębokość później, aby uzyskać zawartość w mniejszym lub większym stopniu za pomocą menu kontekstowego → Uaktualnij element do wersji. W tym oknie dialogowym należy zaznaczyć pole wyboru Określ ściśle zakres.

Korzystanie ze starszego serwera

Serwery pre-1.5 nie rozumieją żądań dotyczących głębokości kopii roboczej, dlatego nie zawsze obsługują sprawnie żądania. Polecenie będzie nadal działać, ale starszy serwer może wysłać wszystkie dane, pozostawiając klientowi odfiltrowanie elementów niewymaganych, co może oznaczać duży ruch w sieci. Jeśli to możliwe, należy uaktualnić serwer przynajmniej do wersji 1,5.

Jeśli projekt zawiera odniesienia do zewnętrznych projektów, których możesz *nie* chcieć pobrać się w tym samym czasie, należy użyć pola wyboru Pomiń zewnętrzne.

Ważne

Jeśli Pomiń zewnętrzne jest zaznaczone lub jeśli chcecie zwiększyć wartość głębokości, trzeba będzie wykonać aktualizacje kopii roboczej za pomocą TortoiseSVN → Uaktualnij element do wersji zamiast TortoiseSVN → Uaktualnij. Standardowo uaktualnienie zawiera wszystkie zależności zewnętrzne i zachowuje aktualną głębokości.

Zaleca się, aby uaktualniać tylko gałąź `trunk` drzewa katalogów, lub folder podrzędny. Jeśli wskażecie korzeń drzewa katalogów w URL, możecie skończyć z pełnym dyskiem twardym, ponieważ dostaniecie kopię całego repozytorium, w tym drzewa każdej gałęzi i etykiety projektu!

Eksportowanie

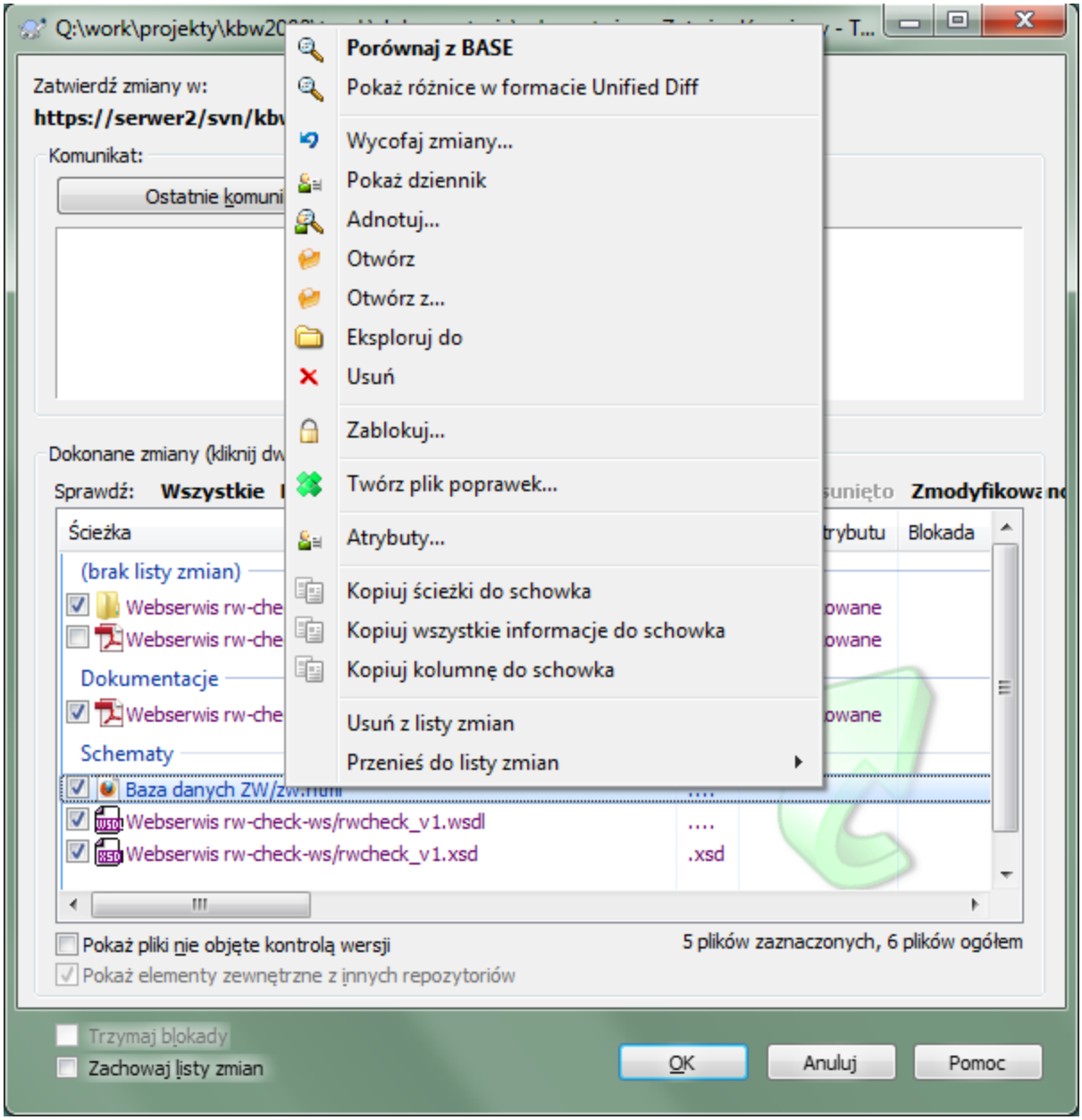
Czasami chcecie utworzyć lokalną kopię bez wszystkich tych katalogów `.svn`, by np. stworzyć archiwum zip źródła. Przeczytajcie Sekcja 4.27, „Eksport kopii roboczej Subversion”, aby dowiedzieć się jak to zrobić.

4.4. Zatwierdzanie zmian do repozytorium

Wysyłanie zmian dokonanych w kopii roboczej jest znane jako *zatwierdzenie* zmiany. Ale zanim zatwierdzicie, musicie upewnić się, że kopia robocza jest aktualna. Możecie użyć bezpośrednio TortoiseSVN → Uaktualnij. Można też użyć najpierw TortoiseSVN → Sprawdź zmiany, aby zobaczyć, które pliki zostały zmienione lokalnie lub na serwerze.

4.4.1. Okno dialogowe zatwierdzenia

Jeśli kopia robocza jest aktualna i nie ma żadnych konfliktów, jesteście gotowi, aby zatwierdzić swoje zmiany. Wybierzcie dowolne pliki lub foldery, które chcecie zatwierdzić, a następnie TortoiseSVN → Zatwierdź zmiany....



Rysunek 4.8. Dialog zatwierdzenia

Okno dialogowe zatwierdzenia pokaże zmienione pliki, w tym dodane, usunięte i niewersjonowane. Jeśli nie chcecie by zmieniony plik został zatwierdzony, wystarczy usunąć zaznaczenie tego pliku. Jeśli chcecie dołączyć plik będący do tej pory poza kontrolą wersji, wystarczy go zaznaczyć, by dodać do zatwierdzenia.

Aby szybko zaznaczyć lub odznaczyć rodzaje plików jak wszystkie wersjonowane pliki lub wszystkie zmienione pliki należy kliknąć na link tuż ponad wyświetlonych plików.

Dla uzyskania informacji o kolowaniu oraz nakładkach na elementy obrazujących ich stan, przejrzyj Sekcja 4.7.3, „Status lokalny i zdalny”.

Pozycje, które zostały przełączone do innej ścieżki repozytorium są również wskazane markerem `(s)`. Możecie mieć coś przełączone podczas pracy na gałęzi i zapomnieć przełączyć z powrotem do linii głównej. To jest wasz sygnał ostrzegawczy!

Zatwierdzić pliki czy foldery?

Kiedy zatwierdza się pliki, okno dialogowe zatwierdzenia pokazuje tylko wybrane pliki. Kiedy zatwierdzasz folder, okno dialogowe wybiera automatycznie zmienione pliki. Jeśli zapomnicie o nowo utworzonym pliku, zatwierdzenie folderu znajdzie go tak czy owak. Zatwierdzenie folderu *nie* oznacza, że każdy plik zostanie oznaczony jako zmieniony, po prostu ułatwia życie, wykonując dla was więcej pracy.

Wiele niewersjonowanych plików w oknie dialogowym zatwierdzenia

Jeśli uważacie, że okno dialogowe zatwierdzenia pokazuje zbyt wiele niewersjonowanych plików (np. generowane przez kompilator lub kopie zapasowe edytora), istnieje kilka sposobów, aby sobie z tym poradzić. Można:

- dodanie pliku (lub symbolu wieloznacznego rozszerzenia) do listy plików wykluczonych na stronie ustawień. Wpłynie to na każdą wczytaną kopię roboczą.
- dodajcie plik do listy `svn:ignore` za pomocą TortoiseSVN → Dodaj do listy ignorowanych. Będzie to miało wpływ tylko na katalog, w którym ustawiono atrybut `svn:ignore`. Korzystając z okna dialogowego atrybutów SVN możecie zmienić atrybut `svn:ignore` dla katalogu.
- dodanie pliku dolisty `svn:global-ignores` korzystając z TortoiseSVN → Dodaj do listy ignorowanych (rekurencyjnie) Będzie to dotyczyć folderu, na którym ustawiliście atrybut `svn:global-ignores` oraz wszystkich jego podfolderów.

Czytajcie Sekcja 4.14, „Ignorowanie plików i folderów” by uzyskać więcej informacji.

Dwuklik na zmodyfikowanym pliku w oknie dialogowym zatwierdzenia uruchamia zewnętrzne narzędzie porównywania, aby pokazać zmiany. Menu kontekstowe daje więcej możliwości, jak to pokazano na zrzucie ekranu. Można także przeciągnąć stąd pliki do innych aplikacji, takich jak edytor tekstu lub IDE.

Można zaznaczyć lub usunąć zaznaczenie elementu klikając pole wyboru po lewej stronie pozycji. Dla katalogów można użyć **SHIFT**-wybór by działanie wykonać rekursywnie.

Kolumny wyświetlane w dolnej części okna są konfigurowalne. Jeśli kliknie się prawym przyciskiem myszy na dowolną kolumnę z nagłówka pojawi się menu, które pozwala wybrać, które kolumny są wyświetlane. Można również zmienić szerokość kolumny przy użyciu uchwytu przeciągnięcia, który pojawia się po najechaniu wskaźnikiem myszy na granicę kolumny. Te dostosowania są zapamiętywane, więc następnym razem widać ten sam poprawiony układ pozycji.

Domyślnie, kiedy zatwierdzacie zmiany, wszystkie blokady,założone na pliki są automatycznie zwalniane gdy zatwierdzenie się powiedzie. Jeśli chcecie zachować te blokady, upewnijcie się, że pole wyboru Trzymaj blokady jest zaznaczone. Domyślny stan tego pola wyboru jest pobierany z opcji `no_unlock` w pliku konfiguracyjnym Subversion. Czytajcie Sekcja 4.31.1, „Ustawienia ogólne” by zyskać informacje na temat, jak edytować plik konfiguracyjny Subversion.

Ostrzeżenie podczas zatwierdzenia do etykiety

Zwykle zatwierdzenia wykonywane są do linii głównej lub gałęzi, nie zaś do etykiet. Ostatecznie etykieta jest uważana za stan poprawnie zdefiniowany i nie powinna się zmieniać.

Gdy następuje próba zatwierdzenia na URL etykiety, TortoiseSVN wyświetla okno dialogowe z prośbą o potwierdzenie, by zapewnić że jest to zamierzone działanie. To zaś dlatego, że takie zatwierdzenie jest wykonywane przez przypadek.

Jednak takie sprawdzenie działa tylko wtedy, gdy układ repozytorium jest jednym z sugerowanych, to jest używane są w nim nazwy `trunk`, `branches` i `tags` by wskazać podstawowe trzy obszary. W przypadku innej konfiguracji, wykrywanie co jest etykietą/gałęzią/linią główną (zwane również `wzorcami klasyfikacji`), może być skonfigurowane w oknie ustawień: Sekcja 4.31.2, „Ustawienia wykresu wersji”

Mechanizm Przeciągnij i Upuść

Możecie przeciągać pliki do okna dialogowego zatwierdzenia z zewnątrz tak długo, jak ich kopie robocze są pobierane z tego samego repozytorium. Na przykład, możecie mieć ogromną kopię roboczą z kilkoma otwartymi oknami eksploratora by przeglądać odległe foldery w hierarchii. Jeśli chcecie uniknąć zatwierdzania z katalogu najwyższego poziomu (z długim indeksowaniem folderu by sprawdzić zmiany) można otworzyć okno dialogowe zatwierdzenia jednego folderu i przeciągnąć do niego elementy z innych okien by zawrzeć je w tym samym atomowym zatwierdzeniu.

Możecie przeciągać pliki bez informacji o wersji, które znajdują się wewnątrz kopii roboczej okna dialogowego zatwierdzenia i zostaną one dodane do SVN automatycznie.

Przeciągnięcie plików z listy na dole okna dialogowego zatwierdzenia do pola edycji wiadomości dziennika wprowadza ścieżki jako zwykły tekst w tym polu edycji. Jest to przydatne jeśli chcecie napisać do dziennika opis zmiany, która zawiera ścieżki dotknięte przez zatwierdzenie.

Naprawa zmian zewnętrznych

Czasami pliki zostają przemianowane poza Subversion, i pokazują się na liście plików w postaci pliku brakującego i drugiego niewersjonowanego. Aby uniknąć utraty historii należy powiadomić Subversion na temat ich połączenia. Wystarczy wybrać zarówno starą nazwę (brakujący) i nową nazwę (bez informacji o wersji) i użyć → Napraw przeniesienie by powiązać dwa pliki jako zmianę nazwy.

Naprawa kopii zewnętrznych

Jeśli zrobiliście kopię pliku, ale zapomnieliście użyć przy tym polecenia Subversion, można naprawić tą kopię by nowy plik nie stracił swojej historii. Wystarczy wybrać zarówno starą nazwę (normalny lub zmian) i nową nazwę (bez informacji o wersji) i użyć Menu kontekstowe → Napraw kopię by powiązać dwa pliki jako kopię.

4.4.2. Listy zmian

Dialog zatwierdzenia obsługuje funkcję listy zmian Subversion, aby pomóc zgrupować powiązane pliki razem. Dowiedz się więcej o tej funkcji w Sekcja 4.8, „Listy zmian”.

4.4.3. Zatwierdzaj tylko części plików

Czasami chcecie zatwierdzić tylko część zmian, jakie wprowadziliście do pliku. Taka sytuacja powstaje zazwyczaj gdy pracujecie nad czymś a w międzyczasie musi być zatwierdzona pilna poprawka, przy czym ta pilna poprawka znajduje się w tym samym pliku, który właśnie rozgrzebaliście.

kliknijcie prawym przyciskiem myszy na pliku i użyjcie Menu kontekstowe → Przywróć po zatwierdzeniu. Spowoduje to utworzenie kopii pliku w stanie, w jakim się znajduje. Następnie możecie edytowac plik, np. w

edytorze tekstowym i wycofać wszystkie zmiany, których nie zamierzacie zatwierdzać. Po zapisaniu tych zmian można zatwierdzić plik.

Użycie TortoiseMerge

Przy użyciu TortoiseMerge do edycji pliku,można albo edytować zmiany, jak to się zwykle czyni, albo zaznaczyć szystkie zmiany do wprowadzenia. kliknijcie prawym przyciskiem myszy na zmienionym bloku i użyjcie Menu Kontekstowego → Zaznacz tę zmianę by dodać zmianę. Na koniec kliknijcie prawym przyciskiem myszy i użyjcie Menu Kontekstowego → Zostaw tylko zaznaczone zmiany co spowoduje zmianę prawego widoku na zawierający tylko zmiany przed chwilą zaznaczone i wycofać zmiany niezaznaczone.

Po wykonaniu zatwierdzenia kopia pliku zostaje automatycznie odtworzona i otrzymujecie z powrotem plik ze wszystkimi waszymi niezatwierdzonymi modyfikacjami.

4.4.4. Wyłączanie elementów z listy zatwierdzenia

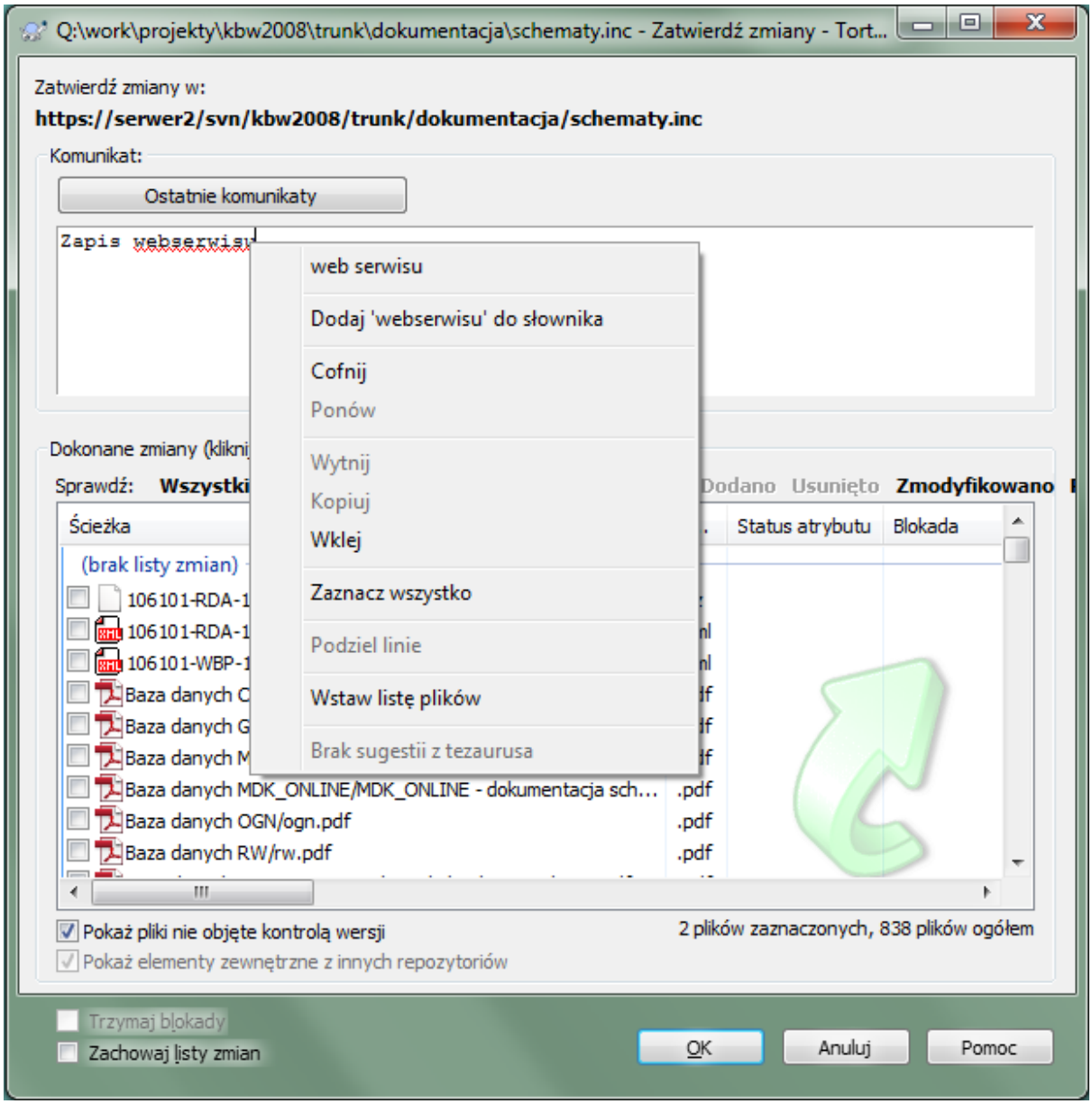
Macie czasem wersjonowane pliki, które często się zmieniają, ale tak naprawdę nie trzeba ich zatwierdzać. Czasem oznacza to błąd w procesie kompilacji - dlaczego te pliki mają być pod kontrolą wersji? może należy używać plików szablonu? Jednak czasami jest to nieuniknione. Klasycznym powodem jest, gdy IDE zmienia znacznik czasu w pliku projektu przy każdej kompilacji. Plik projektu musi być wersjonowany, ponieważ zawiera wszystkie ustawienia kompilacji, ale nie musi być zatwierdzany tylko dlatego, że zmienił się znacznik czasu.

Aby pomóc w takich niewygodnych przypadkach, mamy zarezerwowane listę zmian o nazwie `ignore-on-commit`. Każdy plik dodany do tej listy zmian zostanie automatycznie odznaczona w oknie dialogowym zatwierdzenia. Nadal możesz zatwierdzać zmiany, ale należy wybrać go ręcznie w oknie dialogowym.

4.4.5. Wiadomości dziennika zatwierdzeń

Pamiętajcie, aby podać komunikat dziennika, która opisuje Wasze zmiany. To pomoże wam zorientować się, co i kiedy się stało podczas przeglądania opisów zmian projektu w późniejszym terminie. Wiadomość może być tak długa lub krótka, jak chcecie; wiele projektów posiada wytyczne odnośnie tego co powinno być uwzględnione, używanego języka, a czasami nawet mają ścisły format.

Można zastosować proste formatowanie do opisów zmian przy użyciu konwencji podobnej do tej stosowanej w wiadomości e-mail. Aby zastosować styl do `zapis`, użyj `*zapis*` dla pogrubienia, `_zapis_` dla podkreślenia i `^zapis^` by uzyskać kursywę.



Rysunek 4.9. Sprawdzenie pisowni w oknie dialogowym zatwierdzenia

TortoiseSVN zawiera moduł sprawdzania pisowni, aby pomóc wpisywać poprawnie opisy zmian. W wszelkie błędnie napisane słowa zostają podkreślone. Użyjcie menu kontekstowego by przejść do proponowanych poprawek. Oczywiście, że nie zna on *każdego* terminu technicznego, którego używacie, więc poprawne słowa czasami pojawiają się jako błędy. Ale nie martwcie się. Wystarczy je dodać do słownika użytkownika za pomocą menu kontekstowego.

Okno opisu zmiany posiada także usprawnienie automatycznego uzupełniania nazwy pliku i funkcji. Używa wyrażeń regularnych aby wyodrębnić nazwy klas i funkcji z (tekstowych) plików, które są zatwierdzane, jak również nazwy ich samych. Jeśli wpisywane słowo pasuje do jakiegoś wpisu z listy (po wpisaniu co najmniej 3 znaków, lub wciśnięciu **Ctrl+Spacja**), pojawia się menu rozwijane pozwalając wybrać pełną nazwę. Wyrażenia regularne dostarczane z TortoiseSVN znajdują się się w folderze instalacjyjnym TortoiseSVN `bin`. Można również zdefiniować własne regeksy i przechowywać je w `%APPDATA%\TortoiseSVN\autolist.txt`. Oczywiście prywatna autolista nie zostanie nadpisana podczas aktualizacji instalacji TortoiseSVN. Jeśli nie jesteście zaznajomieni z wyrażeniami regularnymi, rzućcie okiem na wprowadzenie pod adresem https://en.wikipedia.org/wiki/Regular_expression, lub dokumentację online i samouczek pod http://www.regular-expressions.info/.

Ustalenie właściwego wyrażenia regularnego może być trudne, więc aby pomóc dojść do właściwego wyrażenia mamy okno testowe, które pozwala wprowadzić wyrażenie, a następnie wpisać w nazwach plików na których należy je przetestować. Uruchamia się je z wiersza poleceń za pomocą polecenia `TortoiseProc.exe /command:autotexttest`.

Okno komunikatu zatwierdzenia zawiera również udogodnienie podpowiedzi fragmentów komunikatu. Fragmenty są pokazywane w dymku podpowiedzi automatycznych gdy wpisze się skrót fragmentu, a zaznaczenie fragmentu w dymku wstawia jego pełny tekst. Fragmenty dostarczane z TortoiseSVN są przechowywane w podfolderze `bin` folderu instalacyjnego TortoiseSVN. Możecie również definiować własne fragmenty i trzymać w `%APPDATA%\TortoiseSVN\snippet.txt`. # jest znakiem początku komentarza. Znaki nowej linii można zapisać korzystając ze znanego z wyrażeń regularnych zapisu ze znakiem ucieczki `\n` i `\r`. By wstawić backslash należy użyć poprzedzającego ukośnika odwrotnego jako znaku ucieczki `\\`.

Możecie ponownie wykorzystać wcześniej wprowadzone opisy zmian. Wystarczy kliknąć na Ostatnie komunikaty, aby wyświetlić listę ostatnich kilku wiadomości wprowadzonych dla tej kopii roboczej. Liczbę zapisanych wiadomości można dostosować w oknie ustawień TortoiseSVN.

Można usunąć wszystkie zapisane wiadomości zatwierdzenia ze strony ustawień Zapisane dane TortoiseSVN, lub można usunąć pojedyncze wiadomości w oknie dialogowym Ostatnie komunikaty przy pomocy klawisza **Delete**.

Jeśli chcecie użyć zaznaczonych ścieżek w opisie zmiany, możecie użyć polecenia Menu kontekstowe → Wstaw listę plików w polu edycji.

Innym sposobem wprowadzania ścieżki do opisu zmiany jest po prostu przeciągnięcie plików z listy do wnętrza kontrolki.

Specjalne właściwości folderu

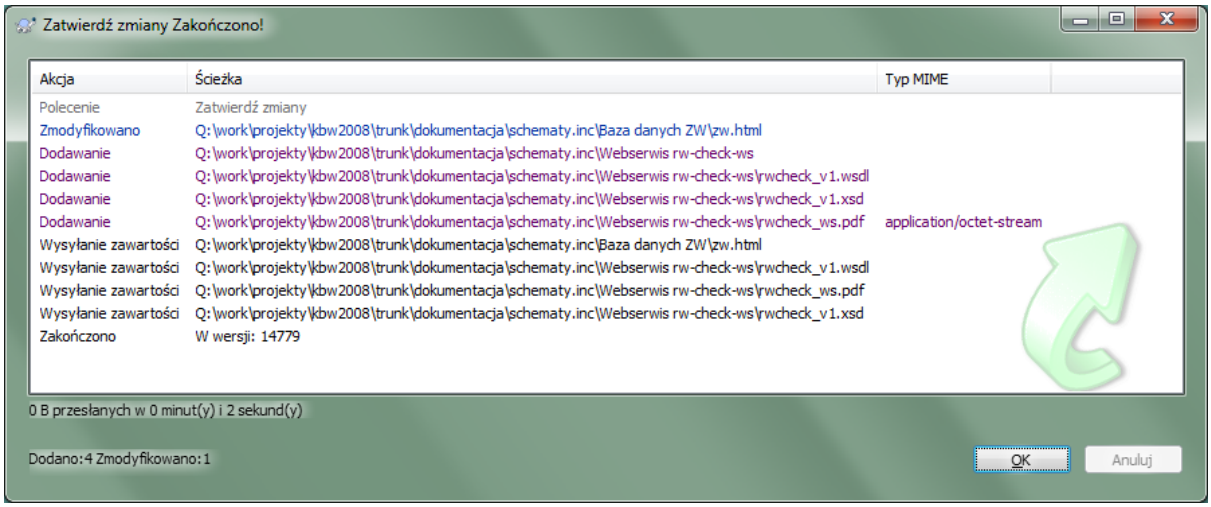
Istnieje kilka specjalnych właściwości folderu, które mogą być wykorzystany do osiągnięcia większej kontroli nad formatowaniem wiadomości dziennika zatwierdzeń i języka używanego przez moduł sprawdzania pisowni. Czytajcie Sekcja 4.18, „Ustawienia projektu” w celu uzyskania dalszych informacji.

Integracja z narzędziami śledzenia błędów

Mając działający system śledzenia błędów, można przypisać jeden lub więcej zagadnień w polu tekstowym ID błędu / Nr zgłoszenia:. Poszczególne zgłoszenia powinny być oddzielone przecinkami. Ewentualnie, jeśli używacie opartego na wyrażeniach regularnych na wsparcia śledzenia błędów, po prostu dodajcie referencje zagadnień jako część opisu zmiany. Dowiedzcie się więcej pod Sekcja 4.29, „Integracja z systemami śledzenia błędów / śledzenia problemów”.

4.4.6. Postęp zatwierdzenia

Po naciśnięciu OK pojawi się okno dialogowe wyświetlające postęp zatwierdzenia.



Rysunek 4.10. Okno dialogowe postępu pokazujące zatwierdzenie w toku

Okno dialogowe postępu wykorzystuje kodowanie kolorami, aby wyróżnić zatwierdzenia różnych działań

Niebieski
: Zatwierdzenie zmiany.

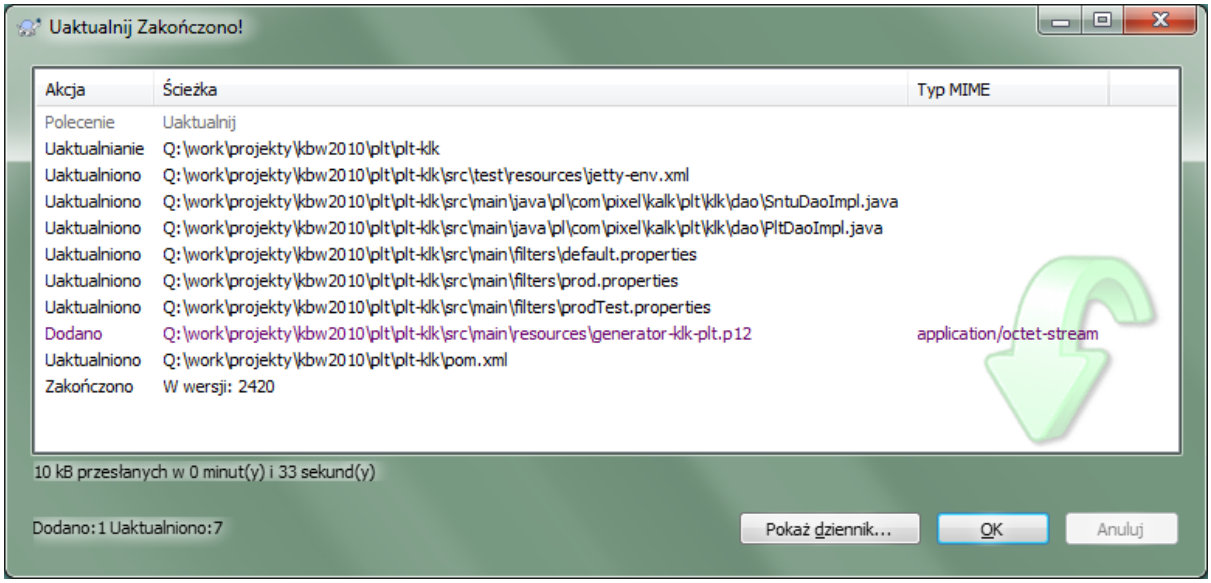
Fioletowy
: Zatwierdzenie świeżo dodanego.

Ciemnoczerwony
: Zatwierdzenie usunięcia lub zastąpienia.

Czarny
: Wszystkie inne elementy.

To jest domyślny schemat kolorów, ale można te kolory dostosować za pomocą okna dialogowego ustawień. Czytajcie Sekcja 4.31.1.5, „Ustawienia kolorów TortoiseSVN” by uzyskać więcej informacji.

4.5. Uaktualnienie kopii roboczej ze zmianami innych osób



Rysunek 4.11. Okno dialogowe postępu, pokazujące zakończenie aktualizacji

Należy okresowo zapewnić, by zmiany dokonywane przez innych zostały dołączone do lokalnej kopii roboczej. Proces uzyskania zmian z serwera do lokalnej kopii jest znany jako *uaktualnienie*. Aktualizacja może być dokonana na pojedynczych plikach, zestawie wybranych plików lub rekurencyjnie na całej hierarchii katalogów. Aby uaktualnić, wybierzcie pliki lub katalogi, które chcecie, klikając prawym przyciskiem myszy i wybierzcie TortoiseSVN → Uaktualnij w menu kontekstowym Eksploratora. Pojawi się okno wyświetlające postęp aktualizacji w trakcie jej wykonywania. Zmiany wprowadzane przez innych zostaną połączone z waszymi plikami, pozostawiając wszelkie zmiany, jakie być może dokonaliście na tych samych plikach. Repozytorium *nie* jest zmieniane podczas aktualizacji.

Okno dialogowe postępu wykorzystuje kodowanie kolorami, aby wyróżnić różne działania uaktualniania

Fioletowy
: Nowy element dodany do KR.

Ciemnoczerwony
: Zbędny element usunięty z KR lub brakujący element zastąpiony w KR.

Zielony
: Zmiany z repozytorium powodzeniem scalone ze zmianami lokalnymi.

Jasnoczerwony
: Zmiany z repozytorium scalone ze zmianami lokalnymi, co w rezultacie doprowadziło do konfliktów, które trzeba rozwiązać.

Czarny
: Niezmieniony element w KR zaktualizowany z nowszej wersji z repozytorium.

To jest domyślny schemat kolorów, ale można te kolory dostosować za pomocą okna dialogowego ustawień. Czytajcie Sekcja 4.31.1.5, „Ustawienia kolorów TortoiseSVN” by uzyskać więcej informacji.

W przypadku wystąpienia jakichkolwiek *konfliktów* podczas aktualizacji (może się to zdarzyć, jeśli inni zmienili te same linie w tym samym pliku co i Wy, i zmiany te nie pasują do siebie), a następnie w oknie dialogowym ukazuje konflikty w kolorze czerwonym. Możecie kliknąć dwukrotnie na tych liniach by uruchomić zewnętrzne narzędzie scalenia w celu rozwiązania konfliktów.

Po zakończeniu uaktualnienia, okno dialogowe postępu pokazuje zestawienie liczby elementów zaktualizowanych, dodanych, w stanie konfliktu, itp. pod listą plików. Zestawienie informacji można skopiować do schowka za pomocą **Ctrl+C**.

Standardowe polecenie Uaktualnij nie ma opcji i tylko aktualizuje kopię roboczą do wersji HEAD repozytorium, co jest najczęstszym przypadkiem użycia. Jeśli chcecie mieć większą kontrolę nad procesem aktualizacji, należy zamiast tego użyć TortoiseSVN → Uaktualnij do wersji.... Pozwala ono na aktualizację własnej kopii roboczej do konkretnej wersji, nie tylko do ostatniej. Załóżmy, że kopia robocza jest w wersji 100, ale chcecie by odzwierciedlała stan, który występował w wersji 50 - po prostu uaktualnijcie do wersji 50.

W tym samym oknie możecie także wybrać *głębokość*, na jaką zaktualizujesz bieżący folder. Użyte pojęcia zostały opisane w Sekcja 4.3.1, „Głębokość pobierania”. Głębokość domyślna to Kopia robocza, która zachowuje istniejące ustawienia głębokości. Można również ustawić głębokość `ścisły`, co spowoduje że kolejne aktualizacje będą używać tej nowej głębokości, czyli taka głębokość jest następnie wykorzystywana jako głębokość domyślna.

Aby ułatwić dołączenie lub pominięcie określonych elementów z pobrania kliknijcie przycisk Wybierz elementy.... Otworzy się nowe okno dialogowe, w którym można sprawdzić wszystkie elementy, których życzycie sobie w swojej kopii roboczej i usunąć zaznaczenie wszystkich elementów, których nie chcecie.

Możecie także wybrać, czy zignorować wszystkie zewnętrzne projekty podczas aktualizacji (tzn. projekty wymienione za pomocą odwołania `svn:externals`).

Ostrzeżenie

W przypadku aktualizacji pliku lub folderu do określonej wersji, nie należy wprowadzać zmian do tych plików. Otrzymacie komunikaty o błędach „nieaktualne” podczas próby ich zatwierdzenia! Jeśli chcecie, aby wycofać zmiany w pliku i rozpocząć na nowo od wcześniejszej wersji, możecie wycofać do poprzedniej wersji w oknie dziennika wersji. Rzućcie okiem na Sekcja B.4, „Wycofywanie (Cofnij) zmiany w repozytorium” po dalsze instrukcje i metody alternatywne.

Uaktualnij do wersji może być czasami przydatne, aby zobaczyć jak projekt wyglądał na jakimś wcześniejszym etapie swojej historii. Ale ogólnie aktualizacja poszczególnych plików do wcześniejszej wersji nie jest dobrym pomysłem, ponieważ kopia robocza pozostaje w niespójnym stanie. Jeśli plik podczas aktualizacji zmienił nazwę, można nawet stwierdzić, że plik po prostu zniknął z kopii roboczej, ponieważ nie ma pliku o tej nazwie w poprzedniej wersji. Należy również pamiętać, że pozycja pojawi się z powrotem z zieloną nakładką, więc jest nie do odróżnienia od plików, które były do tej pory aktualne.

Jeśli życzycie sobie po prostu lokalnej kopii starej wersji pliku lepiej jest użyć polecenia Menu kontekstowe → Zapisz wersję w... z okna dziennika dla tego pliku.

Wiele plików/folderów

W przypadku wybrania wielu plików i folderów w eksploratorze, a następnie wybrania Uaktualnij wszystkie te pliki/foldery są aktualizowane jeden po drugim. TortoiseSVN zapewnia, że wszystkie pliki/foldery, które są z tego samego repozytorium są aktualizowane do tej samej wersji! Nawet jeśli między tymi aktualizacjami nastąpiło inne zatwierdzenie.

4.6. Rozwiązywanie konfliktów

Raz na jakiś czas, pojawi się *konflikt* podczas uaktualnienia/scalania plików z repozytorium lub po przełączeniu kopii roboczej na inny URL. Istnieją dwa rodzaje konfliktów:

konflikt pliku
: Konflikt pliku występuje, gdy dwaj (lub więcej) programiści zmienili te same kilka linijek pliku.

konflikty drzew
: Konflikt drzewa występuje wówczas, gdy programista przeniósł /zmienił nazwę/usunął plik lub folder, który inny programista również przeniósł /zmienił nazwę/usunął lub po prostu zmodyfikował.

4.6.1. Konflikty pliku

Konflikt ma miejsce, gdy dwóch lub więcej programistów zmieniło te same kilka linijek pliku. Jako że Subversion nic nie wie o projekcie, pozostawia programistom rozwiązywanie konfliktów. Obszar konfliktu jest oznaczony w ten sposób:

```
<<<<<<< nazwaPliku
twoje zmiany
=======
kod pobrany z repozytorium
>>>>>>> wersja
```

Ponadto dla każdego konfliktu pliku Subversion wstawia w katalogu trzy dodatkowe pliki:

filename.ext.mine
: Jest to plik, jaki istniał w waszej kopii roboczej zanim ją zaktualizowaliście - czyli bez znaczników konfliktu. Ten plik zawiera Wasze najnowsze zmiany i nic więcej.

filename.ext.rOLDREV
: Jest to plik, który był wersją BAZY przed zaktualizowaniem kopii roboczej. Oznacza to, że plik, który był pobrany, zanim wprowadziliście najnowsze zmiany.

filename.ext.rNEWREV
: Jest to plik, który klient Subversion właśnie otrzymał z serwera podczas aktualizowania kopii roboczej. Ten plik odpowiada wersja HEAD repozytorium.

Można uruchomić zewnętrzne narzędzie scalenia / edytor konfliktów wywołując TortoiseSVN → Edytuj Konflikty, można też użyć dowolnego edytora tekstu w celu rozwiązania konfliktu ręcznie. Należy zdecydować, jak kod powinien wyglądać, wykonać niezbędne zmiany i zapisać plik. Skorzystanie z narzędzia scalenia, takiego jak TortoiseMerge lub jeden z innych popularnych programów jest zazwyczaj łatwiejszą opcją, jako że wyświetlają one z reguły skonfliktowany plik w 3-panelowym oknie i nie musisz się martwić o znaczniki konfliktu. Jeśli używacie edytora tekstu, to należy szukać linii zaczynających się od ciągu znaków <<<<<<<.

Następnie wykonajcie polecenie TortoiseSVN → Rozwiązany i wyślijcie swoje zmiany do repozytorium. Należy pamiętać, że polecenia Rozwiąż konflikt naprawdę nie rozwiąże konfliktu. Ono po prostu usuwa pliki `filename.ext.mine` i `filename.ext.r*`, co pozwala na zatwierdzenie własnych zmian.

Jeśli macie konflikty w plikach binarnych, Subversion nie próbuje scalić pliku. Lokalny plik pozostaje bez zmian (dokładnie tak, jak został ostatnio zmieniony) i dostajecie pliki `filename.ext.r*`. Jeśli chcecie odrzucić swoje zmiany i pozostawić wersję z repozytorium, użyjcie polecenia Wycofaj. Jeśli chcecie zachować swoją wersję i zastąpić wersję z repozytorium, należy skorzystać z polecenia Rozwiązany, a następnie zatwierdzić wersję.

Możecie użyć polecenia Rozwiązany dla wielu plików, gdy klikniecie prawym przyciskiem myszy na folder macierzysty i wybierzecie TortoiseSVN → Rozwiązany Wyświetlone zostanie okno zawierające wszystkie konflikty plików w tym folderze i można wybrać te, które zostaną oznaczone jako rozwiązane.

4.6.2. Konflikty atrybutów

Konflikt atrybutów występuje, gdy dwóch lub większa ilość programistów zmieniło ten sam atrybut. Tak jak dla zawartości pliku, rozwiązanie konfliktu może być dokonane jedynie przez programistów.

Jeśli jedna ze zmian musi nadpisać inne, wybierzcie opcję Rozwiąż używając atrybutu lokalnego lub Rozwiąż używając atrybutu zdalnego. Jeśli zmiany muszą być scalone, wybierzcie Edytuj atrybut ręcznie, wprowadźcie wartość poprawną i oznaczcie atrybut jako rozwiązany.

4.6.3. Konflikty drzewa

Konflikt drzewa występuje wówczas, gdy programista przeniósł/ zmienił nazwę/usunął plik lub folder, który w tym czasie inny programista również przeniósł/przemianował/usunął lub po prostu zmodyfikował zawartość. Istnieje wiele różnych sytuacji, które mogą doprowadzić do konfliktu drzewa, a wszystkie z nich wymagają innych czynności w celu rozwiązania konfliktu.

Gdy plik zostanie usunięty lokalnie w Subversion, jest on usuwany z lokalnego systemu plików, więc nawet jeśli jest częścią konfliktu drzewa nie można pokazać nakładki konfliktu i nie można kliknąć prawym przyciskiem myszy na plik, aby konflikt rozwiązać. Użyjcie okna dialogowego Sprawdź zmiany zamiast eksploratora by uzyskać dostęp do opcji Edytuj konflikty.

TortoiseSVN może pomagać znaleźć odpowiednie miejsce do scalenia zmian, ale z reguły niezbędna jest dodatkowa praca by konflikt rozwiązać. Pamiętajcie, że po aktualizacji BAZA robocza zawsze będzie zawierać wersję każdego elementu z repozytorium na ostatnią jego aktualizację. Zatem jeśli wycofacie zmiany, to po aktualizacji element wraca do aktualnego stanu z repozytorium, a nie tego, z którego rozpoczęliście robić własne, lokalne zmiany.

4.6.3.1. Lokalne usunięcie, przychodząca przy aktualizacji edycja

1. Programista A zmienia zawartość `Foo.c` i zatwierdza plik do repozytorium.
2. Deweloper J w tym samym czasie przeniósł `Foo.c` do `Bar.c` w kopii roboczej, lub po prostu usunął `Foo.c` albo zawierający go folder.

Uaktualnienie kopii roboczej przez programistę J daje w wyniku konflikt drzewa:

- `Foo.c` został usunięty z kopii roboczej, ale jest oznaczony jako konflikt drzewa.

- Jeżeli konflikt wynika ze zmiany nazwy, a nie usunięcia, plik `Bar.c` jest oznaczony jako dodany, ale nie zawiera poprawek programisty A.

Programista J musi teraz zdecydować, czy zachować zmiany programisty A. W przypadku zmiany nazwy plików, może on połączyć zmiany z pliku `Foo.c` do nowego `Bar.c`. Dla prostego usunięcia pliku lub katalogu może wybrać utrzymanie elementu ze zmianami programisty A i odrzucenie usunięcia. Albo, zaznaczając konflikt jako rozwiązany nie robiąc nic, skutecznie usuwa zmiany programisty A.

Dialog edycji konfliktu oferuje scalenie zmian, jeżeli można znaleźć plik oryginalny dla przemianowanego `Bar.c`. Jeśli jest wiele plików mogących być źródłem przeniesionego, przy każdym z nich pokazany zostanie przycisk pozwalający wybrać właściwy plik.

4.6.3.2. Lokalna edycja, przychodzące podczas aktualizacji usunięcie

1. Programista A przenosi `Foo.c` do `Bar.c` i zatwierdza to w repozytorium.
2. Programista J zmienia zawartość `Foo.c` w swojej kopii roboczej.

Lub w przypadku przeniesienia folderu ...

1. Programista A przenosi folder macierzysty `FooFolder` do `BarFolder` i zatwierdza to w repozytorium.
2. Programista J zmienia zawartość `Foo.c` w swojej kopii roboczej.

Uaktualnienie kopii roboczej programisty J powoduje konflikt drzewa. W przypadku prostego konfliktu pliku:

- `Bar.c` zostaje dodany do kopii roboczej jak zwykły plik.
- `Foo.c` jest oznaczony jako dodany (z historią) i posiada konflikt drzewa.

Dla konfliktu folderu:

- `BarFolder` zostaje dodany do kopii roboczej jako zwykły folder.
- `FooFolder` jest oznaczony jako dodany (z historią) oraz posiada konflikt drzewa.

  `Foo.c` jest oznaczony jako zmieniony.

Programista J musi teraz zdecydować, czy zgodzić się na reorganizację A i scalenie swoich zmian do odpowiedniego pliku w nowej strukturze, albo po prostu wycofać zmiany A i zachować swój plik lokalny.

Aby rozwiązać konflikt, programista J musi odnaleźć, na jaką nazwę pliku został przemianowany w katalogu repozytorium plik `Foo.c` będący w stanie konfliktu. Można to zrobić przy użyciu okna dialogowego dziennika. Następnie użyjcie przycisku, który pokazuje poprawny plik źródłowy by rozwiązać konflikt.

Jeśli Programista J zdecyduje, że zmiany A były niewłaściwe, musi wybrać przycisk Oznacz jako rozwiązany w oknie edytora konfliktu. Zaznacza to skonfliktowane pliki/foldery jako rozwiązane, ale zmiany programisty A trzeba usunąć ręcznie. Ponownie okno dziennika pomaga wyśledzić, co zostało przeniesione.

4.6.3.3. Lokalne usunięcie, przychodzące przy uaktualnieniu usunięcie

1. Programista A przenosi `Foo.c` do `Bar.c` i zatwierdza to w repozytorium.
2. Programista J przenosi `Foo.c` do `Bix.c`.

Uaktualnienie kopii roboczej przez programistę J daje w wyniku konflikt drzewa:

- `Bix.c` zostaje zaznaczone jako dodany z historią.
- `Bar.c` zostaje dodany do kopii roboczej ze statusem 'zwykły'.
- `Foo.c` zostaje oznaczony jako usunięty i posiada konflikt drzewa.

Aby rozwiązać konflikt, programista J musi odnaleźć, na jaką nazwę pliku został przemianowany w katalogu repozytorium plik `Foo.c` będący w stanie konfliktu. Można to zrobić przy użyciu okna dialogowego dziennika.

Następnie programista J musi wybrać który z dwóch nowych plików `Foo.c` pozostawić - ten wykonany przez programistę A, czy wykonany przez siebie poprzez zmianę nazwy.

Po tym jak programista J ręcznie rozwiązał skłócenie, konflikt drzewa musi być oznaczony jako rozwiązany za pomocą przycisku w oknie edytora konfliktu.

4.6.3.4. Lokalny brak, przyjście edycji podczas aktualizacji

1. Programista A pracując na linii głównej zmienia zawartość `Foo.c` i zatwierdza go do repozytorium
2. Deweloper J pracując na gałęzi przenosi `Foo.c` do `Bar.c` i zatwierdza to do repozytorium

Scalenie ze zmianami linii głównej programisty A do kopii roboczej gałęzi dewelopera J powoduje konflikt drzewa:

- `Bar.c` jest już w kopii roboczej ze statusem 'zwykły'.
- `Foo.c` jest oznaczony jako brakujący z konfliktem drzewa.

Aby rozwiązać ten konflikt, programista A musi oznaczyć plik jako rozwiązany w oknie edytora konfliktu, co usunie go z listy konfliktu. Potem musi zdecydować, czy skopiować brakujący plik `Foo.cz` repozytorium do kopii roboczej, czy też scalić zmiany programisty A z `Foo.c` do przemianowanego `Bar.c` czy też zignorować zmiany, zaznaczając konflikt jako rozwiązany i nie robiąc nic innego.

Zauważcie, że jeśli skopiujecie brakujący plik z repozytorium, a następnie oznaczycie jako rozwiązany, kopia zostanie usunięta. Musicie rozwiązać konflikt w pierwszej kolejności.

4.6.3.5. Lokalna edycja, przychodzące usunięcie podczas scalenia

1. Programista A pracując na linii głównej przenosi `Foo.c` do `Bar.c` i zatwierdza go w repozytorium.
2. Programista J pracując na gałęzi zmienia `Foo.c` i zatwierdza go w repozytorium.

1. Borgramista A pracując na linii głównej przenosi folder nadrzędny `FooFolder` do `BarFolder` i zatwierdza to w repozytorium.
2. Programista J pracując na gałęzi zmienia `Foo.c` w kopii roboczej.

Scalenie ze zmianami linii głównej programisty A do kopii roboczej gałęzi dewelopera J powoduje konflikt drzewa:

- `Bar.c` zostaje zaznaczony jako dodany.
- `Foo.c` zostaje zaznaczony jako zmieniony z konfliktem drzewa.

Programista J musi teraz zdecydować, czy zgodzić się na reorganizację A i scalenie swoich zmian do odpowiedniego pliku w nowej strukturze, albo po prostu wycofać zmiany A i zachować swój plik lokalny.

Aby scalić miejscowe zmiany z przetasowaniem, Programista J musi najpierw dowiedzieć się, na jaką nazwę plik skonfliktowany `Foo.c` został przemianowany/przeniesiony w repozytorium. Można to zrobić za pomocą okna dziennika dla źródła scalenia. Edytor konfliktu pokazuje tylko dziennik kopii roboczej, ponieważ nie wie, jaka ścieżka została użyta scalenia, zatem będzie musiał znaleźć ją sam. Zmiany muszą być połączone ręcznie, jako że nie ma obecnie sposobu na zautomatyzowanie lub nawet uproszczenie tego procesu. Gdy już zmiany zostały naniesione, skonfliktowana ścieżka jest zbędna i może zostać usunięta.

Jeśli programista J decyduje, że zmiana A jest niepoprawna, musi wybrać przycisk Oznacz jako rozwiązany w oknie edytora konfliktu. Zaznacza to skonfliktowany plik/folder jako rozwiązany, ale zmiany programisty A trzeba usunąć ręcznie. Ponownie okno dziennika dla źródła scalenia pomaga wyśledzić, co zostało przeniesione.

4.6.3.6. Lokalne usunięcie, przychodzące usunięcie podczas scalenia

1. Programista A pracując na linii głównej przenosi `Foo.c` do `Bar.c` i zatwierdza go w repozytorium.

2. Programista J pracując na kopii roboczej przenosi `Foo.c` do `Bix.c` i zatwierdza to do repozytorium.

Scalenie ze zmianami linii głównej programisty A do kopii roboczej gałęzi dewelopera J powoduje konflikt drzewa:

- `Bix.c` zostaje oznaczony statusem zwykły (niezmodyfikowany).
- `Bar.c` jest oznaczony jako dodany z historią.
- `Foo.c`jest oznaczony jako brakujący i ma konflikt drzewa.

Aby rozwiązać konflikt, programista J musi odnaleźć, na jaką nazwę pliku został przemianowany w katalogu repozytorium plik `Foo.c` będący w stanie konfliktu. Można to zrobić przy użyciu okna dialogowego dziennika na źródle scalenia.

Następnie programista J musi wybrać który z dwóch nowych plików `Foo.c` pozostawić - ten wykonany przez programistę A, czy wykonany przez siebie poprzez zmianę nazwy.

Po tym jak programista J ręcznie rozwiązał skłócenie, konflikt drzewa musi być oznaczony jako rozwiązany za pomocą przycisku w oknie edytora konfliktu.

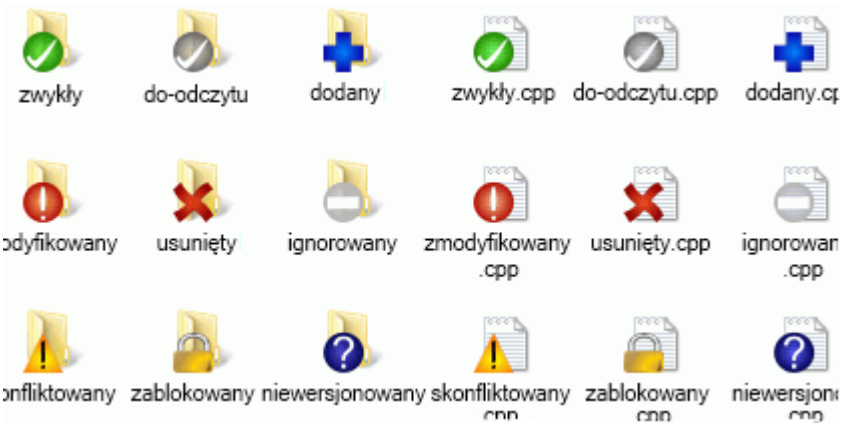
4.6.3.7. Inne konflikty drzewa

Istnieją inne przypadki, które są oznakowane jako konflikty drzewa tylko dlatego, że konflikt dotyczy folderu, a nie pliku. Na przykład, jeśli dodaliście folder o tej samej nazwie zarówno do linii głównej jak i gałęzi, a następnie spróbowaliście je scalić, otrzymacie konfliktu drzewa. Jeśli chcecie zachować folder z celu scalenia, po prostu oznaczcie konflikt jako rozwiązany. Jeśli chcecie zachować źródło scalenia, musicie najpierw SVN usunąć go z celu i dopiero uruchomić ponownie scalenie. Jeśli wymagane jest cokolwiek bardziej skomplikowanego, trzeba to rozwiązać ręcznie.

4.7. Ustalenie informacji o stanie

Podczas pracy na kopii roboczej często trzeba wiedzieć, które pliki zostały zmienione/dodane/usunięte lub przemianowane, a nawet, które pliki były zmienione i zatwierdzone przez innych.

4.7.1. Ikony nakładkowe



Rysunek 4.12. Eksplorator pokazuje nakładki ikon

Teraz, gdy już zaktualizowaliście kopię roboczą z repozytorium Subversion możecie zobaczyć swoje pliki w eksploratorze Windows ze zmienionymi ikonami. Jest to jeden z powodów, dla których TortoiseSVN jest tak popularny. TortoiseSVN dodaje tzw nakładkę ikony na każdej ikonie pliku, która częściowo przesłania oryginalną ikonę pliku. W zależności od stanu pliku w Subversion nakładka ikony jest inna.

nowozaktualizowane pliki kopii roboczej mają jako nakładkę zielony znacznik wyboru. Oznacza ona, że posiadają *zwykły* stan w Subversion.

Jak tylko zaczynacie edytować plik, status zmienia się na *zmodyfikowane* a nakładka ikony zmienia się na czerwony wykrzyknik. W ten sposób można łatwo sprawdzić, które pliki zostały zmienione od czasu ostatniej aktualizacji kopii roboczej i muszą zostać zatwierdzone.

Jeśli podczas aktualizacji ma miejsce *konflikt*, ikona zmienia się na żółty wykrzyknik.

Jeśli macie ustawiony atrybut `svn:needs-lock` na pliku, Subversion, ustawia plik w stan tylko do odczytu do czasu, aż ten plik zablokujecie. Pliki takie mają tę nakładkę aby wskazać, że trzeba nałożyć blokadę zanim można edytować plik.

Jeśli utrzymujecie blokadę na pliku, a jego status Subversion jest *zwykły*, ta nakładka ikony przypomina, że należy zwolnić blokadę jeśli pliku nie używacie, by inni mogli zatwierdzić na nim swoje zmiany.

Ta ikona wskazuje, że niektóre pliki lub foldery w folderze bieżącym zostały zaplanowane do *usunięcia* z kontroli wersji lub brakuje w folderze pliku pod kontrolą wersji.

Znak plus informuje, że plik lub folder został zaplanowany na *dodanie* do kontroli wersji.

Znak paska informuje, że plik lub folder jest *ignorowany* do celów kontroli wersji. Ta nakładka jest opcjonalna.

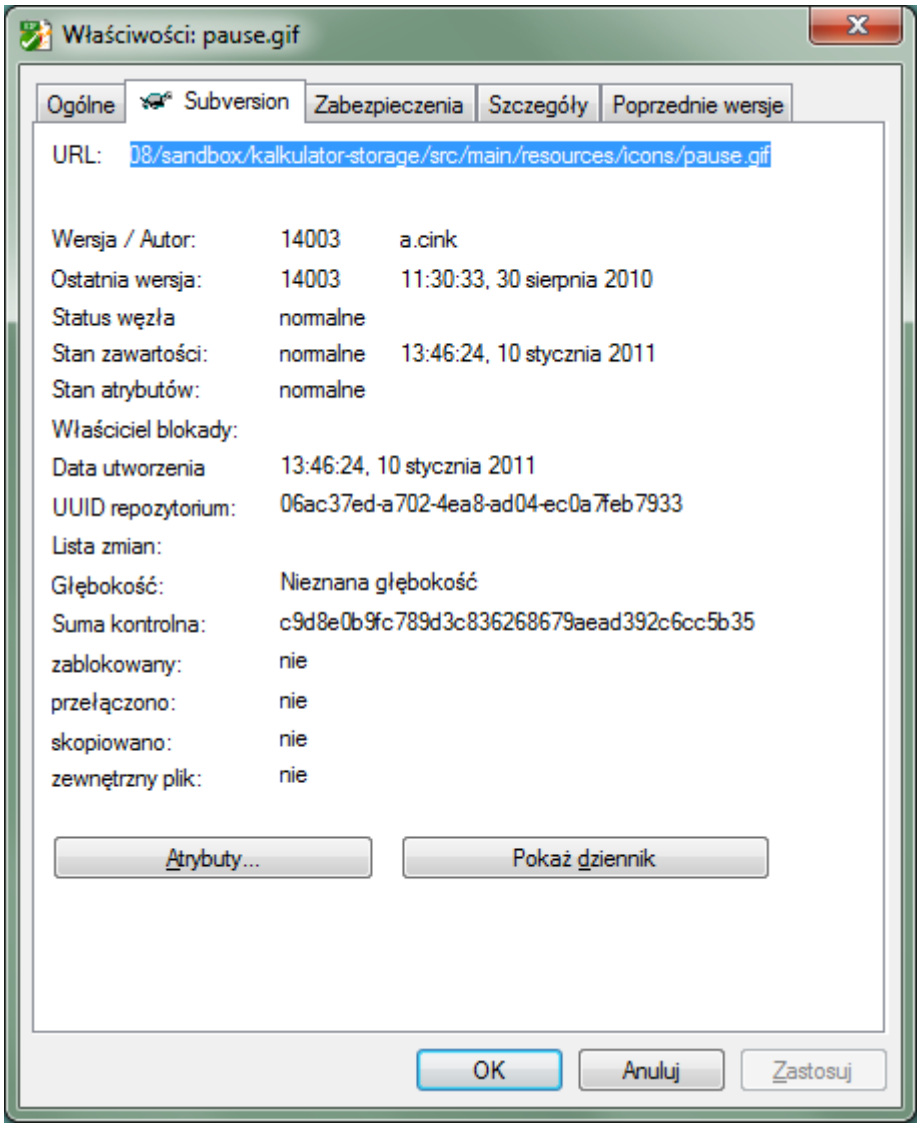
Ta ikona wskazuje pliki i foldery, które nie są pod kontrolą wersji oraz nie zostały zignorowane. Nakładka jest opcjonalna.

W rzeczywistości może się okazać, że nie wszystkie te ikony są używane w systemie. To dlatego, że liczba nakładek dozwolonych przez system Windows jest bardzo ograniczona i jeśli używacie również starszej wersji TortoiseCVS, to nie ma dostępnej wystarczającej liczby gniazd nakładek. TortoiseSVN stara się być „Dobrym Obywatelem (TM)” i ogranicza stosowanie swoich nakładek by dać szansę również innym aplikacjom.

Teraz, jako że jest wokół więcej klientów Tortoise (TortoiseCVS, TortoiseHg, ...) limit ikon staje się prawdziwym problemem. Aby go obejść, projekt TortoiseSVN wprowadził ogólnodostępny wspólny zestaw ikon, ładowany jako DLL, który może być wykorzystany przez wszystkie klienty Tortoise. Skontaktujcie się z operatorem klienta aby sprawdzić, czy ta funkcja została już włączona :-)

Aby poznać zależność pomiędzy nakładkami ikon i odpowiadającymi im stanami Subversion oraz inne szczegóły techniczne, przeczytajcie Sekcja F.1, „Ikony nakładkowe”.

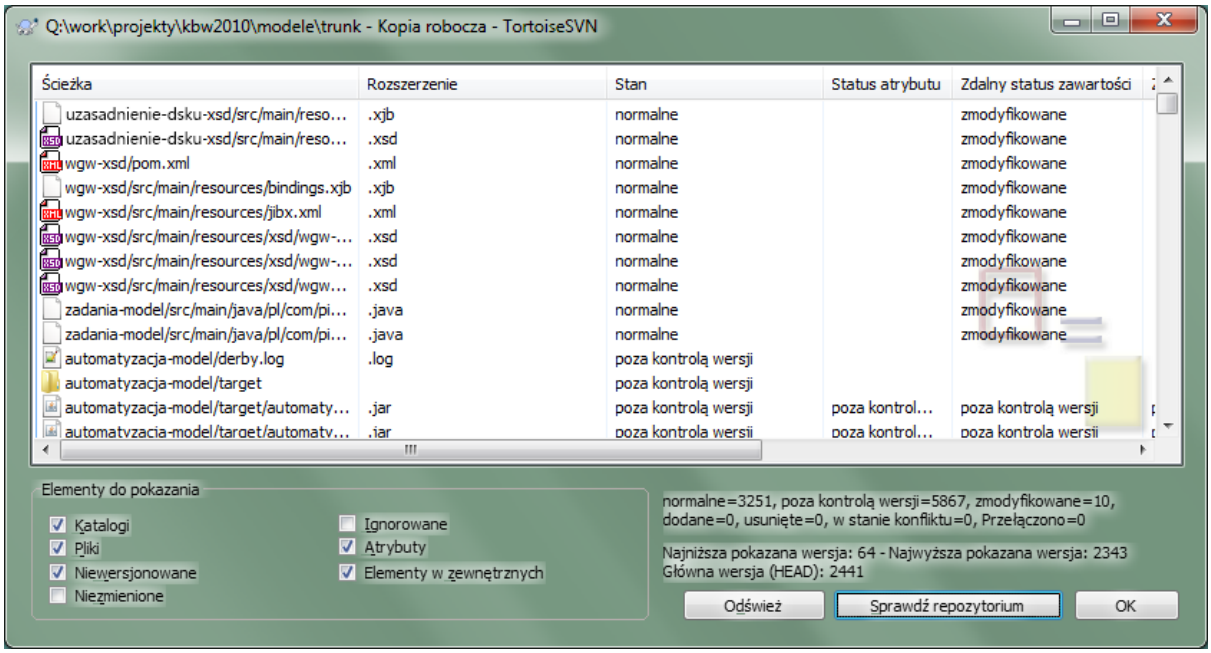
4.7.2. Status szczegółowy



Rysunek 4.13. Strona właściwości Eksploratora, zakładka Subversion

Czasami chcecie uzyskać bardziej szczegółowe informacje o pliku/katalogu niż tylko nakładka ikony. Możecie pobrać wszystkie informacje dostarczane przez Subversion w oknie dialogowym właściwości eksploratora. Wystarczy wybrać plik lub katalog i wskazać Menu kontekstowe Windows → Właściwości (uwaga: to jest normalne wejście menu właściwości obsługiwane przez eksplorator, a nie to z podmenu TortoiseSVN!). W oknie dialogowym właściwości TortoiseSVN dodaje nową zakładkę właściwości dla plików/folderów pod kontrolą Subversion, gdzie można zobaczyć wszystkie istotne informacje o wybranym pliku/katalogu.

4.7.3. Status lokalny i zdalny



Rysunek 4.14. Sprawdź zmiany

Często warto wiedzieć, które pliki się zmieniło, oraz które pliki zostały zmienione i zatwierdzone przez innych. To miejsce, gdzie przydatne staje się polecenie TortoiseSVN → Sprawdź zmiany.... To okno pokazuje wszystkie pliki, które zmieniły się w jakikolwiek sposób w kopii roboczej, jak również te, posiadane być może niewersjonowane pliki.

Jeśli klikniecie na Sprawdź repozytorium, możecie również przeglądać zmiany w repozytorium. W ten sposób można sprawdzić przed aktualizacją, czy mamy jakiś potencjalny konflikt. Można także zaktualizować wybrane pliki z repozytorium bez aktualizowania całego folderu. Domyślnie przycisk Sprawdź repozytorium odczytuje tylko zdalny status na głębokość pobrania z kopii roboczej. Jeśli chcecie zobaczyć wszystkie pliki i foldery w repozytorium, nawet te, które nie zostały pobrane, musicie przytrzymać klawisz **Shift**, podczas kliknięcia na przycisk Sprawdź repozytorium.

Okno dialogowe używa kodowanie kolorami aby zaznaczyć status.

Niebieski
: Elementy zmienione lokalnie.

Fioletowy
: Elementy dodane. Obiekty, które zostały dodane z historią posiadają znak + w kolumnie Status tekstowy, a podpowiedź pokazuje, skąd element został skopiowany.

Ciemnoczerwony
: Elementy usunięte i brakujące.

Zielony
: Pozycje zmienione lokalnie i w repozytorium. Zmiany zostaną scalone podczas aktualizacji.*Mogą* one powodować konflikty podczas aktualizacji.

Jasnoczerwony
: Pozycje zmienione lokalnie, usunięte w repozytorium lub zmienione w repozytorium i usunięte lokalnie. One *muszą* spowodować konflikty aktualizacji.

Czarny
: Elementy niezmienione i niewersjonowane.

To jest domyślny schemat kolorów, ale można te kolory dostosować za pomocą okna dialogowego ustawień. Czytajcie Sekcja 4.31.1.5, „Ustawienia kolorów TortoiseSVN” by uzyskać więcej informacji.

Ikony nakładek używane są również, by wskazywać równiez inne stany. Poniższy zrzut ekranu pokazuje wszystkie nakładki, jakie w raqzie potrzeby mogą zostać wyświetlone.graphic fileref="../images/statuslistoverlays.png"/> Nakładki są pokazywane dla następujących stanów:

- Głębokość pobierania `empty` oznacza tylko sam element.
- Głębokość pobierania `files` oznacza tylko sam element oraz pliki podrzędne bez podrzędnych folderów.
- Głębokość pobierania `immediates` określa tylko sam wskazany element, jego podrzędne pliki i foldery ale już bez elementów podrzędnych folderów podrzędnych
- Elementy zagnieżdżone, tj kopie robocze wewnątrz kopii roboczej.
- Elementy zewnętrzne, tj wszystkie elementy dodane przy pomocy atrybutu `svn:externals`.
- Elementy odtworzone op zatwierdzeniu. Zobacz Sekcja 4.4.3, „Zatwierdzaj tylko części plików” by dowiedzieć się więcej.
- Elementy mające modyfikacje atrybutu, aletylko dla `svn:mergeinfo`. Jeśli jakikolwiek inny atrybut zostanie zmieniony, nakładka nie jest używana.

Pozycje, które zostały przełączone do innej ścieżki repozytorium są również wskazane markerem `(s)`. Możecie mieć coś przełączone podczas pracy na gałęzi i zapomnieć przełączyć z powrotem do linii głównej. To jest wasz sygnał ostrzegawczy! Menu kontekstowe pozwala na przełączanie ich z powrotem do zwykłej ścieżki.

Z menu kontekstowego w oknie dialogowym można pokazać różnice zmian. Sprawdźcie lokalne zmiany wykonane przez *Was* korzystając z Menu kontekstowego → Porównaj z BASE. Sprawdźcie zmiany dokonane w repozytorium przez innych użytkowników Context Menu → Pokaż różnice w formacie Unified Diff.

Można też zawrócić zmiany w poszczególnych plikach. Jeśli usunięto plik przypadkowo, pokaże się jako *Nie znaleziono* i możecie użyć *Wycofaj zmiany*, aby go odzyskać.

Niewersjonowane i ignorowane pliki mogą być wysłane do kosza przy użyciu Menu kontekstowego → Usuń. Jeśli chcecie usunąć pliki na stałe (z pominięciem kosza) przytrzymajcie klawisz **Shift** podczas kliknięcia Usuń.

Jeśli chcecie zbadać plik szczegółowo, możecie przeciągnąć go stąd do innych aplikacji, takich jak edytor tekstowy lub IDE, albo po prostu zapisać kopię przeciągając go do folderu w eksploratorze.

Kolumny są konfigurowalne. Jeśli kliknie się prawym przyciskiem myszy na dowolny nagłówek kolumny, pojawi się menu kontekstowe pozwalające wybrać, które kolumny są wyświetlane. Można również zmienić szerokość kolumny przy użyciu uchwytu przeciągnięcia, który pojawia się po najechaniu wskaźnikiem myszy na granicę kolumny. Te dostosowania są zapamiętywane, więc następnym razem widać ten sam poprawiony układ pozycji.

Jeśli pracujecie nad kilkoma niepowiązanymi zadaniami jednocześnie, można także grupować pliki w listy zmian. Czytajcie Sekcja 4.4.2, „Listy zmian” by uzyskać więcej informacji.

W dolnej części okna można zobaczyć podsumowanie zakresu wersji w repozytorium wykorzystanego w kopii roboczej. Są to wersje *zatwierdzeń*, a nie wersje *aktualizacji*; stanowią one zakres wersji gdzie pliki zostały zatwierdzone po raz ostatni, a nie wersje do których zostały zaktualizowane. Należy pamiętać, że wyświetlany zakres wersji stosuje się tylko do elementów wyświetlanych, a nie do całej kopii roboczej. Jeśli chcecie zobaczyć, te informacje dla całej kopii roboczej należy zaznaczyć pole wyboru Pokaż niezmienione pliki.

Podpowiedź

Jeśli chcecie płaskiego widoku kopii roboczej, czyli wyświetlania wszystkich plików i folderów na każdym szczeblu hierarchii, to okno dialogowe Sprawdź zmiany jest najprostszym sposobem,

aby to osiągnąć. Wystarczy zaznaczyć pole wyboru Pokaż niezmienione pliki, aby wyświetlić wszystkie pliki w kopii roboczej.

Naprawa zmian zewnętrznych

Czasami pliki zostają przemianowane poza Subversion, i pokazują się na liście plików w postaci pliku brakującego i drugiego niewersjonowanego. Aby uniknąć utraty historii należy powiadomić Subversion na temat ich połączenia. Wystarczy wybrać zarówno starą nazwę (brakujący) i nową nazwę (bez informacji o wersji) i użyć → Napraw przeniesienie by powiązać dwa pliki jako zmianę nazwy.

Naprawa kopii zewnętrznych

Jeśli zrobiliście kopię pliku, ale zapomnieliście użyć przy tym polecenia Subversion, można naprawić tą kopię by nowy plik nie stracił swojej historii. Wystarczy wybrać zarówno starą nazwę (normalny lub zmian) i nową nazwę (bez informacji o wersji) i użyć Menu kontekstowe → Napraw kopię by powiązać dwa pliki jako kopię.

4.7.4. Przeglądanie różnic

Często chcecie zajrzeć do plików, aby zobaczyć co zmieniliście. Można to zrobić poprzez wybranie pliku, który został zmieniony i wciśnięcie Porównaj z menu kontekstowego TortoiseSVN. Spowoduje to uruchomienie zewnętrznej przeglądarki różnic, która będzie następnie porównać aktualny plik z kopią pierwotną (wersją `BAZY`), zapisaną podczas ostatniego pobrania lub aktualizacji.

Podpowiedź

Nawet na zewnątrz z kopii roboczej lub w przypadku rozrzucenia wielu wersji pliku, nadal można wyświetlać pliki różnic:

Wybierzcie dwa pliki, które chcecie porównać w eksploratorze (np. za pomocą **Ctrl** i kliknięcia myszą) i wybierzcie Porównaj z menu kontekstowego TortoiseSVN. Plik kliknięty ostatni (ten z fokusem, tj. w kropkowanym prostokącie) będzie uznany za ostatni.

4.8. Listy zmian

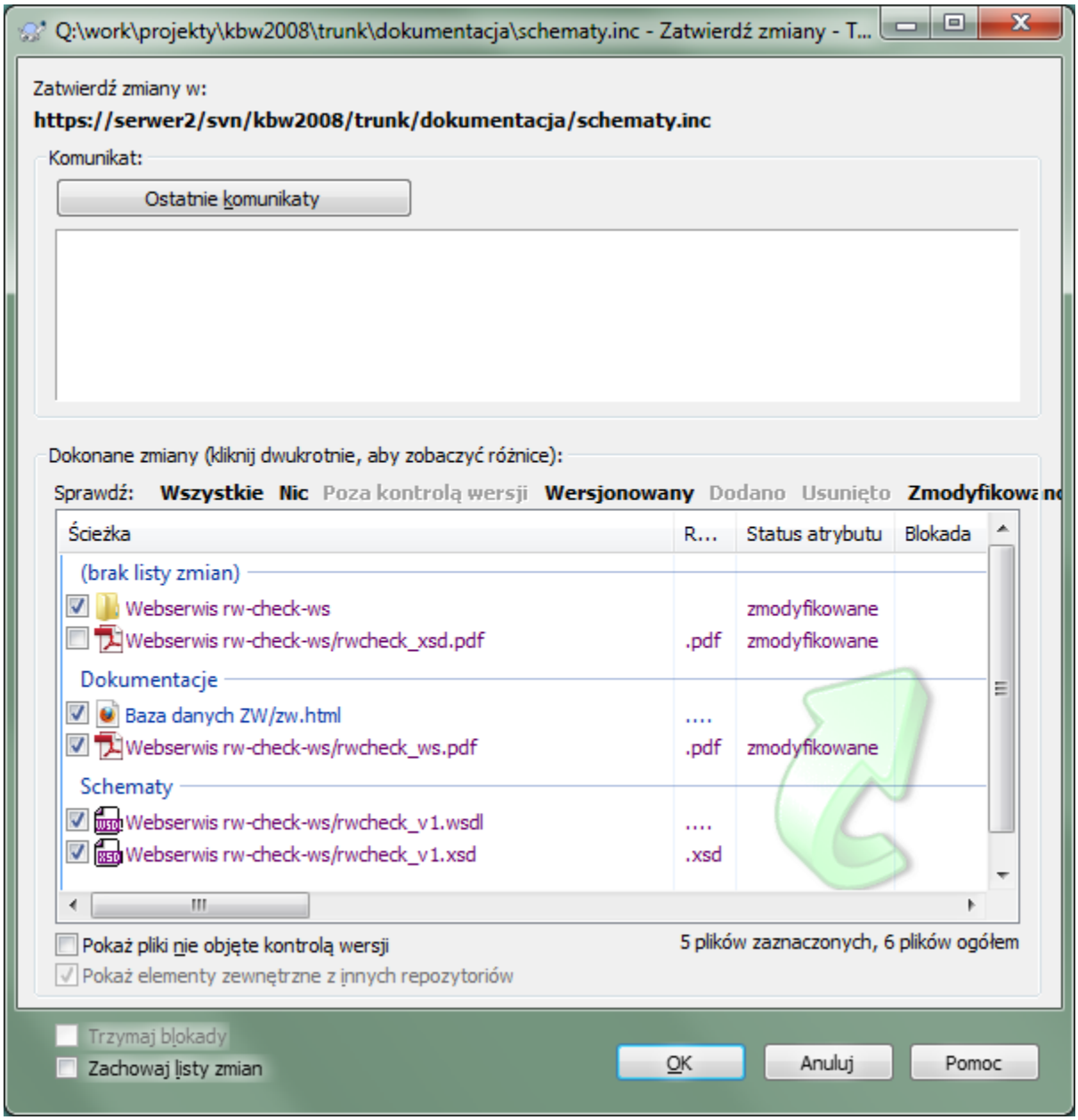
W idealnym świecie pracujecie zawsze tylko nad jedną rzeczą na raz i wasza kopia robocza zawiera tylko jeden zestaw logicznych zmian. OK, powrót do rzeczywistości. Często zdarza się, że trzeba pracować nad kilkoma niepowiązanymi zadaniami na raz, a kiedy patrzycie w dialog zatwierdzenia, wszystkie zmiany są przemieszane razem. Funkcja *lista zmian* umożliwia grupowanie plików, dzięki czemu można łatwiej zorientować się, co robicie. Oczywiście może to tylko wtedy, gdy zmiany nie zachodzą na siebie. Jeśli dwa różne zadania dotykają tego samego pliku, nie ma sposobu, aby oddzielić zmiany.

Możecie zobaczyć listy zmian w kilku miejscach, ale najważniejszymi są okna dialogowe zatwierdzenia i sprawdź-zmiany. Zacznijmy od okna dialogowego sprawdź-zmiany po wykonaniu pracy na kilka funkcjach i na wielu plikach. Przy pierwszym otwarciu okna, wszystkie zmienione pliki są wymienione razem. Załóżmy, że teraz chcecie zorganizować rzeczy i pogrupować te pliki według funkcji.

Wybierzcie jeden lub więcej plików i skorzystajcie z Menu kontekstowego → Przenieś do listy zmian, aby dodać element do listy zmian. Początkowo nie będzie list zmian, więc gdy robicie to po raz pierwszy należy utworzyć nową listę zmian. Nadajcie nazwę, opisującą to, do czego jej używacie, a następnie kliknijcie OK. Okno dialogowe zmienia się teraz, aby pokazać grupy elementów.

Po utworzeniu listy zmian można przeciągać i upuszczać do niej elementy nie należące do żadnej, przynależące do innej listy zmian, lub z eksploratora Windows. Przeciąganie z eksploratora może być przydatne, ponieważ pozwala

na dodawanie elementów do listy zmian zanim jeszcze plik zostanie zmodyfikowany. Moglibyście to zrobić również z okna dialogowego sprawdzenia modyfikacji, ale tylko przez wyświetlenie wszystkich niezmienionych plików.



Rysunek 4.15. Okno dialogowe zatwierdzenia z listami zmian

W oknie dialogowym zatwierdzenia można zobaczyć te same pliki, pogrupowane według listy zmian. Oprócz zapewnienia natychmiastowego wizualnego wskazania grupy, można również użyć pozycji grupy, aby wybrać pliki do zatwierdzenia.

TortoiseSVN rezerwuje jedną nazwę listy zmian na własny użytek, a mianowicie `ignore-on-commit`. Jest ona używana do oznaczania wersjonowanych plików, których prawie nigdy nie macie zamiaru zatwierdzić mimo że są zmieniane lokalnie. Funkcja ta jest opisana w Sekcja 4.4.4, „Wyłączanie elementów z listy zatwierdzenia”.

Kiedy zatwierdzacie pliki należące do listy zmian, to zwykle można się spodziewać, że pozostanie członkiem listy zmian nie jest już potrzebne. Stąd też domyślnie pliki są usuwane z list zmian automatycznie podczas zatwierdzenia. Jeśli chcecie zachować plik na liście zmian, należy użyć pola wyboru Zachowaj listy zmian u dołu okna dialogowego zatwierdzenia.

Podpowiedź

Listy zmian są cechą wyłącznie lokalnego klienta. Tworzenie i usuwanie list zmian nie wpływa na repozytorium, ani niczyich innych w kopii roboczych. Stanowią po prostu wygodny sposób na zorganizowanie swoich plików.

Ostrzeżenie

Należy zauważyć, że jeśli używa się listy zmian, zewnętrzne nie będą już więcej pokazywane w swojej grupie. Gdy są tam listy zmian, pliki i foldery są grupowane przez listy zmian a nie przez zewnętrzne.

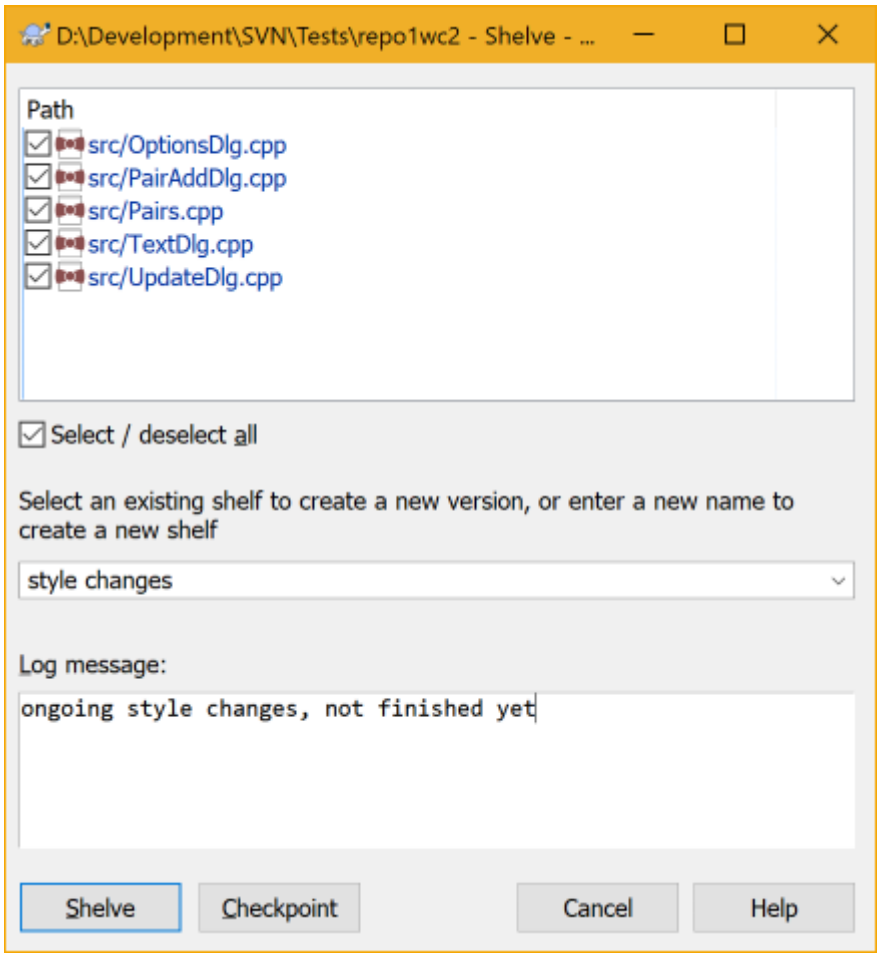
4.9. Shelving

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

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

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

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

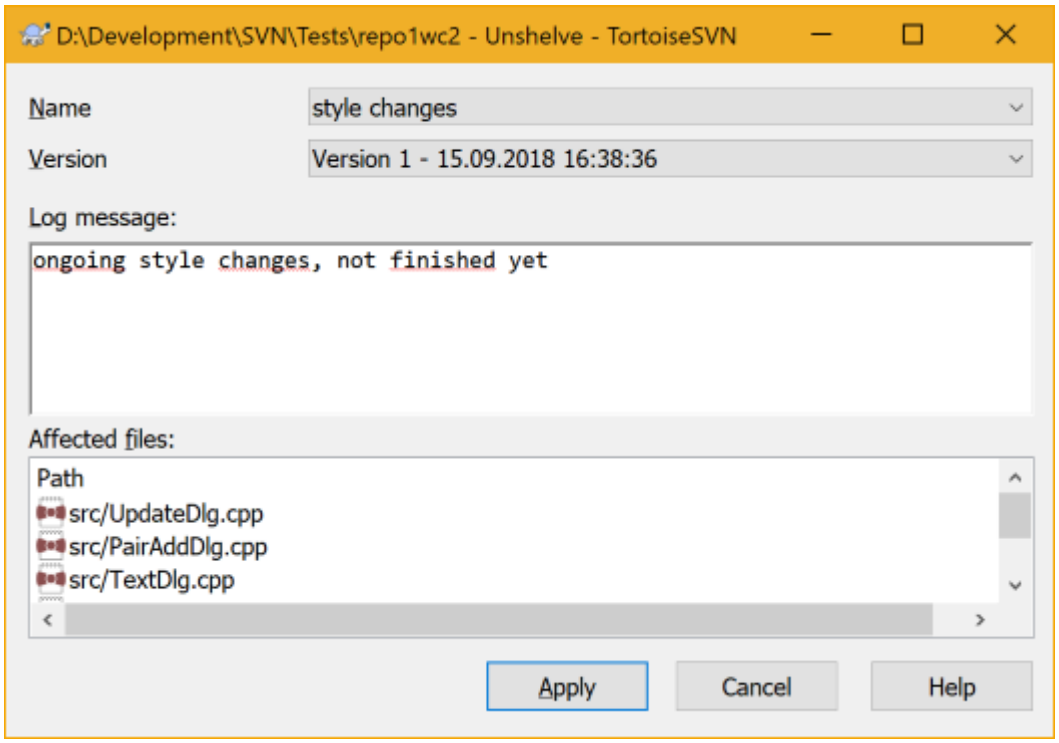


Rysunek 4.16. Shelve dialog

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

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

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



Rysunek 4.17. Unshelve dialog

Podpowiedź

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

Experimental

The shelving feature is still marked as `experimental`.

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

4.10. Okno dialogowe dziennika wersji

Dla każdej wprowadzonej i zatwierdzonej zmiany należy podać opis zmiany wpisywany do dziennika. W ten sposób można później dowiedzieć się, jakie wprowadzono zmiany i dlaczego, macie też szczegółowy dziennik procesu rozwoju.

Okno dialogowe dziennika wersji pobiera wszystkie te opisy zmian i pokazuje je Wam. Wyświetlacz jest podzielony na 3 panele.

- Górna część okna pokazuje listę wersji, w których zmiany pliku/folderu zostały zatwierdzone. Jej podsumowanie zawiera datę i czas, osobę, która zatwierdziła wersję oraz początek opisu zmiany.

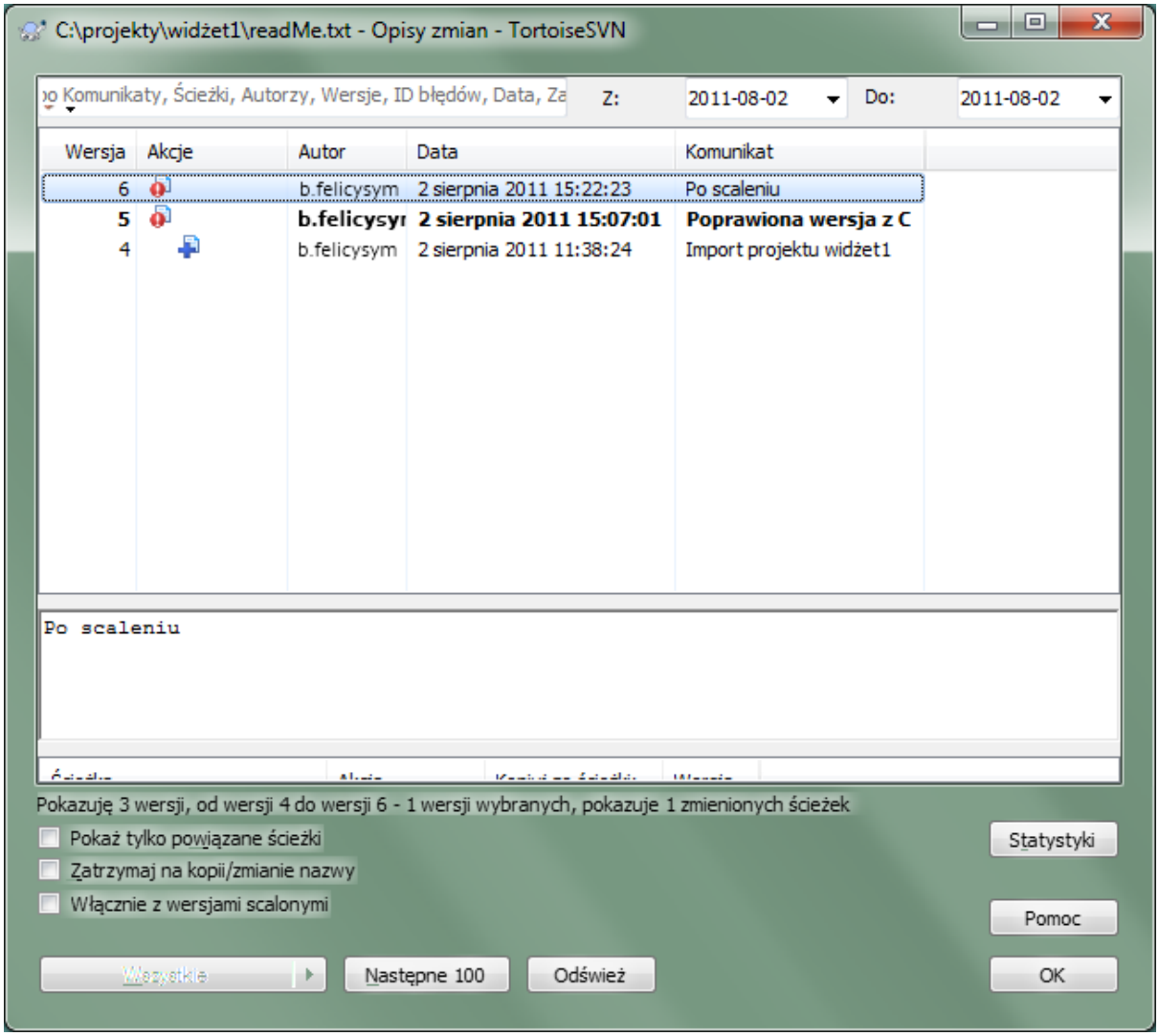
  Linie wyświetlane na niebiesko wskazują, że coś zostało skopiowane do tej linii rozwoju (być może z gałęzi).

- Środkowy panel okna pokazuje pełny opis zmian dla wybranej wersji.

- W dolnej części okna wyświetlana jest lista wszystkich plików i folderów, które zostały zmienione w ramach wybranej wersji.

Ale może ono znacznie więcej - pozwala użyć poleceń z menu kontekstowego, co można wykorzystać, aby uzyskać jeszcze więcej informacji na temat historii projektu.

4.10.1. Wywołanie okna dialogowego dziennika wersji



Rysunek 4.18. Okno dialogowe dziennika wersji

Jest kilka miejsc, z których można wyświetlić okno dialogowe dziennika:

- Z menu kontekstowego TortoiseSVN
- Ze strony atrybutów
- W oknie dialogowym stanu zaawansowania po zakończeniu aktualizacji. Później okno dialogowe dziennika pokazuje tylko te wersje, które zostały zmienione od ostatniej aktualizacji
- Z przeglądarki repozytorium

Jeśli repozytorium jest niedostępne widać okno dialogowe Chcesz przejść offline?, opisane w Sekcja 4.10.10, „Tryb offline”.

4.10.2. Akcje dziennika wersji

Górny panel posiada kolumnę Akcje zawierającą ikony, które podsumowują to, co zostało zrobione we wskazanej wersji. Istnieją cztery różne ikony, każda zajmuje własną kolumnę.

Jeśli wersja zmieniła plik lub folder, ikona *zmodyfikowano* pokazana jest w pierwszej kolumnie.

Jeśli wersja dodała plik lub folder, ikona *dodano* pokazana jest w drugiej kolumnie.

Jeśli wersja usunęła plik lub folder, ikona *usunięto* pokazana jest w trzeciej kolumnie.

Jeśli wersja zastąpiła plik lub folder, ikona *zastąpiono* pokazana jest w czwartej kolumnie.

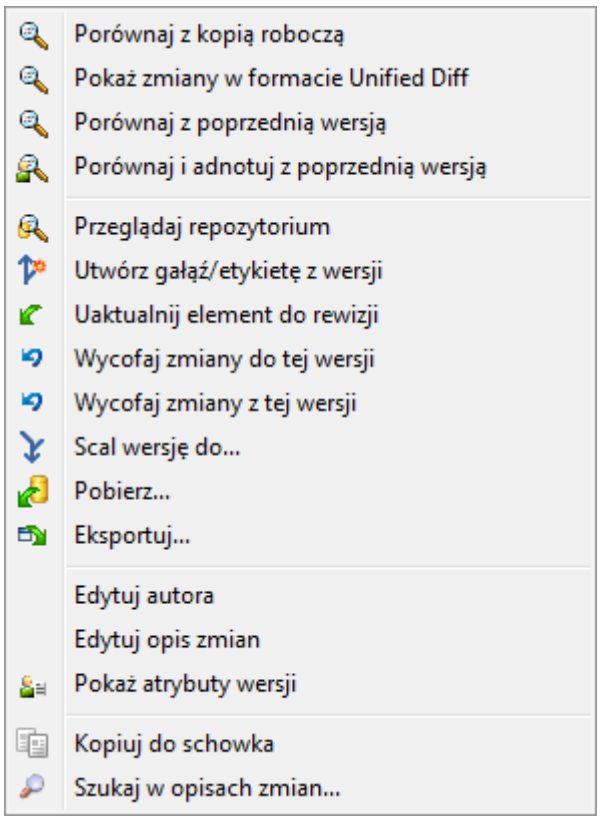
Jeśli wersja przeniosła lub zmieniła nazwę pliku lub folderu, ikona *przeniesiono* pokazana jest w pierwszej kolumnie.

Jeśli wersja zastąpiła plik lub folder przez jego przeniesienie lub zmianę nazwy, ikona *przeniesiono zastąpiono* pokazana jest w czwartej kolumnie.

Jeśli wersja scaliła plik lub folder, ikona *scalono* pokazana jest w czwartej kolumnie.

Jeśli wersja scaliła odwrotnie plik lub folder, ikona *scalono odwrotnie* pokazana jest w czwartej kolumnie.

4.10.3. Ustalenie dodatkowych informacji



Rysunek 4.19. Okno dialogowe dziennika wersji z menu kontekstowym

Górny panel okna dziennika posiada menu kontekstowe, które pozwala na dostęp do znacznie większej liczby informacji. Niektóre z pozycji menu pojawiają się tylko wtedy, gdy dziennik wyświetlany jest dla pliku, a niektóre tylko dla folderu.

Porównaj z kopią roboczą
: Pokazuje porównanie wybraną wersję z kopią roboczą. Domyślne narzędzie porównywania TortoiseMerge jest dostarczane z TortoiseSVN. Jeśli okno dziennika jest otwarte dla folderu, pokaże Wam listę zmienionych plików i pozwala na przegląd zmian wprowadzonych do każdego pliku z osobna.

Porównaj i adnotuj z roboczą BASE
: Adnotuje wybraną wersję i plik z BAZĄ roboczą oraz porównuje raporty adnotacji przy użyciu narzędzia wizualnego porównania. Czytajcie Sekcja 4.24.2, „Różnice adnotacji” by uzyskać więcej szczegółów. (tylko pliki)

Pokaż zmiany jako plik różnicowy
: Pokazuje zmiany wykonane w wybranej wersji jako plik różnicowy Unified-Diff (format poprawek GNU). Pokazuje on tylko różnice w kilku liniach kontekstu. Jest to trudniejsze do odczytania niż wizualne porównywanie plików, ale pokazuje wszystkie zmiany w pliku razem w kompaktowej formie.

  Jeśli przytrzymacie klawisz **Shift** klikając na menu, pokaże się najpierw okno, gdzie można ustawić opcje porównywania. Opcje te obejmują zdolność do ignorowania zmian końca linii i białych znaków.

Porównaj z poprzednią wersją
: Porównuje wybraną wersję z wcześniejszą wersją. Działa w sposób podobny do porównania z kopią roboczą. W przypadku folderów ta opcja będzie najpierw pokazywać okno dialogowe zmienionych plików umożliwiające wybranie plików do porównania.

Porównaj i adnotuj z poprzednią wersją
: Pokazuje okno dialogowe zmienionych plików umożliwiające wybranie plików. Adnotuje wybraną wersję, a także poprzednią wersję, i porównuje wyniki, korzystając z wizualnego narzędzia porównania. (dotyczy tylko folderów)

Zapisz wersję w...
: Zapisz wybraną wersję do pliku, w ten sposób otrzymujecie starszą wersję tego pliku. (tylko pliki)

Otwórz / Otwórz z...
: Otwiera wybrany plik, albo domyślną przeglądarką dla tego typu pliku, albo w wybranym przez Was programie. (tylko pliki)

Adnotuj...
: Adnotuje plik na wybraną wersję. (tylko pliki)

Przeglądaj repozytorium
: Otwiera przeglądarkę repozytorium by sprawdzić jak wybrany plik lub folder wyglądał na wybraną wersję w repozytorium.

Tworzy gałąź/etykietę z wersji
: Tworzenie gałęzi lub etykiety z wybranej wersji. Jest to przydatne np. jeśli zapomnieliście utworzyć etykietę i już zatwierdziliście pewne zmiany, które nie powinny znaleźć się w tym wydaniu.

Uaktualnij element do wersji
: Aktualizuje kopię roboczą do wybranej wersji. Przydatne jeśli chcecie mieć kopię roboczą odzwierciedlającą pewien okres w przeszłości, lub jeśli zostały już wykonane kolejne zatwierdzenia do repozytorium a Wy chcecie aktualizować kopię roboczą krok po kroku. Najlepiej jest zaktualizować cały katalog w kopii roboczej, a nie pojedynczy plik, w przeciwnym razie kopia robocza może być niespójna.

  Jeśli chcecie cofnąć wcześniejsze zmiany na stałe, należy zamiast tego użyć Wycofaj zmiany do tej wersji.

Wycofaj zmiany do tej wersji
: Powraca do wcześniejszej wersji. Jeśli wprowadzono kilka zmian, a następnie zdecydowaliście, że naprawdę chcecie, aby wrócić do tego, jak wszystko wyglądało w wersji N, jest to polecenie, którego potrzebujecie. Zmiany zostają cofnięte w kopii roboczej więc ta operacja *nie* wpływa na repozytorium dopóki nie

zatwierdzicie zmian. Zauważcie, że to spowoduje cofnięcie *wszystkich* zmian dokonanych po wybranej wersji przez zastąpienie pliku/folderu przez poprzednią wersję.

Jeśli kopia robocza jest w niezmienionym stanie, po wykonaniu tej czynności kopia robocza pojawi się jako zmodyfikowana. Jeśli masz już lokalne zmiany, polecenie scali *wycofane* zmian do kopii roboczej.

To co się dzieje wewnątrz opisujemy jako przeprowadzenie przez Subversion odwrotnego scalenia wszystkich zmian wprowadzonych po wybranej wersji, cofając wpływ tych ostatnich zatwierdzeń.

Jeśli po wykonaniu tej czynności zdecydujecie, że chcecie *cofnąć to wycofanie* i uzyskać dostęp do kopii roboczej z powrotem do poprzedniego niezmienionego staniu, powinniście użyć TortoiseSVN → Wycofaj zmiany w eksploratorze Windows, co odrzuci lokalne modyfikacje dokonana przez działanie odwrotnego scalenia.

Jeśli chcecie po prostu zobaczyć, jak plik lub folder wyglądał jak w poprzedniej wersji, należy zamiast tego użyć Uaktualnij do wersji lub Zapisz wersję jako....

Wycofaj zmiany z tej wersji
: Wycofanie zmian, które zostały wykonane w wybranej wersji. Zmiany są cofnięte w kopii roboczej więc ta operacja *nie* ma zupełnie wpływu na repozytorium! Zauważcie, że zostaną cofnięte zmiany wprowadzone tylko w tej wersji; nie zamieni całej kopii roboczej z plikiem na jej wcześniejszą wersję. Jest to bardzo przydatne dla wycofania wcześniejszej zmiany, gdy już jakieś inne zmiany zostały dokonane od tego czasu.

  Jeśli kopia robocza jest w niezmienionym stanie, po wykonaniu tej czynności kopia robocza pojawi się jako zmodyfikowana. Jeśli masz już lokalne zmiany, polecenie scali *wycofane* zmian do kopii roboczej.

  To co się dzieje wewnątrz opisujemy jako przeprowadzenie przez Subversion odwrotnego scalenia wszystkich zmian wprowadzonych w tej jednej wersji, wycofując jej efekty z poprzedniego zatwierdzenia.

  Możecie *cofnąć wycofanie* jak to opisano powyżej przez Wycofaj zmiany do tej wersji.

Scal wersję do...
: Scalanie wybranej(ych) wersji do innej kopii roboczej. Okno wyboru folderu pozwala wybrać kopię roboczą jako cel scalenia, ale po nim nie ma dialogu potwierdzenia, ani możliwości przetestowania wyniku scalenia. Dobrym pomysłem jest, by scalać do niezmodyfikowanej kopii roboczej, dzięki czemu można cofnąć zmiany, jeśli coś nie wyjdzie! Jest to przydatna funkcja, jeśli chcesz scalić wybrane wersje z jednej gałęzi do drugiej.

Pobierz...
: Dodaje świeże pobranie wybranego folderu na wybranej wersji. Pojawi się okienko do potwierdzenia adresu URL i wersji, a następnie wyboru lokalizacji docelowej pobrania.

Eksport...
: Eksportuje wybrane pliki/foldery w wybranej wersji. Pojawi się okienko do potwierdzenia adresu URL i wersji, a następnie wyboru lokalizacji docelowej.

Edytuj autora / opis zmian
: Edytuje opis zmian lub autora dołączone do poprzedniego zatwierdzenia. Czytajcie Sekcja 4.10.7, „Modyfikowanie opisu zmiany i autora”, aby dowiedzieć się jak to działa.

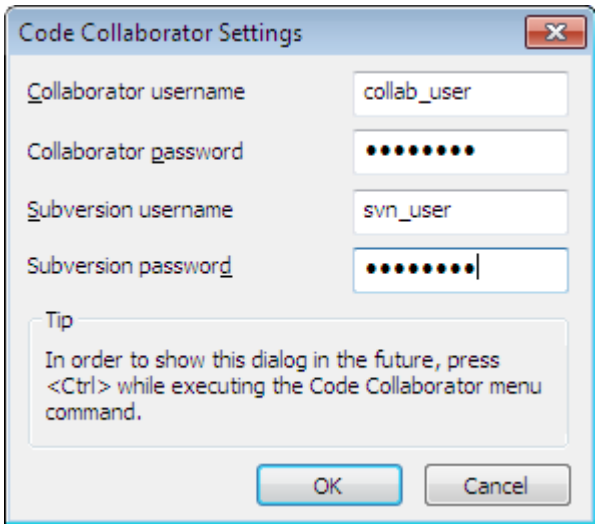
Pokaż atrybuty wersji
: Wyświetla i modyfikuje atrybut zmiany, nie tylko opis zmian i autora. Zapoznajcie się z Sekcja 4.10.7, „Modyfikowanie opisu zmiany i autora”.

Kopiuj do schowka
: Kopiowanie szczegółów dziennika z zaznaczonych wersji do schowka. Spowoduje to skopiowanie numeru wersji, autora, daty, opisu zmian i listy zmienionych elementów dla każdej wersji.

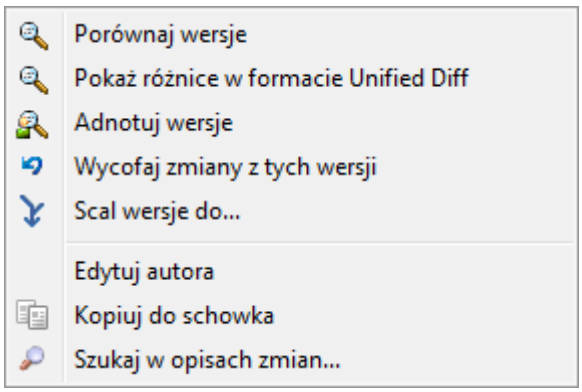
Szukaj w opisach zmian...
: Wyszukiwanie opisów zmian na podstawie wprowadzonego tekstu. Przeszukiwane są wprowadzone przez deweloperów opisy zmian, a także podsumowania akcji stworzone przez Subversion (wyświetlane w dolnym panelu). Wyszukiwanie jest niewrażliwe na wielkość liter.

Utwórz przegląd współtwórców kodu...

Manu pokazuje się jedynie gdy zainstalowane jest narzędzie współtworzenia kodu SmartBear. Przy pierwszym wywołaniu pokazuje okno dialogowe z prośbą o podanie przez użytkownika poświadczenia do narzędzia i SVN. Gdy ustawienia są zapisane, okno dialogowe nie jest pokazywane podczas wywołania menu, z wyjątkiem sytuacji w której użytkownik przytrzyma klawisz **Ctrl** podczas jego uruchomienia. Konfiguracja i wybrana rewizja(e) są używane do wywołania graficznego klienta narzędzia przeglądu kodu, w którym tworzony jest przegląd dla wybranych rewizji.



Rysunek 4.20. Okno Dialogowe Ustawień Współtwórcy Kodu



Rysunek 4.21. Menu kontekstowe górnego panelu dla dwóch wybranych wersji

Jeśli wybierzesz dwie wersje na raz (przy użyciu zwykłego modyfikatora **Ctrl**), menu kontekstowe zmieni się i pokaże mniej opcji:

Porównaj wersje

Porównuje dwie wybrane wersje używając narzędzia wizualnego porównania. Domyślną porównywarką jest TortoiseMerge, dostarczana wraz z TortoiseSVN.

Jeśli wybierzecie tę opcję dla folderu, pojawi się następne okno, a w nim lista zmienionych plików, oferując dodatkowe opcje porównania. Przeczytajcie więcej o oknie dialogowym Porównaj Wersje w Sekcja 4.11.3, „Porównanie folderów”.

Adnotuj wersje

Adnotuje dwie wersje i porównuje raporty adnotacji przy użyciu narzędzia wizualnego porównania. Przeczytajcie Sekcja 4.24.2, „Różnice adnotacji” by uzyskać dodatkowe informacje.

Pokaż różnice w formacie unified diff
: Wyświetla różnice między dwiema wybranymi wersjami wybrane jako plik różnicowy Unified-Diff. Polecenie działa dla plików i folderów.

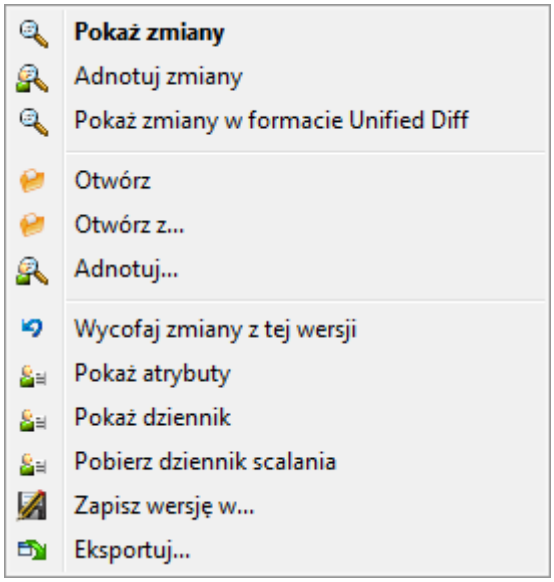
Kopiuj do schowka
: Kopiuje opisy zmian do schowka jak opisano powyżej.

Szukaj w opisach zmian...
: Szukanie w opisach zmian jak opisano powyżej.

W przypadku wybrania dwóch lub więcej wersji (przy użyciu zwykłych modyfikatorów **Ctrl** lub **Shift**), menu kontekstowe będzie zawierać pozycję do Przywróć wszystkie zmiany, które zostały dokonane w wybranych wersjach. Jest to najprostszy sposób wycofywania grupy zmian w jednym ruchu.

Możecie także wybrać scalenie wybranych wersji do innej kopii roboczej, jak opisano powyżej.

Jeżeli wszystkie wybrane wersje mają tego samego autora, można edytować autora wszystkich tych wersji w jednym kroku.



Rysunek 4.22. Dolny panel okna dialogowego dziennika wersji z menu kontekstowym

Dolny panel okna dialogowego Dziennika posiada również menu kontekstowe pozwalające na

Pokaż zmiany
: Pokazuje zmiany wykonane w wybranej wersji na wskazanym pliku.

Adnotuj zmiany
: Adnotuje wersje wybraną i poprzednią dla wybranego pliku i porównuje raporty adnotacji za pomocą wizualnego narzędzia porównywania. Czytaj Sekcja 4.24.2, „Różnice adnotacji” by uzyskać więcej szczegółów.

Pokaż w formacie unified diff
: Pokazuje zmiany pliku w standardowym formacie porównywania. To menu kontekstowe jest dostępne tylko dla plików oznaczonych jako *zmienione*.

Otwórz / Otwórz z...
: Otwiera wybrany plik, albo przy użyciu domyślnej dla tego typu pliku przeglądarki, albo w wybranym przez Was programie.

Adnotuj...
: Otwiera okno dialogowe adnotacji, pozwalając adnotować wybraną wersję.

Wycofaj zmiany z tej wersji
: Wycofuje zmiany dokonane w wybranym pliku w tej wersji.

Pokaż atrybuty
: Wyświetla atrybuty Subversion dla wybranego elementu.

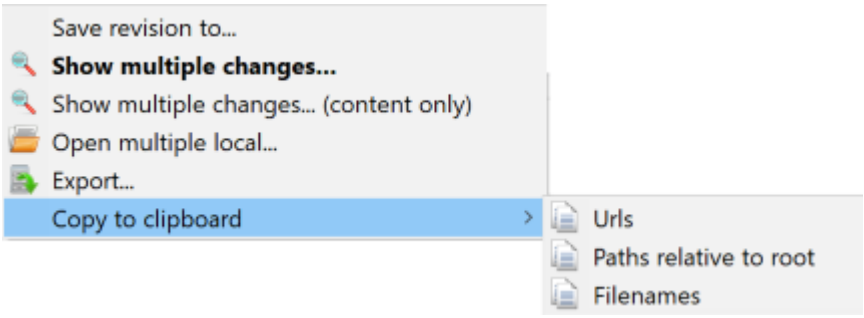
Pokaż dziennik
: Pokazuje dziennik zmian dla wybranego pojedynczego pliku.

Pobierz dziennik scalania
: Pokazuje dziennik zmian dla wybranego pojedynczego pliku, w tym scalone zmiany. Dowiedzcie się więcej w Sekcja 4.10.6, „Funkcje śledzenia scaleń”.

Zapisz wersję w...
: Zapisuje wybraną wersję do pliku, dzięki czemu macie starszą wersję tego pliku.

Eksport...
: Eksport wybranych pozycji z tej wersji do folderu, z zachowaniem hierarchii plików.

Gdy na dolnym panelu okna Dziennika zaznaczonych jest wiele plików, menu kontekstowe zmienia się na następujące:



Rysunek 4.23. Dolny panel okna Dziennika z menu kontekstowym przy wybraniu wielu plików.

Zapisz wersję w...
: Zapisuje wybraną wersję do pliku, dzięki czemu macie starszą wersję tego pliku.

Pokaż wiele zmian...
: Pokazuje zmiany wykonane w wybranej wersji w wybranych plikach. Należy zauważyć, że funkcja pokazania zmian jest wywoływana wiele razy, co może otworzyć wiele kopii wybranego narzędzia porównywania lub zwyczajnie dodać nową zakładkę porównania w porównywarce. Jeśli wybrano ponad 15 plików, zostaniecie poproszeni o potwierdzenie akcji.

Otwórz wiele lokalnych...
: Powoduje otwarcie plików z kopii roboczej odpowiadających wybranym plikom przy użyciu aplikacji przypisanych do ich rozszerzeń. [Zachowanie analogiczne do dwukliku na pliku(ach) z kopii roboczej w eksploratorze Windows]. W zależności od powiązania rozszerzeń plików z aplikacjami może się to okazać powolną operacją. W najgorszym przypadku zostaną uruchomione przez Windows nowe instancje aplikacji dla każdego z wybranych plików.

  Jeśli podczas wywoływania tego polecenia wciśnięty jest **Ctrl**, pliki kopii roboczej są wczytywane za pomocą Visual Studio. Działa to tylko gdy spełnione są następujące warunki: Visual Studio musi działać w tym samym kontekście użytkownika mając ten sam poziom integralności procesu [uruchomione jako admin lub nie], jak TortoiseProc.exe. Może być pożądane, by mieć już program zawierający już załadowane zmienione pliki, chociaż nie jest to bezwzględnie konieczne. Tylko znajdujące się na dysku pliki z rozszerzeniami [.cpp, .h, .cs, .rc, .resx, .xaml, .js, .html, .htm, .asp, .aspx, .php, .css and .xml] zostaną załadowane.Maksymalnie 100 plików może być załadowanych jednocześnie do Visual Studio, a pliki zawsze są ładowane w nowych kartach do aktualnie otwartej instancji Visual Studio. Korzyść z przeglądu zmian

kodu w Visual Studio polega na tym, że można użyć wbudowanej nawigacji, znalezienie odwołań, analizy statycznej kodu i innych narzędzi wbudowanych w Visual Studio.

Eksport...
: Eksport wybranych plików/folderu w wybranej wersji. Otwiera okno dialogowe do potwierdzenia adresu URL i wersji oraz wybrania miejsca docelowego eksportu.

Podpowiedź

Zauważyliście że czasem odwołujemy się do zmian a kiedy indziej do różnic. Jaka jest różnica?

Subversion wykorzystuje numery wersji by określić 2 różne rzeczy. Wersja zwykle oznacza stan repozytorium w pewnym punkcie czasu, ale może być również używana jako reprezentacja zestawu zmian który wprowadziła dana wersja, np. „Wprowadzone w r1234” oznacza, że zmiany zatwierdzone w r1234 realizują funkcję X. Aby stało się jasne, jakie znaczenie zostało użyte, stosujemy dwa różne terminy.

Jeśli wybierzecie dwie wersje N i M, menu kontekstowe pozwoli pokazać *różnicę* pomiędzy tymi dwiema wersjami. W kategoriach Subversion to `diff -r M:N`.

Po wybraniu jednej wersji N, menu kontekstowe pozwoli pokazać *zmiany* wykonane w tej wersji. W kategoriach Subversion to `diff -r N-1:N` lub `diff -c N`.

Dolny panel pokazuje pliki zmienione we wszystkich zaznaczonych wersjach, więc w menu kontekstowym zawsze pozwala pokazać *zmiany*.

4.10.4. Otrzymywanie dokładniejszych wiadomości dziennika

Okno dialogowe dziennika nie zawsze pokazuje wszystkie zmiany w historii z wielu powodów:

- Dla dużego repozytorium mogą być setki lub nawet tysiące zmian a pobieranie ich wszystkich może zająć dużo czasu. Zazwyczaj jesteście zainteresowani jedynie najnowszymi zmianami. Domyślnie, liczba wczytywanych wiadomości dziennika jest ograniczona do 100, ale można zmienić tę wartość w TortoiseSVN → Ustawienia (Sekcja 4.31.1.2, „Ustawienia TortoiseSVN dla okien dialogowych 1”),

- Przy zaznaczeniu pola wyboru Zatrzymaj na kopii/zmianie nazwy, Pokaż dziennik zatrzyma się w miejscu, gdy wybrany plik lub folder został skopiowany z innego miejsca w repozytorium. Może to być przydatne, gdy przeglądacie gałęzie (lub etykiety), ponieważ zatrzymuje się na folderze głównym tej gałęzi i daje szybkie wskazanie zmian tylko dla tej gałęzi.

  Zwykle chcecie zostawić tą opcję niezaznaczoną. TortoiseSVN pamięta stan pola wyboru, więc uszanuje Wasze preferencje.

  Gdy okno dialogowe Pokaż dziennik jest wywoływane z poziomu okna Scalania, pole jest zawsze domyślnie zaznaczone. To dlatego, że scalenie często przegląda zmiany w gałęziach i cofnięcie poza podstawę gałęzi nie ma sensu w tej instancji.

  Należy pamiętać, że Subversion obecnie wdraża zmianę nazwy jako kopia/usuięcie pary, więc zmiana nazwy pliku lub folderu spowoduje także zatrzymanie wyświetlenia dziennika, jeśli opcja ta jest zaznaczona.

Jeśli chcecie zobaczyć więcej opisów zmian, należy kliknąć Następne 100 by pobrać kolejne 100 opisów zmian z dziennika. Można to powtarzać tyle razy, ile potrzeba.

Obok tego przycisku znajduje się kolejny przycisk wielofunkcyjny, który pamięta ostatni sposób wykorzystania. Kliknijcie na strzałkę aby zobaczyć inne oferowane opcje.

Użyjcie Pokaż zakres... jeśli chcecie przejrzeć określony zakres zmian. Okno dialogowe wyświetli monit o podanie wersji początkowej i końcowej.

Użyjcie Wszystkie jeśli chcecie zobaczyć *wszystkie* wiadomości dziennika od HEAD z do wersji 1.

Aby odświeżyć najnowszą wersję jeśli wystąpiły inne zatwierdzenia, podczas otwarcia okna dziennika wciśnijcie klawisz **F5**.

Aby odświeżyć bufor dziennika, wciśnijcie klawisze **Ctrl-F5**.

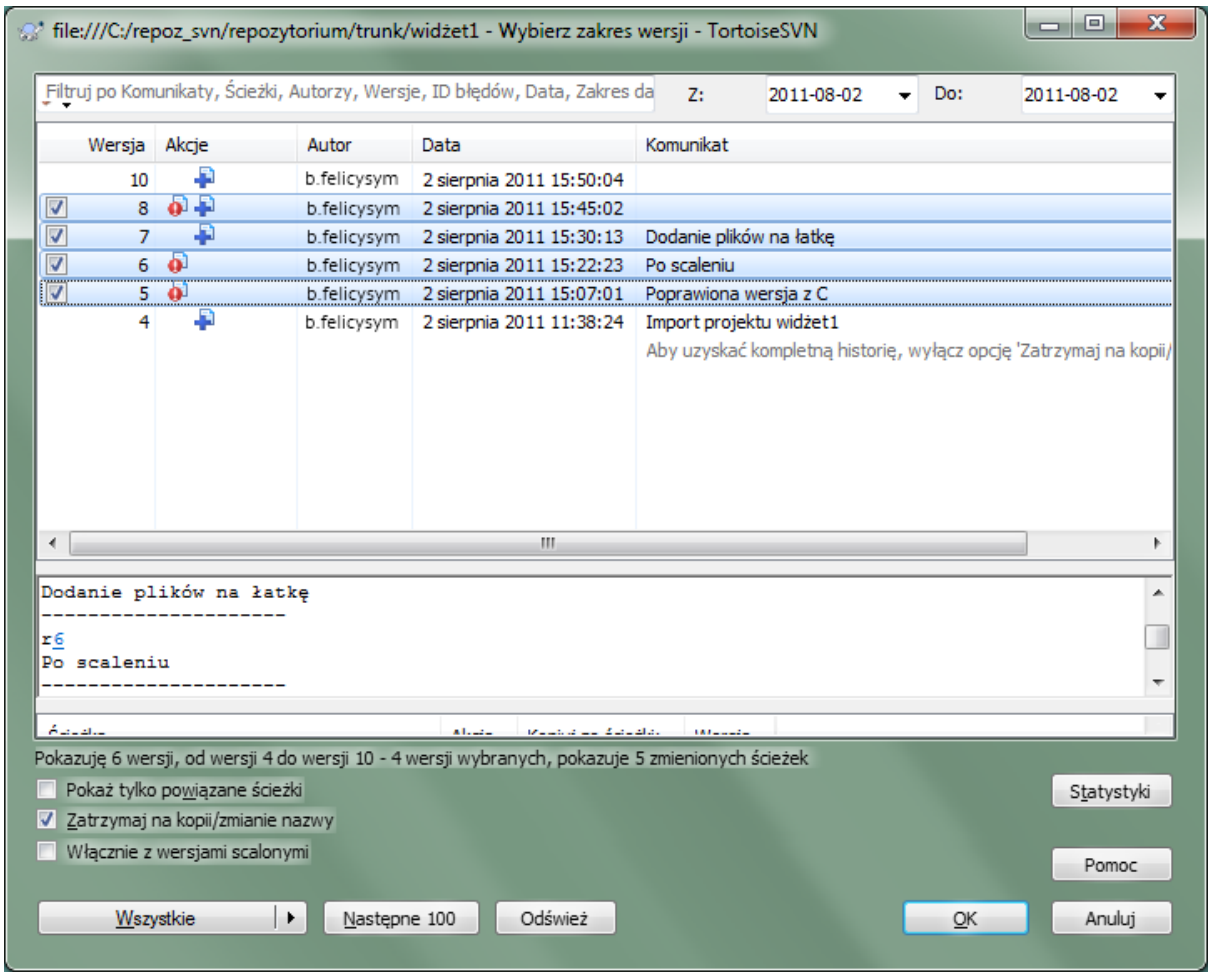
4.10.5. Bieżąca wersja kopii roboczej

Ponieważ okno dziennika pokazuje opis zmian z wersji HEAD, nie z aktualnej wersji kopii roboczej, często zdarza się, że są tam opisy dotyczące treści, które nie zostały jeszcze zaktualizowane w kopii roboczej. Aby lepiej to uwidocznić, opis zmiany, który odpowiada wersji z Waszej w kopii roboczej wyświetlany jest pogrubioną czcionką.

Kiedy pokazuje się dziennik dla folderu, zaznaczona jest najwyższa wersja znaleziona gdziekolwiek w tym folderze, co wymaga zindeksowania kopii roboczej. Indeksowanie odbywa się w osobnym wątku aby nie opóźnić wyświetlenia dziennika, jednak w powoduje to, że podświetlanie folderu może nie być widoczne od razu.

4.10.6. Funkcje śledzenia scaleń

Subversion 1.5 i kolejne rejestruje scalenia za pomocą atrybutów. Pozwala nam to uzyskać bardziej szczegółową historię scalonych zmian. Na przykład, w jeśli rozwijacie nową funkcję w gałęzi, a następnie scalacie gałąź z powrotem do linii głównej, rozwój funkcji pojawi się w dzienniku linii głównej jako jedno zatwierdzenie dla scalenia, chociaż mogło być 1000 zatwierdzeń w czasie rozwoju gałęzi.



Rysunek 4.24. Okno dziennika pokazujące wersje śledzenia scaleń

Jeśli chcecie, aby zobaczyć szczegóły, które wersje zostały scalone w ramach tego zatwierdzenia, użyjcie pola wyboru Włącznie z wersjami scalonymi. Powoduje to ponowne pobranie dziennika, tym razem przeplecione

opisami zmian z wersjami, z których pliki zostały scalone. Scalone wersje są wyświetlane w kolorze szarym, ponieważ stanowią one zmiany dokonane w innej części drzewa.

Oczywiście, scalenie nigdy nie jest proste! Podczas rozwoju funkcji na gałęzi występowały będą prawdopodobnie okazjonalne scalenia z linią główną, aby utrzymać gałąź w synchronizacji z główną linią kodu. Tak więc historia scaleń gałęzi będzie również kolejną warstwą historii scalenia. Te inne warstwy są wyświetlane w oknie dziennika przy użyciu poziomów wcięć.

4.10.7. Modyfikowanie opisu zmiany i autora

Atrybuty wersji są całkowicie różne od atrybutów Subversion dla każdego elementu. Revprops opisują elementy, które są związane z jednym konkretnym numerem wersji w repozytorium, takimi jak opis zmiany, data zatwierdzenia i nazwa zatwierdzającego (autora).

Czasami chcecie zmodyfikować opis zmiany wprowadzony już raz, może dlatego, że wystąpił w nim błąd pisowni lub chcecie poprawić wiadomość lub zmienić go z innych powodów. Albo też chcecie zmienić autora zatwierdzenia, bo zapomnieliście skonfigurować uwierzytelnianie albo...

Subversion umożliwia zmianę atrybutów wersji w dowolnym momencie. Ale ponieważ takich zmian nie można cofnąć (zmiany te nie są wersjonowane) ta funkcja jest domyślnie wyłączona. Aby zaczęła działać, należy skonfigurować przechwycenie pre-revprop-change. Prosimy odnieść się do rozdziału na temat *Skrypty przechwytujące* [http://svnbook.red-bean.com/en/1.8/svn.reposadmin.create.html#svn.reposadmin.create.hooks] w Księdze Subversion by poznać dokładnie jak to zrobić. Czytajcie Sekcja 3.3, „Skrypty przechwytujące po stronie serwera” by znaleźć dodatkowe uwagi dotyczące wykonania przechwyceń na komputerze z systemem Windows.

Po skonfigurowaniu serwera z wymaganymi przechwyceniami, można zmienić autora i opis zmiany (lub dowolną inną revprop) dowolnej wersji za pomocą menu kontekstowego z górnego panelu w oknie dziennika. Można także edytować opis zmiany przy użyciu menu kontekstowego ze środkowego panelu.

Ostrzeżenie

Ponieważ atrybuty wersji Subversion nie są wersjonowane, wprowadzenie zmian do tych atrybutów (na przykład, atrybut `svn:log` opisu zmiany) spowoduje nadpisanie poprzedniej wartości tego atrybutu *na zawsze*.

Ważne

Ponieważ TortoiseSVN trzyma bufor wszystkich informacji dziennika, zmiany w odniesieniu do autora i opisu zmiany pojawiają się tylko na lokalnej instalacji. Inni użytkownicy za pomocą TortoiseSVN nadal będą widzieli w zbuforowane (stare) zapisy autorów i opisów zmian do czasu odświeżenia bufora dziennika. Sprawdźcie w Sekcja 4.10.11, „Odświeżanie widoku”

4.10.8. Filtrowanie dziennika

Jeśli chcecie ograniczyć komunikaty dziennika, aby pokazać tylko te, które są interesujące, a nie przewijać listę z setkami pozycji, możecie użyć kontrolki filtru na górze okna dziennika. Data początkowa i końcowa pozwala ograniczyć listę pozycji do znanego zakresu dat. Pole wyszukiwania umożliwia wyświetlanie tylko wiadomości, które zawierają konkretne wyrażenie.

Kliknijcie na ikonie wyszukiwania, aby wybrać, jakie informacje chcecie odnaleźć, lub by wybrać tryb *regeksu*. Zwykle wystarczy proste wyszukiwanie podciągu, ale jeśli potrzebujecie bardziej elastycznych warunków wyszukiwania, można użyć wyrażeń regularnych. Po najechaniu myszką na pole, pojawia się podpowiedź ze wskazówkami, jak korzystać z funkcji regeksu lub podciągu. Filtr działa poprzez sprawdzenie, czy wpisany szablon pasuje do wpisów w dzienniku, a następnie tylko te pozycje, które *pasują* do szablonu są wyświetlane.

Proste wyszukiwania podciągu działa w sposób podobny do wyszukiwarki. Ciągi znaków do wyszukiwania są oddzielone spacjami, a wszystkie ciągi muszą być zgodne. Można użyć znaku wiodącego −, aby określić, że dany

podciąg ma nie zostać znaleziony (odwrócić pasujące do tego terminu), i można użyć ! na początku wyrażenia, aby odwrócić spasowanie dla całego wyrażenia. Możecie użyć wiodącego +, aby określić, że podciąg powinien być włączony, nawet jeśli wcześniej był wyłączony przez −. Należy pamiętać, że kolejność włączenia/wyłączenia jest tutaj istotna. Możecie używać cudzysłowu, aby otoczyć łańcuch, który musi zawierać spacje natomiast jeśli chcecie szukać użycia cudzysłów należy użyć dwóch cudzysłowów razem jako własnej sekwencji ucieczki. Zauważcie, że znak odwrotnego ukośnika *nie* jest znakiem ucieczki i nie ma specjalnego znaczenia w wyszukiwaniu prostego podciągu. Przykłady ułatwią zrozumienie:

```
Alicja Robert -Ewa
```

wyszukuje ciągi znaków zawierające zarówno Alicja jak i Robert, ale nie Ewa.

```
Alicja -Robert +Ewa
```

wyszukuje ciągi znaków zawierające Alicja ale nie Robert lub ciągi zawierające Ewa.

```
-Case +SpecialCase
```

wyszukuje ciągi nie zawierające Case, ale wciąż dołącza ciągi zawierające SpecialCase.

```
!Alicja Robert
```

wyszukuje ciągów, które nie zawierają ani Alicja ani Robert

```
!-Alicja -Robert
```

czy pamiętasz prawa De Morgana? NOT(NOT Alicja AND NOT Robert) redukuje się do (Alice OR Bob).

```
"Alicja i Robert"
```

wyszukuje dosłownego wyrażenia „Alicja i Robert”

```
""
```

wyszukuje cudzysłowu wszędzie w treści

```
"Alicja mówi ""cześć"" do Roberta"
```

wyszukuje dosłownego wyrażenia „Alicja mówi "cześć" do Roberta”.

Opis użycia wyszukiwania przy pomocy wyrażeń regularnych wykracza poza zakres tego podręcznika, ale można znaleźć dokumentację online i samouczek na http://www.regular-expressions.info/.

Należy pamiętać, że filtry te działają na wiadomości już dostępne. Nie kontrolują pobierania wiadomości z repozytorium.

Można także filtrować nazwy ścieżek w panelu na dole przy użyciu pola wyboru Pokaż tylko powiązane ścieżki. Powiązane ścieżki to te, które zawierają ścieżkę użytą w wyświetlanym opisie zmiany. Jeśli pobieramy dziennik folderu, oznacza to cokolwiek w tym folderze lub poniżej. W przypadku pliku, oznacza to tylko ten jeden plik. Normalnie lista ścieżek pokazuje wszystkie inne ścieżki, które są dotknięte tym samym zatwierdzeniem, ale w kolorze szarym. Jeśli pole jest zaznaczone, te ścieżki są ukryte.

Czasami metody pracy wymagają by opisy zmian były w określonym formacie, co oznacza, że tekst opisujący zmiany nie jest widoczny w skróconym podsumowaniu widocznym w górnym panelu. Przez atrybut `tsvn:logsummary` można wyodrębnić część opisu zmiany do wyświetlenia w górnym panelu. Czytajcie Sekcja 4.18.2, „Atrybuty projektu TortoiseSVN”, aby dowiedzieć się jak korzystać z tego atrybutu.

Brak Formatowania dziennika z przeglądarki repozytorium

Ponieważ formatowanie zależy od dostępu do właściwości Subversion, wyniki można zobaczyć tylko używając pobranej kopii roboczej. Pobieranie właściwości zdalnie jest działaniem powolnym, więc nie będzie widać działania tej funkcji na przeglądarkce repozytorium.

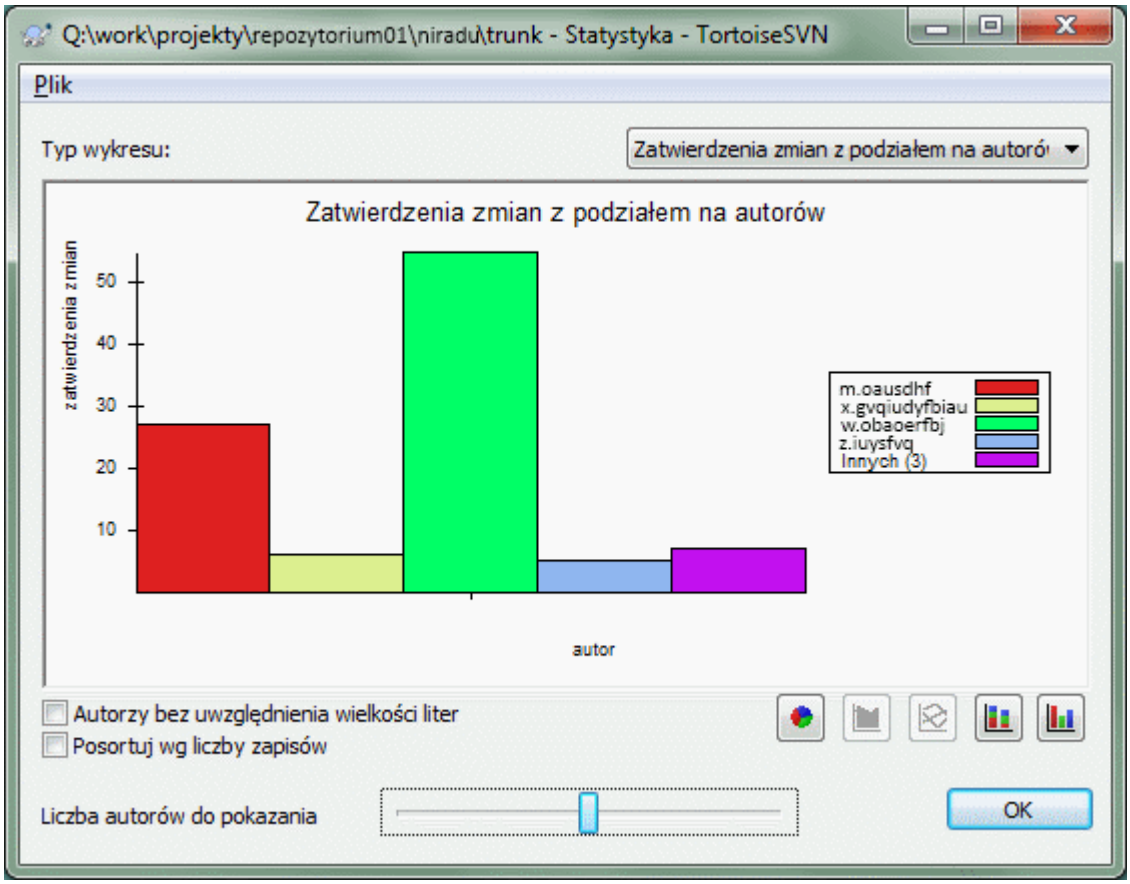
4.10.9. Informacje statystyczne

Przycisk Statystyka wyświetla pole pokazujące kilka ciekawych informacji na temat wersji przedstawionych w oknie dziennika. Pokazuje ono, jak wielu autorów było przy pracy, ile zatwierdzeń zrobili, tygodniowe postępy i wiele innych. Teraz widać na pierwszy rzut oka, który pracuje najciężej, a kto się obija ;-)

4.10.9.1. Strona statystyk

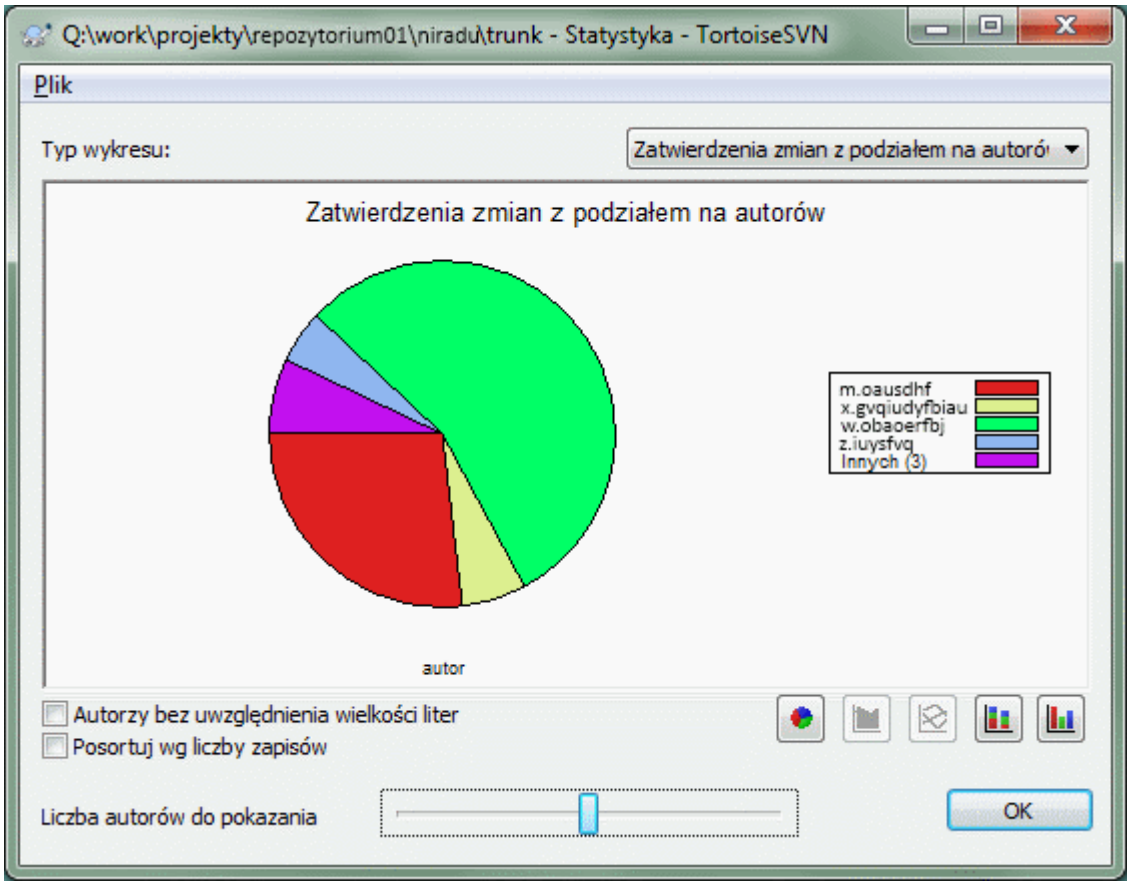
Ta strona zawiera wszystkie liczby jakie można wymyślić, w szczególności okres i liczbę objętych nim wersji, a także niektóre minima/maksima/wartości średnie.

4.10.9.2. Strona zatwierdzeń według autorów



Rysunek 4.25. Histogram zatwierdzenia-wg-autorów

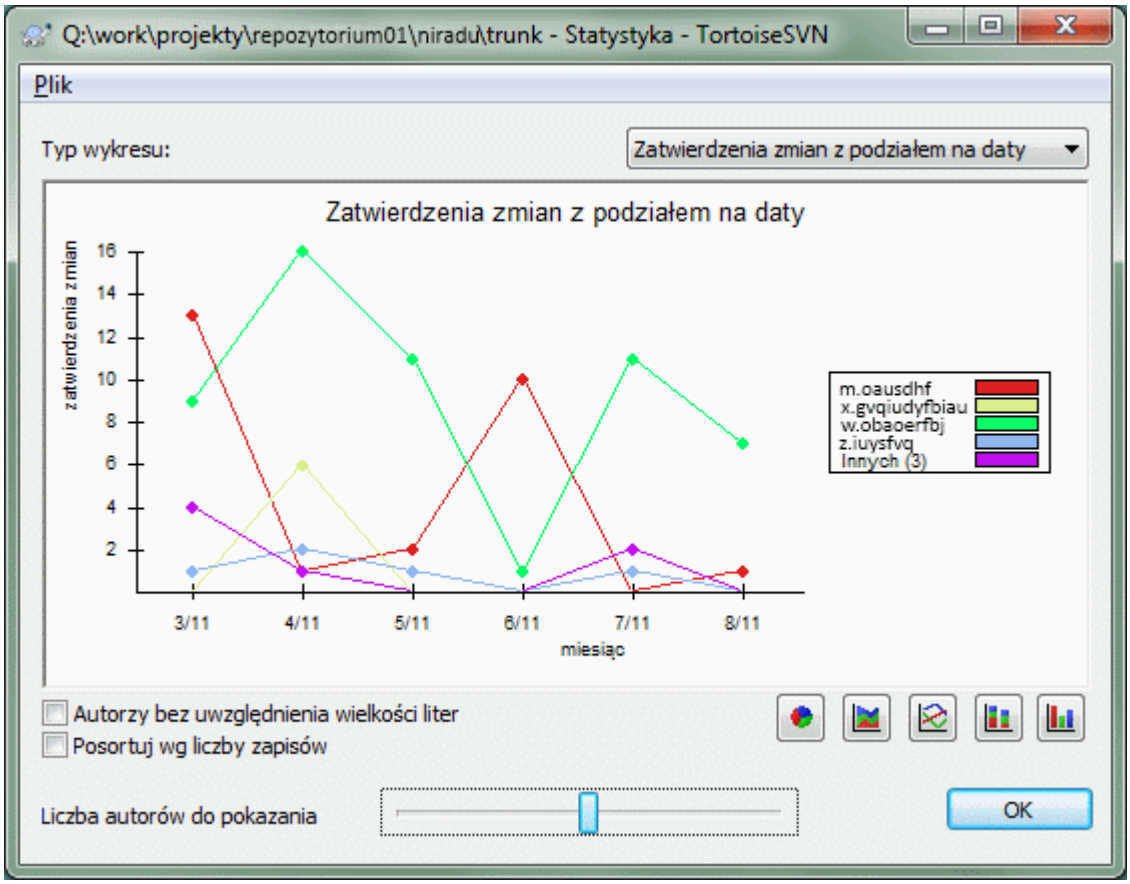
Ten wykres pokazuje, które autorzy pracują nad projektem w formie wykresu prostego słupkowego, skumulowanego słupkowego lub kołowego.



Rysunek 4.26. Wykres kołowy zatwierdzenia-wg-autorów

W przypadku, gdy istnieje kilku głównych autorów i wielu pomniejszych współpracowników, liczba małych segmentów może sprawić, że wykres bardzo stanie się trudny do odczytania. Suwak na dole pozwala na ustawienie progu (jako procent wszystkich zatwierdzeń), poniżej którego wszelkie działania są przydzielone do kategorii *Innych*.

4.10.9.3. Strona zatwierdzeń według daty



Rysunek 4.27. Wykres zatwierdzeń według daty

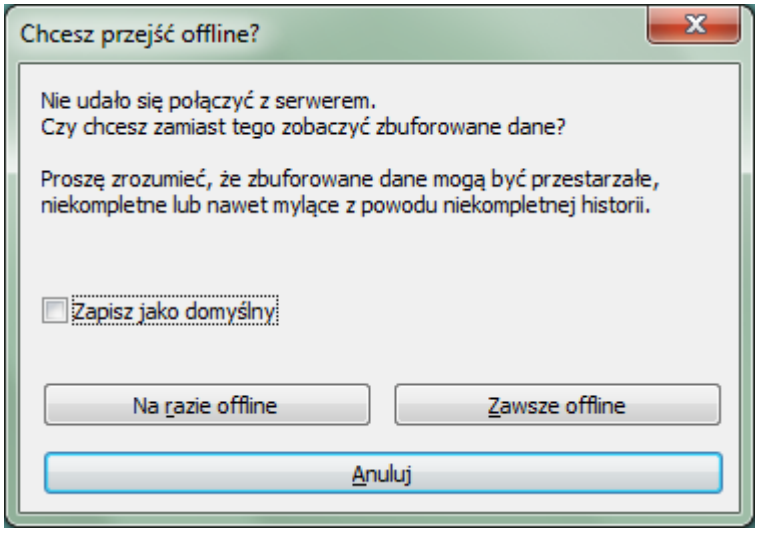
Na tej stronie znajduje się graficzna reprezentacja działań związanych z projektem pod względem liczby zatwierdzeń *i* autorów. To daje pewne wyobrażenie, kiedy pracowano nad projektem i kto pracował w jakim czasie.

Kiedy jest wielu autorów, otrzymacie dużo linii na wykresie. Istnieją dwa dostępne tutaj widoki: *normalne*, gdzie aktywność każdego autora jest względna w stosunku do linii bazowej, a *skumulowane*, gdzie aktywność każdego autora mierzona jest w stosunku do linii poniżej. Ta druga opcja pozwala na uniknięcie przecinania się linii, co czynić może wykres bardziej czytelnym, ale trudniej znaleźć pracę jednego autora.

Domyślnie analiza uwzględnia wielkość liter, więc użytkownicy `PeterEgan` i `PeteRegan` są traktowani jako różni autorzy. Jednak w wielu przypadkach nazwy użytkowników nie rozróżniają wielkości liter, a czasami są wprowadzone niekonsekwentnie, więc należy `DavidMorgan` i `davidmorgan` traktować jako tą samą osobę. Użyj pola wyboru Autorzy bez uwzględnienia wielkości liter, aby kontrolować sposób obsługi.

Należy pamiętać, że statystyki dotyczą tego samego okresu co okno dziennika. Jeśli jest ono wyświetlanie tylko dla jednej wersji, to statystyki nie powiedzą wam zbyt wiele.

4.10.10. Tryb offline



Rysunek 4.28. Okno dialogowe przejścia offline

Jeśli serwer jest nieosiągalny, a macie włączone buforowanie dziennika, możecie użyć okna dziennika i wykresu wersji w trybie offline. Używa ono danych z bufora, co pozwala na kontynuowanie pracy mimo że informacje mogą nie być aktualne lub nawet pełne.

Są tutaj trzy opcje:

Na razie offline
: Wykonuje bieżącą operację w trybie offline, ale spróbuje połączyć się do repozytorium następnym razem, gdy wymagane są dane dziennika.

Zawsze offline
: Pozostanie w trybie offline do czasu żądane jest sprawdzenie repozytorium. Zobaczcie Sekcja 4.10.11, „Odświeżanie widoku”.

Anuluj
: Jeśli nie chcecie kontynuować operacji na prawdopodobnie nieaktualnych danych, po prostu anulujcie.

Pole wyboru Zapisz jako domyślny zapobiega ponownemu pojawieniu się tego okna i zawsze stosuje opcję, którą teraz wybierzecie. Możecie jeszcze zmienić (lub usunąć) to domyślne ustawienie w korzystając z TortoiseSVN → Ustawienia.

4.10.11. Odświeżanie widoku

Jeśli chcecie ponownie sprawdzić serwer czy są nowe opisy zmian, można po prostu odświeżyć widok używając przycisku **F5**. Jeśli używacie bufora dziennika (domyślnie włączonego), repozytorium zostanie sprawdzone czy są nowe opisy i pobierane są tylko nowe. Jeśli bufor dziennika w trybie offline, nastąpi próba powrotu do trybu online.

Jeśli używacie buforu dziennika i uważacie, że treści opisu lub autor mógł się zmienić, można użyć **Shift-F5** lub **Ctrl-F5** by ponownie pobrać wyświetlane opisy z serwera i zaktualizować bufor dziennika. Zauważcie, że dotyczy to tylko wiadomości aktualnie wyświetlanych i nie unieważnia całej zawartości bufora dla tego repozytorium.

4.11. Przeglądanie różnic

Jednym z najczęstszych wymagań w rozwoju projektu jest możliwość przeglądu modyfikacji. Warto przejrzeć różnice między dwiema wersjami tego samego pliku, czy też różnice między dwoma oddzielnymi plikami.

TortoiseSVN posiada wbudowane narzędzie o nazwie TortoiseMerge do przeglądania różnic w plikach tekstowych. Do oglądania różnic plików graficznych, TortoiseSVN posiada również narzędzie o nazwie TortoiseIDiff. Oczywiście, można użyć własnego ulubionego programu porównującego, jeśli chcecie.

4.11.1. Różnice pliku

Zmiany lokalne
: Jeśli chcecie zobaczyć, jakie zmiany *sami* dokonaliście w kopii roboczej, wystarczy skorzystać z menu kontekstowego eksploratora i wybrać TortoiseSVN → Porównaj.

Różnica w stosunku do innej gałęzi/etykiety
: Jeśli chcecie zobaczyć co się zmieniło w linii głównej (podczas gdy pracujecie na gałęzi) lub konkretnej gałęzi (jeśli pracujecie na linii głównej), możecie skorzystać z menu kontekstowego eksploratora. Wystarczy przytrzymać klawisz **Shift**, podczas kliknięcia prawym przyciskiem myszy na pliku. A następnie wybrać TortoiseSVN → Porównaj z URL. W następnym oknie należy określić URL w repozytorium, z którym chcecie porównać lokalny plik.

  Można również użyć przeglądarki repozytorium i wybrać dwa drzewa do porównania, można dwie etykiety, lub gałąź/etykietę i linię główną. Menu kontekstowe pozwala porównać je za pomocą Porównaj wersje. Czytajcie więcej w Sekcja 4.11.3, „Porównanie folderów”.

Różnice w stosunku do wersji poprzedniej
: Jeśli chcecie zobaczyć różnicę między określoną wersją i kopią roboczą, należy użyć okna dialogowego Dziennika wersji, wybierzcie interesującą Was wersję, a następnie opcję Porównaj z kopią roboczą z menu kontekstowego.

  Jeśli chcecie zobaczyć różnicę między ostatnią ostatnio zatwierdzoną wersją i kopią roboczą, przy założeniu, że kopia robocza nie została zmodyfikowana, wystarczy kliknąć prawym przyciskiem myszy na pliku. Następnie wybrać TortoiseSVN → Porównaj z poprzednią wersją. Zostanie wykonane porównanie między wersją przed "datą ostatniego zatwierdzenia" (jaką zapisano w kopii roboczej) oraz BAZĄ roboczą. Zostaną wyświetlone ostatnie zmiany wprowadzone do tego pliku by doprowadzić go do bieżącego stanu widocznego w kopii roboczej. Nie pokażą się zmiany nowsze niż w kopii roboczej.

Różnice pomiędzy dwiema wcześniejszymi wersjami
: Jeśli chcecie zobaczyć różnicę pomiędzy dwiema wersjami, które są już zatwierdzone, skorzystajcie z okna dialogowego Dziennika wersji i wybierzcie dwie wersje, które chcecie porównać (przy użyciu zwykłego modyfikatora **Ctrl**). Następnie wybierzcie Porównaj wersje z menu kontekstowego.

  Jeśli zrobiliście to z dziennika wersji dla folderu, pojawi się okno dialogowe Porównaj Wersje, pokazując listę zmienionych plików w tym folderze. Czytajcie więcej w Sekcja 4.11.3, „Porównanie folderów”.

Wszystkie zmiany dokonane w zatwierdzeniu
: Jeśli chcecie zobaczyć zmiany wprowadzone do wszystkich plików w określonej wersji w jednym widoku, można użyć wyjścia Unified-Diff (format poprawek GNU). Pokazuje on tylko różnice z kilkoma liniami kontekstu. Jest to trudniejsze do odczytania niż w wizualnej porównywarce plików, ale pokazuje wszystkie zmiany razem. W oknie dialogowym Dziennika wersji wybierzcie interesującą Was wersję, a następnie wybierzcie Pokaż różnice w formacie Unified Diff z menu kontekstowego.

Różnice między plikami
: Jeśli chcecie zobaczyć różnice między dwoma różnymi plikami, możecie to zrobić bezpośrednio w eksploratorze wybierając oba pliki (za pomocą zwykłego modyfikatora **Ctrl**). Następnie z menu kontekstowego Eksploratora wybierzcie TortoiseSVN → Porównaj.

  Jeśli pliki do porównwnia nie znajdują się w tym samym folderze, użyjcie polecenia TortoiseSVN → Porównaj później by zaznaczyć pierwszy plik do porównania, następnie przejdźcie w eksploratorze do drugiego pliku i skorzystajcie z TortoiseSVN → Porównaj z "ścieżka/do/zaznaczonego/plku". By usunąć

zaznaczony plik, należy użyć polecenia TortoiseSVN → Porównaj później raz jeszcze, ale wciskając jednocześnie klawisz modyfikujący **Ctrl** podczas kliknięcia.

Różnica między plikiem/folderem KR i URL
: Jeśli chcecie zobaczyć różnice pomiędzy plikiem z kopii roboczej, a plikiem w dowolnym repozytorium, możecie to zrobić bezpośrednio w Eksploratorze wybierając plik następnie przytrzymując klawisz **Shift**, podczas kliknięcia prawym przyciskiem myszy by uzyskać menu kontekstowe. Wybierzcie TortoiseSVN → Porównaj z URL. Możecie zrobić to samo dla folderu w kopii roboczej. TortoiseMerge pokazuje te różnice w ten sam sposób, jak plik poprawek - lista zmienionych plików, które można zobaczyć po jednym na raz.

Różnice z informacjami adnotacji
: Jeśli chcecie zobaczyć nie tylko różnice, ale także autora, wersję i datę, kiedy zmiany zostały wprowadzone, można połączyć porównanie i raporty adnotacji w oknie dialogowym dziennika wersji. Czytajcie Sekcja 4.24.2, „Różnice adnotacji” by poznać więcej szczegółów.

Różnice między folderami
: Wbudowane narzędzia dostarczane z TortoiseSVN nie obsługują wyświetlania różnic między hierarchiami katalogów. Ale jeśli macie zewnętrzne narzędzie, które ma takie możliwości, możecie go użyć w tym miejscu. W Sekcja 4.11.6, „Zewnętrzne narzędzia porównywania/scalania” mówimy o kilku narzędziach, których używaliśmy.

Jeśli skonfigurowano zewnętrzne narzędzie porównywania, można użyć **Shift** podczas wybierania polecenia Porównaj by skorzystać z alternatywnych narzędzi. Czytajcie Sekcja 4.31.5, „Ustawienia programów zewnętrznych”, aby dowiedzieć się na temat konfigurowania innych narzędzi porównania.

4.11.2. Opcje końca linii i białych znaków

Czasem w życiu danego projektu można zmienić znaki końca linii z `CRLF` do `LF`, lub możecie zmienić wcięcia sekcji. Niestety działanie to oznaczy dużą liczbę linii jako zmienioną, chociaż nie wprowadza żadnych zmian znaczenia dla kodu. Wymienione tu opcje pomagają zarządzać takimi zmianami, jeśli chodzi o porównanie i stosowanie różnic. Zobaczycie te ustawienia w oknach dialogowych Scalanie i Adnotuj, jak również w ustawieniach TortoiseMerge.

Ignoruj zakończenia linii wyklucza zmiany, które wynikają wyłącznie z powodu różnicy w znaku końca linii.

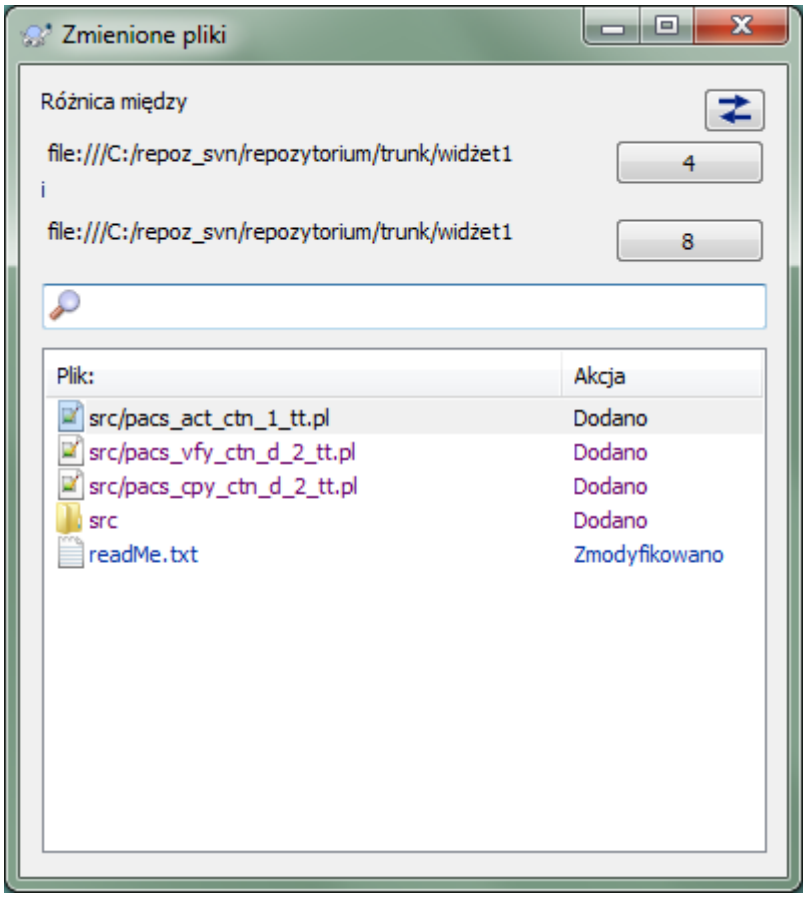
Porównuj białe znaki dołącza wszystkie zmiany wcięcia i białe znaki w ramach linii jako dodawane/usuwane linie.

Ignoruj zmiany białych znaków wyklucza zmiany, które nastąpiły wyłącznie z powodu zmiany ilości lub rodzaju białych znaków, np. zmiany wcięcia lub zmiany tabulacji na spacje. Dodawanie białych znaków, gdzie nie było przedtem żadnego lub usuwanie wszystkich białych znaków jest nadal wyświetlane jako zmiana.

Ignoruj wszystkie zmiany białych znaków wyklucza wszystkie zmiany białych znaków.

Oczywiście, każda linia ze zmienioną treścią jest zawsze wliczana do porównania.

4.11.3. Porównanie folderów



Rysunek 4.29. Okno dialogowe Porównania wersji

Po wybraniu dwóch drzew w przeglądarce repozytorium lub po wybraniu dwóch wersji folderu w oknie dziennika, można wykonać Mmenu kontekstowe → Porównaj wersje.

To okno pokazuje listę wszystkich plików, które się zmieniły i pozwala porównać lub je lub adnotować indywidualnie za pomocą menu kontekstowego.

Można eksportować *drzewo zmian*, co jest przydatne, gdy trzeba wysłać komuś innemu swoją strukturę drzewa projektu, ale dołączając tylko pliki, które uległy zmianie. Operacja ta działa tylko na wybranych plikach, więc musisz wybrać interesujące Was pliki - zwykle oznacza, że wszystkie - a następnie Menu kontekstowe → Eksportuj wybrane elementy do.... Pojawi się monit o lokalizację do zapisania drzewa zmian.

Możecie także wyeksportować *listę* zmienionych plików do pliku tekstowego za pomocą Menu kontekstowego → Zapisz listę wybranych plików do....

Jeśli chcecie wyeksportować listę plików *oraz* akcje (zmodyfikowany, dodany, usunięty), można to zrobić za pomocą Menu kontekstowego → Kopiuj zaznaczenie do schowka.

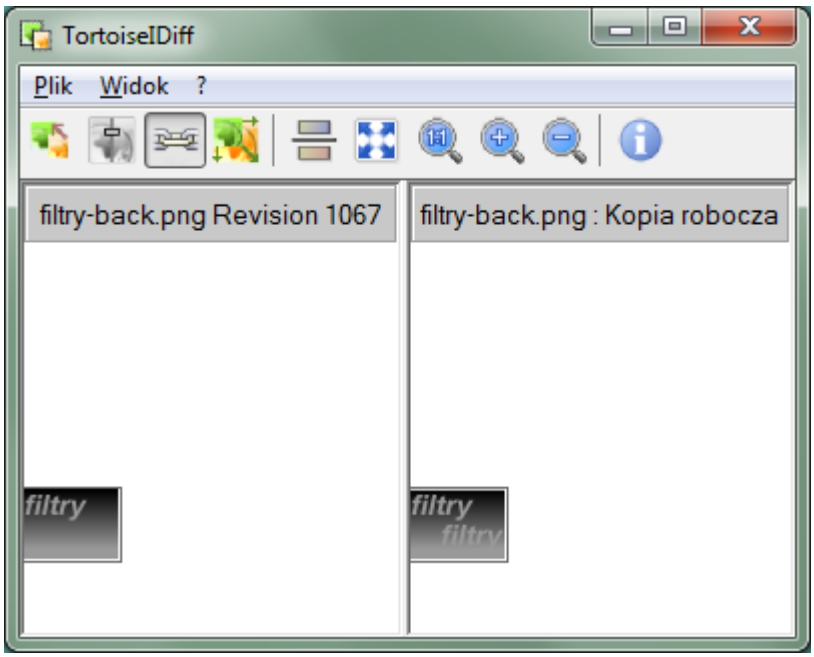
Przycisk u góry pozwala na zmianę kierunku porównania. Można pokazać że zmiany muszą mieć kierunek z punktu A do B, lub jeśli wolicie, od B do A.

Przyciski z numerami wersji można zmienić na inny zakres wersji. Po zmianie zakresu, lista elementów, które różnią się pomiędzy dwiema wersjami zostanie automatycznie zaktualizowana.

Jeśli lista plików jest bardzo długa, można użyć pola wyszukiwania do ograniczenia listy nazw plików zawierających określony tekst. Należy pamiętać, że używane jest proste wyszukiwanie tekstowe, więc jeśli chcesz ograniczyć listę do plików źródłowych w C powinieneś wpisać `.c`, a nie `*.c`.

4.11.4. Porównywanie obrazków przy użyciu TortoiseIDiff

Istnieje wiele dostępnych narzędzi do porównywania plików tekstowych, w tym własne TortoiseMerge, ale często sami chcemy zobaczyć, jak plik zmienił się plik obrazu. Dlatego stworzyliśmy TortoiseIDiff.



Rysunek 4.30. Przeglądarka różnic obrazu

TortoiseSVN → Porównaj dla każdego z popularnych formatów plików obrazu uruchamia TortoiseIDiff by pokazać różnice obrazu. Domyślnie obrazy są wyświetlane obok siebie, ale można skorzystać z menu Widok lub paska narzędzi, aby przejść do widoku góra - dół, lub jeśli wolicie, możecie nałożyć obrazy i udawać, że korzystacie z wyświetlarki Lightbox.

Oczywiście można także powiększać i pomniejszać i przesuwać okno obrazu. Można również przesunąć obraz przeciągając go lewym klawiszem myszki. Jeśli wybierzecie Opcję Powiąż zdjęcia razem, a następnie wykonujecie przesuwanie (paski przewijania, kółko myszy) oba obrazy są połączone.

Pole informacyjne obrazu pokazuje szczegółowe informacje o pliku obrazu, takie jak rozmiar w pikselach, rozdzielczość i głębia kolorów. Jeśli to pole zasłania widok, użyjcie Widok → Informacja o obrazie aby je ukryć. Możecie uzyskać te same informacje w dymku po najechaniu myszką na pasku tytułu obrazu.

Kiedy obrazy są nakładane, względna intensywność obrazów (alpha blend) jest kontrolowana przez suwak po lewej stronie. Możecie kliknąć w dowolnym miejscu na suwak, aby ustawić mieszanie bezpośrednio, lub przeciągać suwak, aby zmienić mieszanie interaktywnie. **Ctrl**+**Shift**-kółko by zmienić mieszanie.

Przycisk powyżej suwaka przełącza pomiędzy 0% i 100% krycia, a jeśli dwukrotnie kliknąć przycisk, mieszanie przełącza automatycznie co sekundę aż do momentu ponownego kliknięcia przycisku. Może to być przydatne przy poszukiwaniu wielu małych zmian.

Czasami chcecie zobaczyć różnicę, a nie mieszankę. Możecie mieć pliki obrazów dla dwóch wersji płytki drukowanej i chcecie sprawdzić, które ścieżki zostały zmienione. Jeśli wyłączycie tryb alfa blend, różnica będzie wyświetlana jako *XOR* z wartości koloru pikseli. Obszary niezmienione będą zupełnie białe, a zmiany będą kolorowe.

4.11.5. Porównywanie Dokumentów Office

Jeśli chcecie porównać nietekstowe dokumenty, normalnie należałoby skorzystać z oprogramowania używanego do tworzenia dokumentu gdyż rozumie ono format pliku. Do najczęściej używanych Microsoft Office i Open

Office mamy jednak pewne wsparcie dla wyświetlania różnic i TortoiseSVN zawiera skrypty wywoływane przy odpowiednich ustawieniach, gdy porównujemy pliki z dobrze znanymi rozszerzeniami. Możecie sprawdzić, które rozszerzenia plików są obsługiwane i dodawać własne, wchodząc w TortoiseSVN → Ustawienia i klikając Zaawansowane w sekcji Program zewnętrzny.

Problemy z Office 2010

Jeśli zainstalowaliście wersję *Click-to-Run* pakietu Office 2010 i próbujecie porównać dokumenty może pojawić się komunikat o błędzie systemu Windows Script Host w stylu: „Składnik ActiveX nie może utworzyć obiektu : Word.Application”. Wydaje się, musicie użyć opartej na MSI wersji pakietu Office, aby uzyskać funkcjonalność porównywania.

4.11.6. Zewnętrzne narzędzia porównywania/scalania

Jeśli dostarczone przez nas narzędzia nie robią tego, czego potrzebujecie, spróbujcie jednego z wielu dostępnych programów open-source lub komercyjnych. Każdy ma swoje ulubione i ta lista wcale nie jest kompletna, ale zawiera kilka, które warto rozważyć:

WinMerge
: *WinMerge* [https://winmerge.sourceforge.net/] jest doskonałym opensursowym narzędziem porównywania, które obsługuje także katalogi.

Perforce Merge
: Perforce jest komercyjnym RCS, ale można pobrać narzędzie porównywania/scalenia za darmo. Więcej informacji można uzyskać z *Perforce* [https://www.perforce.com/perforce/products/merge.html].

KDiff3
: KDiff3 jest darmowym narzędziem porównywania, które obsługuje także foldery. Można go pobrać *stąd* [http://kdiff3.sf.net/].

SourceGear DiffMerge
: SourceGear Vault jest komercyjnym RCS, ale można pobrać narzędzie porównywania/scalenia za darmo. Więcej informacji można uzyskać z *SourceGear* [https://www.sourcegear.com/diffmerge/].

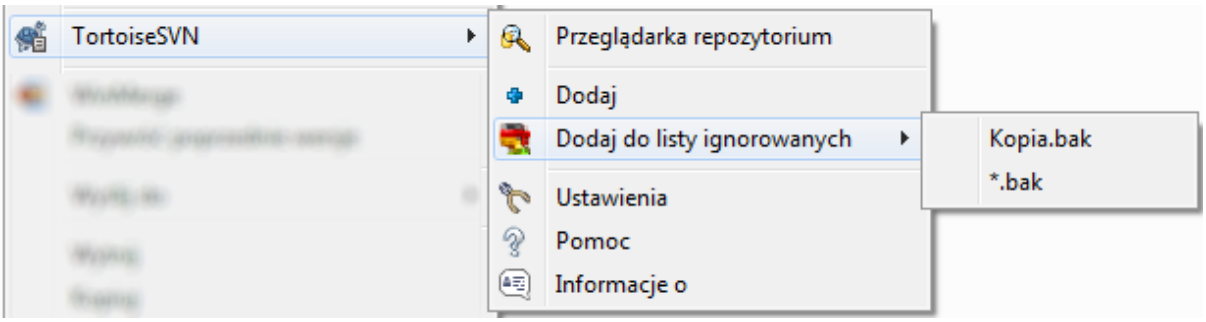
ExamDiff
: ExamDiff Standard jest programem bezpłatnym. Obsługuje on pliki, ale nie foldery. ExamDiff Pro jest shareware i dodaje kilka bajerów w tym porównywanie katalogu i możliwość edycji. W obu smakach, wersja 3.2 i powyżej mogą obsługiwać unicode. Można je pobrać z *PrestoSoft* [http://www.prestosoft.com/].

Beyond Compare
: Podobnie do ExamDiff Pro jest to wspaniałe narzędzie shareware do porównywania obsługujące zarówno foldery jak i unikod. Do pobrania ze *Scooter Software* [https://www.scootersoftware.com/].

Araxis Merge
: Araxis Merge jest użytecznym komercyjnym narzędziem do porównywania i scalania zarówno plików i folderów. Wykonuje ono trójdrożnego porównania podczas scalania i ma wiązania synchronizacji do użycia jeśli zmienicie kolejność funkcji. Pobrać go można z *Araxis* [https://www.araxis.com/merge/index.html].

Czytaj Sekcja 4.31.5, „Ustawienia programów zewnętrznych” by uzyskać informacje jak skonfigurować TortoiseSVN by używać tych narzędzi.

4.12. Dodawanie nowych plików i folderów



Rysunek 4.31. Menu kontekstowe eksploratora dla niewersjonowanych plików

Jeśli utworzyliście nowe pliki i/lub katalogi w trakcie procesu rozwoju, musicie dodać je również do kontroli wersji. Wybierzcie plik(i) i/lub folder(y) i skorzystajcie z TortoiseSVN → Dodaj.

Po dodaniu plików/katalogów do kontroli wersji, plik pojawia się z nakładką ikony `dodane` co oznacza, że teraz trzeba zatwierdzić kopię roboczą, aby te pliki/katalogi stały się dostępne dla innych programistów. Dodanie pliku/katalogu *nie* wpływa na repozytorium!

Wiele dodań

Można również użyć polecenia Dodaj na już wersjonowanych folderach. W tym przypadku okno dodawania pokaże wszystkie pliki bez informacji o wersji wewnątrz tego wersjonowanego folderu. Jest to pomocne, jeśli macie wiele nowych plików i trzeba dodać je wszystkie naraz.

Aby dodać pliki spoza kopii roboczej można użyć funkcji obsługi przeciągnij i upuść:

1. wybierzcie pliki, które chcecie dodać
2. przeciągnijcie prawym przyciskiem do nowej lokalizacji wewnątrz kopii roboczej
3. puśćcie prawy przycisk myszy
4. wybierzcie Menu kontekstowe → SVN kopiuj i dodaj pliki do tej kopii roboczej. Pliki zostaną skopiowane do kopii roboczej i dodane do kontroli wersji.

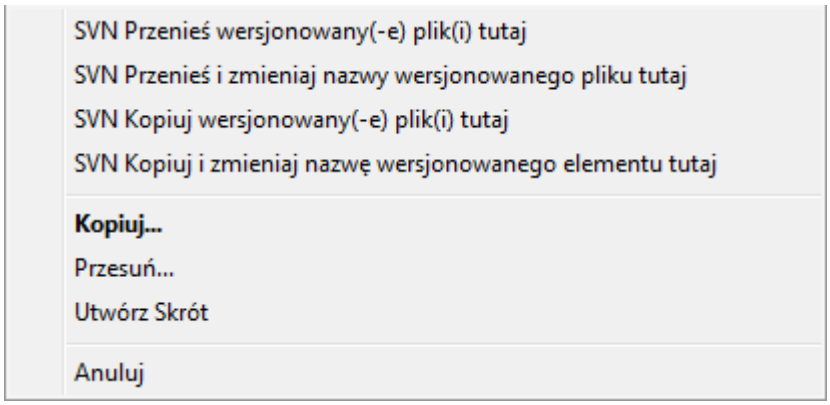
Możecie również dodać pliki z kopii roboczej po prostu klikając i przeciągając je lewym klawiszem na okno dialogowe zatwierdzenia.

Jeśli dodaliście plik lub folder przez pomyłkę, można cofnąć dodanie, zanim je zatwierdzicie za pomocą TortoiseSVN → Cofnij dodanie....

4.13. Kopiowanie/przenoszenie/zmiana nazwy plików i folderów

Często zdarza się, że macie już potrzebne pliki w innym projekcie w repozytorium i chcecie po prostu skopiować stamtąd. Możecie po prostu skopiować pliki i dodać je w sposób opisany powyżej, ale ten sposób nie dołączy żadnej historii. A jeśli później naprawicie błąd w oryginalnych plikach, można scalić korektę automatycznie tylko wtedy, gdy nowa kopia jest powiązana z oryginałem w Subversion.

Najprostszym sposobem kopiowania plików i folderów w kopii roboczej jest wykorzystanie prawego menu przeciągania. Kiedy przeciąga się prawym przyciskiem plik lub folder w jednej kopii roboczej do innego, lub nawet tego samego folderu, pojawi się menu kontekstowe po zwolnieniu przycisku myszy.



Rysunek 4.32. Menu prawo-przeciągnięcia dla katalogu pod kontrolą wersji

Teraz możecie skopiować istniejącą wersjonowaną zawartość do nowej lokalizacji, być może zmieniając nazwę w tym samym czasie.

Możecie również skopiować lub przenieść wersjonowane pliki wewnątrz kopii roboczej, lub między dwiema kopiami roboczymi, za pomocą znanej metody wytnij i wklej. Wykorzystajcie windowsowy standard Kopiuj lub Wytnij by skopiować jeden lub więcej wersjonowanych elementów do schowka. Jeśli schowek zawiera takie wersjonowane elementy, można użyć TortoiseSVN → Wklej (uwaga: nie standardowego windowsowego Wklej) do skopiowania lub przeniesienia tych elementów na nowe miejsce w kopii roboczej.

Można kopiować pliki i foldery z kopii roboczej do innej lokalizacji w repozytorium przez TortoiseSVN → Gałąź/etykieta. Zapoznajcie się z Sekcja 4.20.1, „Tworzenie gałęzi lub etykiety” by dowiedzieć się więcej.

Można znaleźć starszą wersję pliku lub folderu w oknie dziennika i skopiować ją do nowej lokalizacji w repozytorium bezpośrednio z okna dziennika przy użyciu Menu kontekstowego → Utwórz gałąź/etykietę z wersji. Zapoznajcie się z Sekcja 4.10.3, „Ustalenie dodatkowych informacji” by dowiedzieć się więcej.

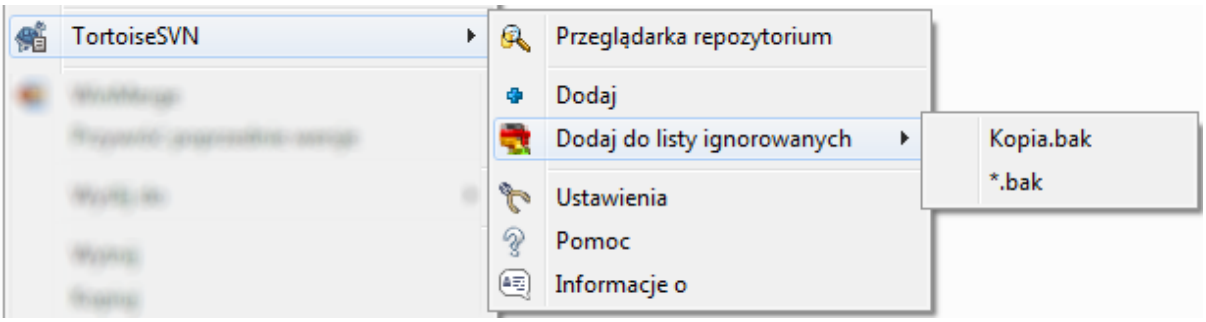
Można również użyć przeglądarki repozytorium, aby zlokalizować dane, które chcecie i skopiować je do swojej kopii roboczej bezpośrednio z repozytorium, albo skopiować pomiędzy dwiema lokalizacjami w repozytorium. Zapoznajcie się z Sekcja 4.25, „Przeglądarka repozytorium” aby dowiedzieć się więcej.

Nie można kopiować pomiędzy repozytoriami

Chociaż można kopiować lub przenosić pliki i foldery *wewnątrz* repozytorium, *nie* można kopiować i przenosić z jednego repozytorium do drugiego z zachowaniem historii za pomocą TortoiseSVN. Nawet jeśli repozytoria leżą na tym samym serwerze. Wszystko, co możecie zrobić, to skopiować zawartość w jej obecnym stanie i dodać ją jako nową zawartość do drugiego repozytorium.

Jeśli nie jesteście pewni czy dwa adresy URL na tym samym serwerze wskazują na to samo czy dwa różne repozytoria, należy przy użyciu przeglądarki repozytorium otworzyć jeden z adresów URL i dowiedzieć się, gdzie znajduje się folder główny repozytorium. Jeśli widzicie obie lokalizacje w jednym oknie przeglądarki to są one w tym samym repozytorium.

4.14. Ignorowanie plików i folderów



Rysunek 4.33. Menu kontekstowe eksploratora dla niewersjonowanych plików

W większości projektów występować będą pliki i foldery, które nie powinny podlegać kontroli wersji. Mogą to być pliki utworzone przez kompilator, `*.obj`, `*.lst`, albo folder wyjściowy używany do przechowywania pliku wykonywalnego. Kiedykolwiek zatwierdzacie zmiany, TortoiseSVN pokazuje niewersjonowane pliki, które zapełniają listę plików w oknie dialogowym zatwierdzenia. Oczywiście można wyłączyć ich wyświetlanie, ale wtedy można zapomnieć o dodaniu nowego pliku źródłowego.

Najlepszym sposobem na uniknięcie tych problemów jest dodanie plików pochodnych do listy ignorowanych projektu. W ten sposób nigdy nie pojawi się w oknie dialogowym zatwierdzenia, a niewersjonowane oryginalne pliki źródłowe nadal będą oznakowane.

Wykonując prawoklik na pojedynczym niewersjonowanym pliku i wybierając polecenie TortoiseSVN → Dodaj do Listy Ignorowanych z menu kontekstowego, wyświetlimy menu podrzędne pozwalające na wybór tylko tego pliku lub wszystkich o tym samym rozszerzeniu. Oba podmenu mają także równoważnik `(rekurencyjnie)`. Jeśli wybierze się wiele plików, nie ma podmenu i można dodać tylko takie pliki/foldery.

Przy wybraniu wersji `(rekurencyjnie)` menu kontekstowego ignorowania, element zostanie pominięty nie tylko w wybranym folderze ale również w folderach podrzędnych. Wymaga to jednak wersji klienta SVN 1.8 lub wyższej.

Jeśli chcecie usunąć jeden lub więcej elementów z listy ignorowanych, trzeba kliknąć prawym przyciskiem myszy na te elementy, a następnie wybrać TortoiseSVN → Usuń z listy ignorowanych Możecie również uzyskać dostęp do atrybutu folderu `svn:ignore` bezpośrednio. To pozwala określić bardziej ogólne wzorce przy użyciu masek pliku, opisanych w rozdziale poniżej. Czytajcie Sekcja 4.18, „Ustawienia projektu” by uzyskać więcej informacji na temat ustawiania właściwości bezpośrednio. Należy pamiętać, że każdy wzorzec ignorować musi być umieszczony w osobnej linii. Oddzielanie ich spacjami nie działa.

Globalna lista ignorowania

Innym sposobem na ignorowanie plików, jest dodanie ich do *globalnej listy ignorowania*. Największą różnicą jest to, że globalne listy ignorowania są własnością klienta. Dotyczy to *wszystkich* projektów Subversion, ale tylko na kliencie PC. Ogólnie lepiej jest używać w miarę możliwości atrybutu `svn:ignore`, ponieważ może być stosowany do określonych obszarów projektu, i działa dla każdego, kto pobierze projekt. Czytajcie Sekcja 4.31.1, „Ustawienia ogólne” dla uzyskania dokładniejszych informacji.

Ignorowanie elementów wersjonowanych

Wersjonowane pliki i foldery nie mogą być ignorowane - to cecha Subversion. Jeśli włączyliście plik do kontroli wersji przez pomyłkę, przeczytajcie Sekcja B.8, „Ignorowanie plików, które już są pod kontrolą wersji” by uzyskać instrukcję „odwersjonowania” go.

4.14.1. Dopasowanie wzorców w listach ignorowania

Wzorce ignorowania Subversion wykorzystują z obsługę masek pliku, technikę pierwotnie używaną w Unixie by określić pliki za pomocą wieloznacznych meta-znaków. Następujące znaki mają specjalne znaczenie:

*
: Pasuje do dowolnego ciągu znaków, w tym ciągu pustego (bez znaków).

?
: Zastępuje dowolny pojedynczy znak.

[...]
: Pasuje do każdego ze znaków zamkniętych w nawiasach kwadratowych. Wewnątrz nawiasów, para znaków oddzielonych „-” pasuje do dowolnego znaku leksykalnego pomiędzy nimi. Na przykład `[AGM-p]` pasuje do każdego pojedynczego znaku z `A`, `G`, `m`, `n`, `o` lub `p`.

Dopasowanie do wzorca jest wrażliwe na wielkość liter, co może powodować problemy w systemie Windows. Możecie wymusić niewrażliwości na krótko przez parowanie znaków, np. by zignorować `*. tmp`, niezależnie od przypadku, należy użyć wzoru jak `*.[Tt][Mm][Pp]`.

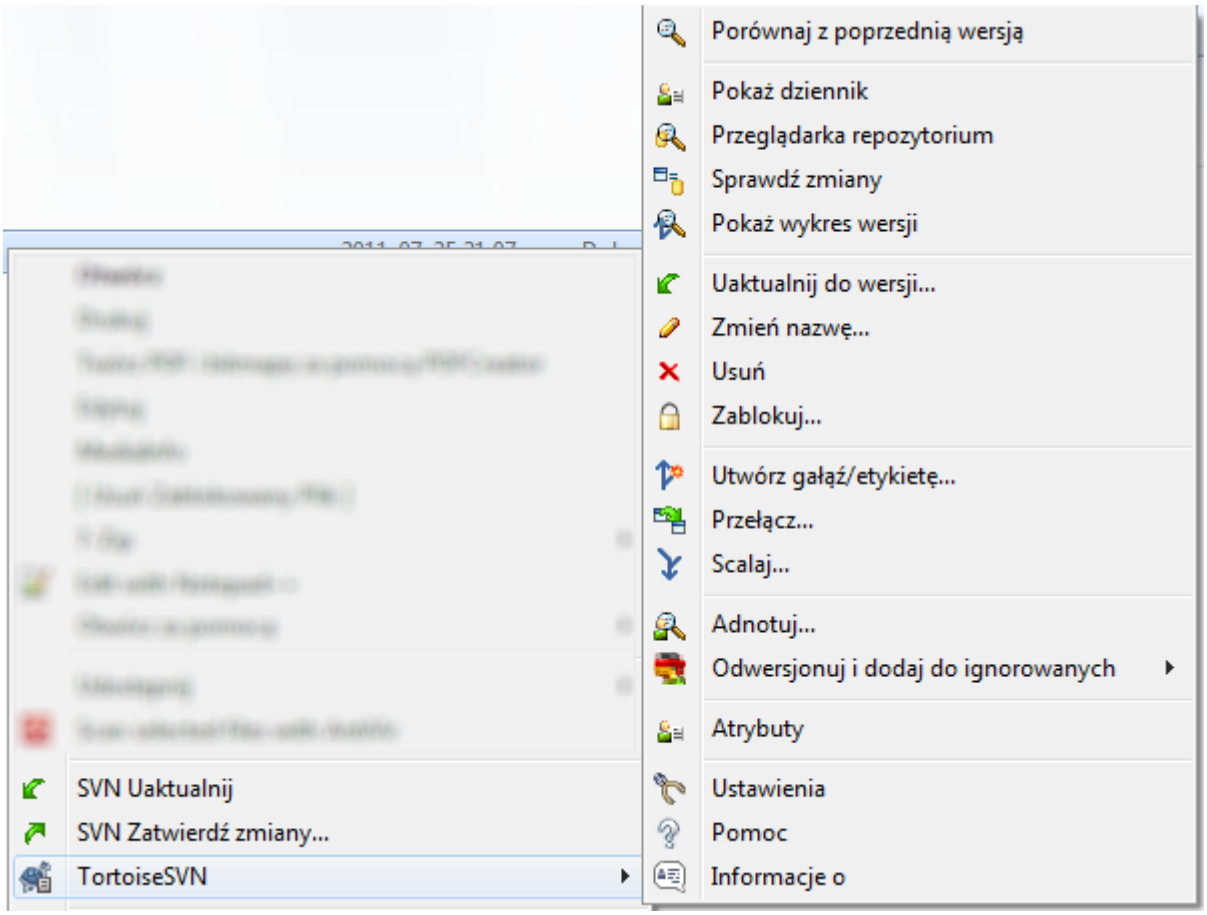
Jeśli potrzebujecie oficjalnej definicji dla obsługi masek, można go znaleźć w specyfikacji IEEE dla języka powłoki poleceń *Pattern Matching Notation* [http://www.opengroup.org/onlinepubs/009695399/utilities/xcu_chap02.html#tag_02_13].

Brak ścieżek w globalnej liście ignorowanych

Nie powinno się zawierać informacji o ścieżce we wzorcu. Wzorzec dopasowania ma być użyty na pełnych nazwach plików i folderów. Jeśli chcecie, aby ignorować wszystkie foldery `CVS`, po prostu dodaj `CVS` do listy ignorowanych. Nie ma potrzeby, aby określić `CVS */CVS`, tak jak w poprzednich wersjach. Jeśli chcecie, aby ignorować wszystkie foldery `tmp`, gdy znajdują się w folderach `prog`, ale nie w folderach `doc` należy użyć atrybutu `svn:ignore`. Nie ma niezawodnego sposobu osiągnięcia tego celu za pomocą globalnego wzorca ignorowania.

4.15. Usuwanie, przenoszenie i zmiana nazwy

Subversion umożliwia zmianę nazwy i przenoszenie plików i folderów. Dlatego są wpisy menu pozwalające usuwać i zmieniać nazwy w podmenu TortoiseSVN.



Rysunek 4.34. Menu kontekstowe eksploratora dla wersjonowanych plików

4.15.1. Usuwanie plików i katalogów

Użyj TortoiseSVN → Usuń by usunąć pliki i foldery z Subversion.

Podczas wykonywania TortoiseSVN → Usuń na pliku lub folderze, jest on usuwany natychmiastowo z kopii roboczej oraz zaznaczony do usunięcia w repozytorium podczas następnego zatwierdzenia. Folder nadrzędny elementu wyświetla nakładkę ikony „zmieniono”. Do czasu zatwierdzenia zmiany, można odzyskać plik używając TortoiseSVN → Wycofaj na folderze nadrzędnym.

Jeśli chcecie usunąć element z repozytorium, ale zachować go lokalnie jako niewersjonowany plik/folder, użyjcie Rozszerzonego menu kontekstowego → Usuń (zachowaj lokalnie). Musicie przytrzymać klawisz **Shift**, podczas gdy prawy przycisk myszy wciskacie na elemencie w panelu listy plików w eksploratorze (prawy panel), aby zobaczyć to rozszerzone menu kontekstowe.

Jeśli element został usunięty w eksploratorze zamiast przy użyciu menu kontekstowego TortoiseSVN, okno zatwierdzenia wyświetla te elementy jako brakujące i pozwala wam usunąć je równiez z kontroli wersji przed zatwierdzeniem. Jednakże podczas uaktualnienia kopii roboczej, Subversion zauważy brak elementu i wypełni go ostatnią wersją z repozytorium. Jeśli potrzebujesz usunąć plik z kontroli wersji, zawsze korzystajcie z TortoiseSVN → Usuń by Subversion nie musiało zgadywać, co macie zamiar zrobić.

Odzyskiwanie usuniętego pliku lub folderu

Jeśli usunięto plik lub folder i już zatwierdzono tą operację usuwania do repozytorium, zwykłe TortoiseSVN → Wycofaj zmiany nie może go już przywrócić. Jednak plik lub folder nie jest

zupełnie stracony. Jeśli znacie wersję, w której plik lub folder zostały usunięte (jeśli nie, skorzystajcie z okna dziennika, aby się dowiedzieć) otwórzcie przeglądarkę repozytorium i przełączcie się do tej wersji. Następnie wybierzcie ten usunięty plik lub folder, kliknijcie prawym przyciskiem myszy i wybierzcie Menu kontekstowe → Kopiuj do ... jako cel tej operacji kopiowania wybierzcie ścieżkę do kopii roboczej.

4.15.2. Przenoszenie plików i katalogów

Jeśli chcecie zrobić prostą miejscową zmianę nazwy z pliku lub folderu, użyjcie Menu kontekstowe → Zmień nazwę... Wpiszcie nową nazwę elementu i gotowe.

Jeśli chcecie przenieść pliki wewnątrz kopii roboczej, być może do innego podkatalogu, należy użyć obsługi przeciągnięcia prawym przyciskiem myszy:

1. wybierzcie pliki lub katalogi, które chcecie przenieść
2. przeciągnijcie prawym przyciskiem do nowej lokalizacji wewnątrz kopii roboczej
3. puśćcie prawy przycisk myszy
4. w menu kontekstowym wybierzcie Menu kontekstowe → SVN Przenieś wersjonowany(-e) plik(i) tutaj

Zatwierdź folder nadrzędny

Ponieważ zmiana nazwy i przesunięcie są wykonywane jako usunięcie, a następnie dodanie trzeba zatwierdzić folder nadrzędny do przemianowanego/przeniesionego pliku, aby usuwana część zmiany nazwy/przeniesienia pojawiła się w oknie dialogowym zatwierdzenia. Jeśli nie zatwierdzi się usuwana część przemianowania/przeniesienia, to pozostanie w repozytorium, a kiedy współpracownicy zaktualizują kopię roboczą, stary plik nie zostanie usunięty. tj. będą posiadać *obie*, zarówno starą i nową kopię.

Jest *konieczne* zatwierdzenie zmiany nazwy folderu przed zmianą jakiegokolwiek plików w tym folderze, w przeciwnym razie w kopii roboczej zrobi się prawdziwy rozgardiasz.

Innym sposobem przenoszenia lub kopiowania plików jest użycie poleceń Windows kopiuj/wytnij. Wybierzcie pliki, które chcecie skopiować, kliknijcie prawym przyciskiem myszy i wybierzcie Menu kontekstowe → Kopiuj z menu kontekstowego eksploratora Windows. Następnie przejdźcie do folderu docelowego, kliknijcie prawym przyciskiem myszy i wybierzcie TortoiseSVN → Wklej. Przy przenoszeniu plików, wybierzcie Menu kontekstowe → Wytnij zamiast Menu kontekstowe → Kopiuj.

Można również użyć przeglądarki repozytorium, aby przenosić elementy wokół. Czytajcie Sekcja 4.25, „Przeglądarka repozytorium”, aby dowiedzieć się więcej.

Nie SVN przenoście zewnętrznych

Nie możecie używać poleceń TortoiseSVN Move lub Zmień nazwę w folderze, który został stworzony przy wykorzystaniu `svn:externals`. Ta akcja spowoduje, że zewnętrzny element zostanie usunięty ze swojego repozytorium, prawdopodobnie denerwując wielu innych ludzi. Jeśli chcecie przenieść folder zewnętrzny należy użyć zwykłych poleceń przenoszących powłoki, a następnie dostosować atrybuty `svn:externals` folderów nadrzędnych źródła i przeznaczenia.

4.15.3. Radzenie sobie z konfliktami wielkości liter nazwy pliku

Jeśli repozytorium zawiera już dwa pliki o tej samej nazwie, ale różniące się tylko w wielkością liter (np. `TEST.TXT` i `test.txt`), nie będziecie mogli zaktualizować ani pobrać katalogu nadrzędnego na kliencie Windows. Chociaż Subversion obsługuje różnice w wielkości liter w nazwach plików, system Windows nie.

To się czasem zdarza, kiedy dwaj ludzie zatwierdzają z oddzielnych kopii roboczych pliki, które czasem mają taką samą nazwę, ale z różnicą wielkości liter. Może się również zdarzyć, gdy pliki są zatwierdzone z systemu obsługującego rozróżnianie wielkości liter, takich jak Linux.

W takim przypadku trzeba zdecydować, który z nich ma pozostać a drugi usunąć (lub zmienić nazwę) z repozytorium.

Zapobieganie podwójnym plikom o tej samej nazwie

Istnieje skrypt przechwytujący serwera dostępny pod adresem: https://svn.apache.org/repos/asf/subversion/trunk/contrib/hook-scripts/, który zapobiega pobraniom, w wyniku których powstaną konflikty wielkości liter.

4.15.4. Naprawa zmian zewnętrznych

Czasami przyjazne IDE będzie zmieniać nazwy plików za Was w ramach wykonywania refaktoryzacji, i oczywiście nie poinformuje Subversion. Jeśli spróbujecie zatwierdzić swoje zmiany, Subversion będzie widzieć stare pliki jako brakujące, a nowe jako niewersjonowane. Możecie po prostu zaznaczyć nową nazwę pliku, do dodania, ale stracicie wtedy ślad historyczny, jako że Subversion nie wie, że pliki są powiązane.

Lepszym sposobem jest powiadomienie Subversion, że ta zmiana jest rzeczywiście przemianowaniem, a można to zrobić w oknach Zatwierdź zmiany i Sprawdź zmiany. Wystarczy wybrać jednocześnie starą nazwę (brak) i nową nazwę (bez informacji o wersji) i użyć Menu kontekstowe → Napraw przeniesienie by powiązać dwa pliki jako zmianę nazwy.

4.15.5. Usunięcie niewersjonowanych plików

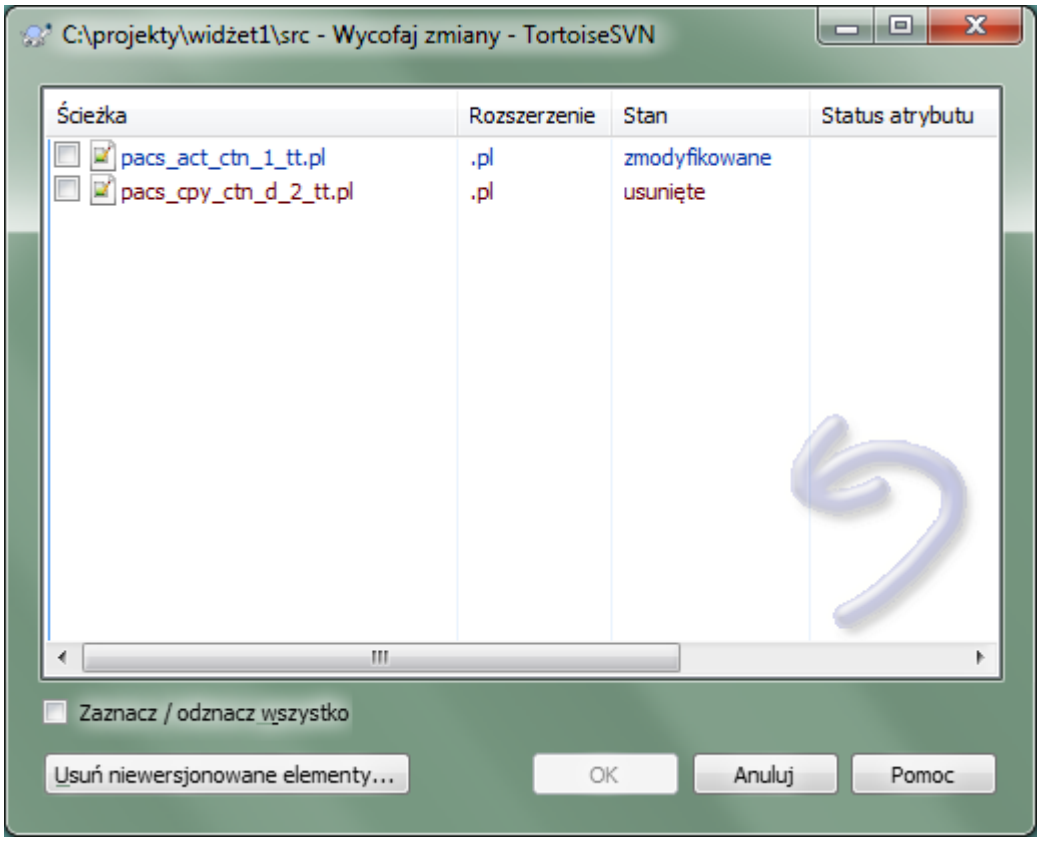
Zazwyczaj ustawia się listy ignorowanych tak, że wszystkie wygenerowane pliki są ignorowane w Subversion. Ale co, jeśli chcecie wyczyścić wszystkie te ignorowane elementy w celu wykonania czystej kompilacji? Zazwyczaj ustawia się to w pliku makefile, ale jeśli debugujecie makefile, lub zmieniacie system kompilacji warto mieć sposób oczyszczania pokładów.

TortoiseSVN zapewnia właśnie taką opcję za pomocą Rozszerzonego menu kontekstowego → Usuń niewersjonowane elementy.... Musicie przytrzymać **Shift** podczas kliknięcia prawym przyciskiem myszy na folder w okienku eksploratora listy (prawy panel), aby zobaczyć rozszerzone menu kontekstowe. Wciśnięcie wejścia menu wyświetli okno dialogowe, które listuje wszystkie pliki bez informacji o wersji w dowolnym miejscu kopii roboczej. Możecie wybrać lub odznaczyć elementy do usunięcia.

Gdy takie elementy są usuwane, wykorzystuje się kosz, więc jeśli się tu pomylicie i usuniecie plik, który powinien być wersjonowany, nadal można go odzyskać.

4.16. Cofnij zmiany

Jeśli chcecie cofnąć wszystkie zmiany dokonane w pliku od ostatniej aktualizacji należy wybrać plik, kliknąć prawym przyciskiem myszy by wyświetlić menu kontekstowe, a następnie wybrać polecenie TortoiseSVN → Wycofaj zmiany pojawi się okno dialogowe pokazujące pliki, które zostały zmienione i mogą być przywrócone. Wybierzcie te, które chcecie przywrócić i kliknijcie OK.



Rysunek 4.35. Okno dialogowe wycofania zmian

Jeśli chcecie wyczyścić wszystkie ustawione listy zmian, zaznacz pole wyboru u dołu okna.

Jeśli chcecie cofnąć usunięcie lub zmianę nazwy, należy użyć Wycofaj zmiany na folderze nadrzędnym jako że element usunięty nie istnieje by móc na nim kliknąć prawym przyciskiem myszy.

Jeśli chcecie cofnąć dodanie elementu, to pojawia się w menu kontekstowym opcja TortoiseSVN → Cofnij dodanie.... To jest naprawdę również wycofaniem zmian, ale nazwa została zmieniona aby uczynić je bardziej oczywistym.

Kolumny w tym oknie można dostosować w ten sam sposób, jak kolumny w dialogu Sprawdź zmiany. Czytajcie Sekcja 4.7.3, „Status lokalny i zdalny” dla dalszych szczegółów.

Ponieważ wycofanie zmian jest czasem używane do czyszczenia kopii roboczej, pojawia się dodatkowy przycisk, który pozwala usunąć również niewersjonowane elementy. Po kliknięciu na ten przycisk pojawia się kolejne okno z listą wszystkich elementów bez informacji o wersji, które można następnie wybrać do usunięcia.

Cofnięcie zmian już zatwierdzonych

Wycofaj zmiany cofa tylko zmiany lokalne. *Nie* cofa wszelkich zmian, które zostały już zatwierdzone. Jeżeli chcecie cofnąć wszystkie zmiany, które zostały popełnione w określonej wersji, przeczytajcie Sekcja 4.10, „Okno dialogowe dziennika wersji” dla uzyskania dalszych informacji.

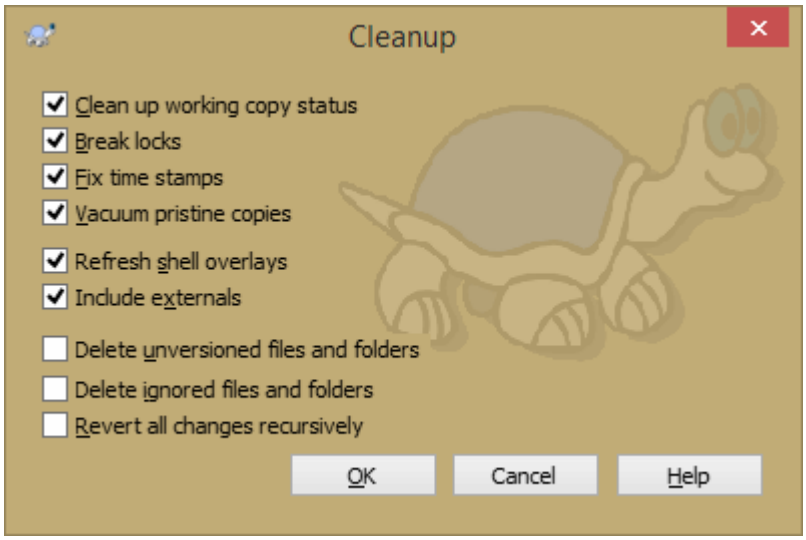
Wycofanie zmian jest powolne

Gdy wycofujecie zmiany może się okazać, że operacja trwa o wiele dłużej niż się spodziewacie. To dlatego, że zmodyfikowana wersja pliku jest wysyłana do kosza, więc można odzyskać zmiany jeśli wycofujecie je przez pomyłkę. Jeśli jednak Kosz jest pełny, Windows zajmuje dużo czasu, aby

znaleźć miejsce do umieszczenia pliku. Rozwiązanie jest proste: albo opróżnić Kosz, albo wyłączyć pole wyboru Użyj kosza przy wycofywaniu zmian w oknie ustawień TortoiseSVN.

4.17. Uporządkuj

Jeżeli polecenie Subversion nie może zakończyć się pomyślnie, być może z powodu problemów z serwerem, kopia robocza może zostać pozostawiona w niespójnym stanie. W takim przypadku należy użyć TortoiseSVN → Uporządkuj na folderze. Jest dobrym pomysłem, aby wykonać je na najwyższym poziomie w kopii roboczej.



Rysunek 4.36. Okno Uporządkuj

W oknie dialogowym porządkowania istnieją także inne przydatne opcje, aby przywrócić kopię roboczą do `czystego` stanu.

Status uporządkowania kopii roboczej
: Jak wspomniano powyżej, opcja ta stara się doprowadzić niespójną kopię roboczą do działającego i używalnego stanu. Nie ma to wpływu na jakiekolwiek dane a jedynie wewnętrzne stany bazy danych kopii roboczej. To jest rzeczywiście polecenie `Uporządkuj` znane ze starszych klientów TortoiseSVN lub innych klientów SVN.

Złam blokady zapisu
: Gdy zaznaczone, wszyskie blokady zapisu są usuwane z bazy danych kopii roboczej. W większości sytuacji jest to wymagane by czyszczenie zadziałało!

  Wyłączajcie tę opcję tylko gdy kopia robocza jest używana w tym czasie przez innych użytkowników/klienty. Jednak jeśli czyszczenie się nie powiedzie, musicie zaznaczyć tę opcję by czyszczenie się udało.

Napraw znaczniki czasu
: Zmienia wszystkie znaczniki czasowe plików na czas ostatniego zatwierdzenia.

Uporządkuj pierwotne kopie
: Usuwa nieużywane kopie pierwotne i kompresuje wszystkie pozostałe kopie pierwotne plików kopii roboczej.

Odśwież nakładki powłoki
: Czasem nakładki powłoki, szczególnie w widoku drzewa po lewej stronie eksploratora nie pokazują aktualnego stanu, lub bufor statusu nie uwzględnił zmian. W tej sytuacji można użyć do wymuszenia odświeżenia.

Dołącz zewnętrzne
: Jeśli jest ono zaznaczone, to wszystkie działania są wykonywane również dla wszystkich plików i folderów z atrybutem `svn:externals`.

Usuń niewersjonowane pliki i foldery, usuwanie zignorowanych plików i folderów
: Jest to szybki i łatwy sposób na usunięcie wszystkich wygenerowanych plików w kopii roboczej. Wszystkie pliki i foldery, które nie są wersjonowane są przenoszone do kosza.

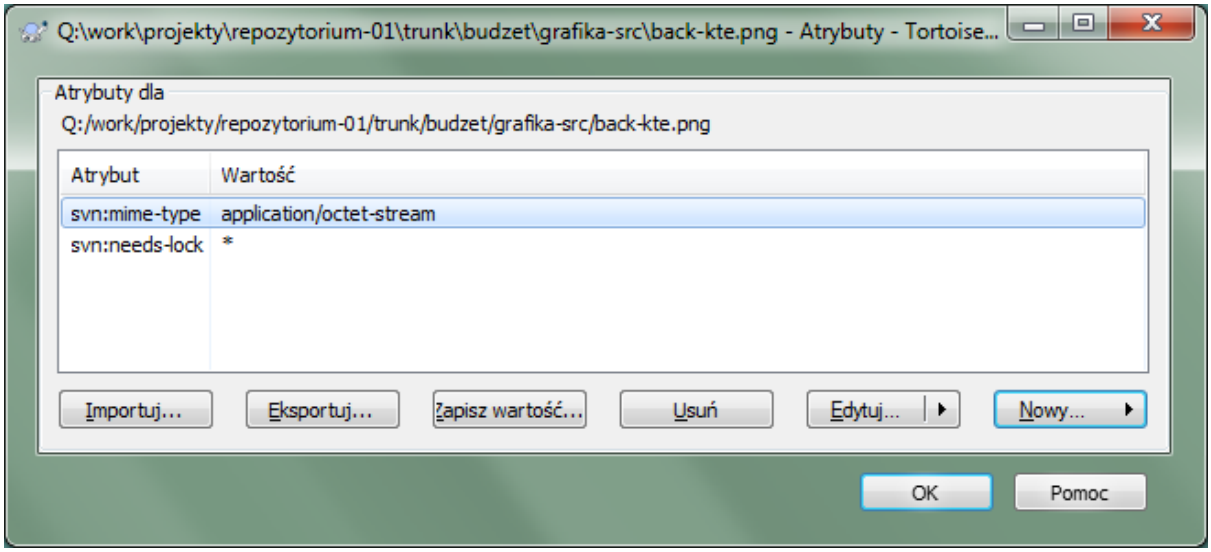
  Uwaga: można również zrobić to samo w oknie TortoiseSVN → Wycofaj zmiany. Można tu również uzyskać listę wszystkich plików i folderów bez informacji o wersji i wybrać do usunięcia.

Przywróć wszystkie zmiany rekurencyjnie
: To polecenie przywraca wszystkie lokalne modyfikacje, które nie są jeszcze zatwierdzone.

  Uwaga: lepiej użyć polecenia TortoiseSVN → Wycofaj zmiany, bo można najpierw zobaczyć i wybrać pliki, które chcesz przywrócić.

4.18. Ustawienia projektu

4.18.1. Atrybuty Subversion



Rysunek 4.37. Strona atrybutu Subversion

Możecie czytać i ustawiać atrybuty Subversion w oknie właściwości systemu Windows, ale także w TortoiseSVN → Atrybuty i na listach stanu TortoiseSVN z menu kontekstowego → Właściwości.

Możecie dodać własne atrybuty, lub pewne atrybuty o specjalnym znaczeniu w Subversion. Zaczynają się one od `svn:`. `svn:externals` jest takim atrybutem; zobaczcie jak obsługiwać zewnętrzne w Sekcja 4.19, „Elementy zewnętrzne”.

4.18.1.1. svn:keywords

Subversion obsługuje CVSopodobne rozwinięcia słów kluczowych, które można wykorzystać do umieszczenia nazwy pliku pliku i informacji o wersji w samym pliku. Obsługiwane są obecnie słowa kluczowe:

$Date$
: Data ostatniego znanego zatwierdzenia. Pochodzi ona z informacji uzyskanych podczas uaktualniania kopii roboczej. *Nie* jest sprawdzane repozytorium, aby znaleźć późniejsze zmiany.

$Revision$
: Wersja ostatniego znanego zatwierdzenia.

$Author$
: Autor wykonujący ostatnie znane zatwierdzenie.

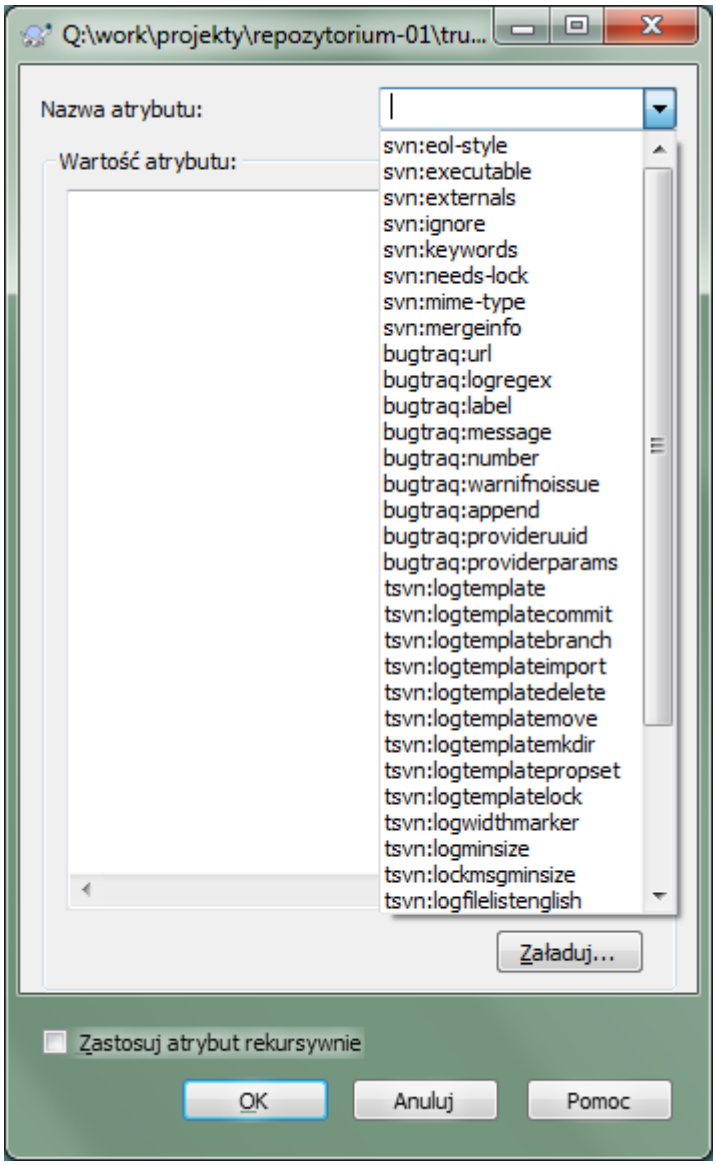
$HeadURL$
Pełny adres URL tego pliku w repozytorium.

Id
Skompresowane połączenie czterech poprzednich słów kluczowych.

Aby dowiedzieć się, jak używać tych słów kluczowych, spójrzcie na *svn:keywords section* [http://svnbook.red-bean.com/en/1.8/svn.advanced.props.special.keywords.html] w księdze Subversion, która daje pełny opis tych słów kluczowych oraz jak je włączyć i używać.

By dowiedzieć się więcej o atrybutach w Subversion przejdźcie do *Atrybuty Specjalne* [http://svnbook.red-bean.com/en/1.8/svn.advanced.props.html].

4.18.1.2. Dodawanie i edytowanie atrybutów



Rysunek 4.38. Dodawanie atrybutów

Aby dodać nowy atrybut, kliknijcie na Nowy.... Wybierzcie odpowiednią nazwę atrybutu z menu, a następnie uzupełnijcie wymagane informacje w określonym oknie dialogowym atrybutu. Te specyficzne okna dialogowe atrybutów są opisane bardziej szczegółowo w Sekcja 4.18.3, „Edytory atrybutów”.

Aby dodać atrybut, który nie ma własnego okna dialogowego, wybierzcie Zaawansowane z menu Nowy.... Następnie wybierzcie istniejący atrybut z listy rozwijanej lub wprowadźcie własną nazwę atrybutu.

Jeśli chcecie zastosować atrybut do wielu elementów naraz, wybierzcie pliki/foldery w eksploratorze, a następnie wybierzcie menu kontekstowe → atrybuty.

Jeśli chcecie zastosować właściwość do *każdego* pliku i folderu w hierarchii poniżej bieżącego folderu, zaznaczcie pole wyboru Rekursywne.

Jeśli chcecie edytować istniejący atrybut, zaznaczcie atrybut na liście istniejących atrybutów, a następnie kliknijcie Edytuj....

Jeśli chcecie usunąć istniejący atrybut, zaznaczcie go na liście istniejących atrybutów, a następnie kliknijcie Usuń.

Atrybut `svn:externals` może być wykorzystany do podciągnięcia elementów w innych projektach z tego samego lub innego repozytorium. Aby uzyskać więcej informacji, przeczytajcie Sekcja 4.19, „Elementy zewnętrzne”.

Edycja atrybutów w wersji HEAD

Ponieważ atrybuty są wersjonowane, nie można edytować atrybutów poprzednich wersji. Jeśli spojrzeć na atrybuty w oknie dialogowym dziennika, lub z wersji nie-HEAD w przeglądarce repozytorium, pojawi się lista właściwości i wartości, ale nie pola edycji.

4.18.1.3. Eksport i import atrybutów

Często okazuje się, że stosuje się ten sam zestaw właściwości wielokrotnie, na przykład `bugtraq:logregex`. Aby uprościć proces kopiowania właściwości z jednego projektu do drugiego, można użyć funkcji Eksport/Import.

Z pliku lub folderu, gdzie atrybuty są już ustawione, użyjcie TortoiseSVN → atrybuty, wybierzcie atrybuty, które chcecie wyeksportować i kliknijcie Eksport.... Zostaniecie poproszeni o nazwę pliku, w którym nazwy atrybutów i wartości zostaną zapisane.

Na folderze(ach), gdzie chcecie zastosować te atrybuty, należy użyć TortoiseSVN → atrybuty i kliknąć Importuj.... Zostaniecie poproszeni o nazwę pliku źródłowego, więc przejdźcie do miejsca zapisania pliku wcześniej wyeksportowanego i wybierzcie go. Atrybuty zostaną dodane do folderów nierekurencyjnie.

Jeśli chcecie dodać właściwości do drzewa rekurencyjnie, wykonajcie czynności opisane powyżej, a następnie w oknie atrybutów wybierzcie każdy atrybut po kolei, kliknijcie na Edytuj..., zaznaczcie pole wyboru Zastosuj atrybut rekursywnie i kliknijcie na OK.

Format pliku importu jest binarny i przeznaczony jedynie dla TortoiseSVN. Jego jedynym celem jest przeniesienie atrybutów za pomocą importu i eksportu, nie ma zatem potrzeby, aby edytować te pliki.

4.18.1.4. Atrybuty binarne

TortoiseSVN może obsługiwać binarne wartości atrybutów przy użyciu plików. Aby odczytać binarne wartości atrybutów, Zapisz... do pliku. Aby ustawić wartość binarną, należy użyć edytora szesnastkowego lub innego właściwego narzędzia do tworzenia pliku którego treści wymagasz, a następnie Załaduj... z tego pliku.

Chociaż atrybuty binarne nie są często stosowane, mogą być przydatne w niektórych zastosowaniach. Na przykład do przechowywania ogromnych plików graficznych lub jeśli aplikacja służąca do załadowania pliku jest ogromna, więc może chcecie przechowywać miniaturę jako właściwość, dzięki czemu można uzyskać szybko podgląd.

4.18.1.5. Automatyczne ustawienie atrybutu

Możecie skonfigurować Subversion i TortoiseSVN by ustawiać automatycznie atrybuty plików i folderów, gdy są one dodawane do repozytorium. Istnieją dwa sposoby na zrobienie tego.

Możecie edytować plik konfiguracyjny Subversion by włączyć tą funkcję klienta. Strona Ogólne w oknie ustawień TortoiseSVN ma przycisk prowadzący tam bezpośrednio. Plik konfiguracyjny to prosty plik tekstowy kontrolujący pewne działania Subversion. Powinniście zmienić dwie rzeczy: po pierwsze w sekcji z nagłówkiem `miscellany` usuńcie komentarz z linii `enable-auto-props = yes`. po drugie powinniście poprawić sekcjęponiżejby

określić, które właściwości chcecie dodać do poszczególnych typów plików. Ta metoda to standardowa funkcja Subversion i działa na każdym kliencie Subversion. Musi być jednak zdefiniowana na każdym kliencie osobno - nie ma sposobu by rozpropagować te ustawienia z repozytorium.

Alternatywną metodą jest ustawienie atrybutu `tsvn:autoprops` na folderach, jak opisano w następnej sekcji. Ta metoda działa tylko dla klientów TortoiseSVN, ale rozsyłany jest do wszystkich kopii roboczych podczas aktualizacji.

W Subversion 1.8 można również ustawić atrybut `svn:auto-props` na folderze głównym. Wartość atrybutu jest automatycznie dziedziczona przez wszystkie elementy podrzędne.

Niezależnie od wybranej metody, należy pamiętać, że auto-props są stosowane tylko do plików w czasie, gdy te są dodawane do kopii roboczej. Auto-props nigdy nie zmieni właściwości plików, które są już wersjonowane.

Jeśli chcecie mieć pewność, że nowe pliki mają ustawione odpowiednie atrybuty, należy skonfigurować w repozytorium przechwycenie pre-commit do odrzucenia zatwierdzeń gdzie wymagane atrybuty nie są ustawione.

Atrybuty zatwierdzenia

Atrybuty Subversion są wersjonowane. Po zmianie lub dodaniu atrybutu musicie zatwierdzić swoje zmiany.

Konflikty atrybutów

Jeśli występuje konflikt w czasie zatwierdzenia zmian, ponieważ inny użytkownik zmienił ten sam atrybut, Subversion generuje plik `.prej`. Należy usunąć ten plik po rozwiązaniu konfliktu.

4.18.2. Atrybuty projektu TortoiseSVN

TortoiseSVN ma kilka specjalnych atrybutów, a ich nazwy zaczynają się od `tsvn:`.

- `tsvn:logminsize` ustawia minimalną długość opisu zmiany podczas zatwierdzenia. Jeśli wprowadzicie krótszy opis niż podano tutaj, zatwierdzenie zostaje zablokowane. Ta funkcja jest bardzo przydatna dla przypomnienia o wprowadzeniu właściwego opisowego komunikatu dla każdego zatwierdzenia. Jeśli ten atrybut nie jest ustawiony, lub jego wartość wynosi zero, puste opisy zmian są dozwolone.

  `tsvn:lockmsgminsize` ustawia minimalną długość komunikatu blokady. Jeśli wprowadzicie krótszy opis niż podano tutaj, blokada zostaje zablokowana. Ta funkcja jest bardzo przydatna dla przypomnienia o wprowadzeniu właściwego opisowego komunikatu dla każdej blokady. Jeśli ten atrybut nie jest ustawiony, lub jego wartość wynosi zero, puste komunikaty blokady są dozwolone.

- `tsvn:logwidthmarker` jest używany w projektach, które wymagają opisów zmian sformatowanych z pewną maksymalną szerokością (zwykle 80 znaków) przed znakiem nowej linii. Ustawienie tej właściwości na różną od zera będzie robić 2 rzeczy w oknie dialogowym dziennika: umieszcza znacznik, aby wskazać maksymalną szerokość oraz wyłącza zawijanie podczas wyświetlania, dzięki czemu można sprawdzić, czy wpisany tekst nie jest zbyt długi. Uwaga: funkcja ta będzie działała poprawnie tylko jeżeli używacie czcionki o stałej szerokości do wprowadzania opisów zmian.

- `tsvn:logtemplate` jest używany w projektach, w których stosuje się zasady formatowania wiadomości dziennika. Atrybut zawiera wieloliniowy tekst, który zostanie wstawiony w pole opisu zmiany wyświetlane podczas uruchamiania zatwierdzenia. Następnie można edytować go uzupełniając wymagane informacje. Uwaga: jeśli używacie także `tsvn:logminsize`, należy ustawić długość większą niż szablon lub stracicie mechanizm ochrony.

  Istnieją także szablony związane z akcjami, które można używać zamiast `tsvn:logtemplate`. Szablony związane z akcjami są używane jeśli zostaną ustawione, zaś `tsvn:logtemplate` będzie używany gdy żaden szablon związany z akcją nie został ustawiony.

  Lista szablonów związanych z akcjami:

- `tsvn:logtemplatecommit` jest używany dla wszystkich zatwierdzeń z kopii roboczej.
- `tsvn:logtemplatebranch` jest używany podczas tworzenia gałęzi/etykiety, lub gdy kopiujesz plik lub folder bezpośrednio w przeglądarce repozytorium.
- `tsvn:logtemplateimport` jest używany do importów.
- `tsvn:logtemplatedelete` jest używany gdy usuwa się element bezpośrednio w przeglądarce repozytorium.
- `tsvn:logtemplatemove` jest używany podczas zmiany nazwy lub przenoszenia elementów w przeglądarce repozytorium.
- `tsvn:logtemplatemkdir` jest używany podczas tworzenia folderów w przeglądarce repozytorium.
- `tsvn:logtemplatepropset` jest używany podczas modyfikowania atrybutów w przeglądarce repozytorium.
- `tsvn:logtemplatelock` jest używany gdy wprowadzamy blokadę.

- Subversion allows you to set „autoprops” which will be applied to newly added or imported files, based on the file extension. This depends on every client having set appropriate autoprops in their Subversion configuration file. `tsvn:autoprops` can be set on folders and these will be merged with the user's local autoprops when importing or adding files. The format is the same as for Subversion autoprops, e.g. `*.sh = svn:eol-style=native;svn:executable` sets two properties on files with the `.sh` extension.

  Jeśli istnieje konflikt między lokalnymi autoatrybutami i `tsvn:autoprops`, ustawienia projektu mają pierwszeństwo, ponieważ są one specyficzne dla tego projektu.

  W Subversion 1.8, powinno się użyć atrybutu `svn:auto-props` zamiast `tsvn:autoprops` ponieważ ma taką samą funkcjonalność, jednak działa ze wszystkimi klientami svn a nie jest specyficzny dla TortoiseSVN.

- W oknie zatwierdzenia macie możliwość, aby wkleić listę zmienionych plików, w tym stan każdego pliku (dodane, zmienione, itp.). `tsvn:logfilelistenglish` określa, czy status jest wprowadzony w języku angielskim lub w wersji zlokalizowanej. Jeśli właściwość nie jest ustawiona, domyślnie jest to `true`.

- TortoiseSVN może wykonywać sprawdzanie pisowni. Na Windows 10 używane jest sprawdzanie pisowni Systemu Operacyjnego. W starszych wersjch Windows wykorzystywane są moduły sprawdzania pisowni, które są również używane przez OpenOffice i Mozilla. Jeśli macie je zainstalowane, ten atrybut określa, które sprawdzanie pisowni jest używane, czyli w jakim języku powinny być pisane opisy zmian dla projektu. `tsvn:projectlanguage` ustawia język w module silnika sprawdzenia pisowni, którego należy używać podczas wprowadzania opisu zmiany. Możecie znaleźć wartości dla danego języka na stronie: *MSDN: Language Identifiers* [http://msdn2.microsoft.com/en-us/library/ms776260.aspx].

  Możecie wprowadzić tę wartość w postaci dziesiętnej, lub w systemie szesnastkowym poprzedzone ciągiem `0x`. Na przykład angielski (US) można wprowadzić jako `0x0409` lub `1033`.

- Atrybut `tsvn:logsummary` jest używany by wyodrębnić część opisu zmiany, która jest następnie wyświetlana w oknie dziennika jako podsumowanie opisu zmiany.

  Wartość atrybutu `tsvn:logsummary` musi być ustawiona jako jednoliniowy łańcuch regeks, który zawiera jedną grupę regeks. Cokolwiek pasuje do wyników tej grupy jest używane jako podsumowanie.

  Przykład: `\[SUMMARY\]:\s+(.*)` Będzie obejmować wszystko po „[SUMMARY]” w opisie zmiany i używa tego jako podsumowania.

- Atrybut `tsvn:logrevregex` określa wyrażenie regularne, które pasuje do odniesień do zmian w opisie zmiany. Jest ono używane w oknie dziennika by zmienić takie odniesienia na linki, które po kliknięciu będzie albo przewijać do wskazanej wersji (jeśli wersja jest już wyświetlana w oknie logowania, lub jeśli jest ona dostępna w buforze dziennika) lub otworzyć nowe okno dziennika pokazujące tą wersję.

Wyrażenie regularne musi pasować do całego odniesienia, nie tylko numeru wersji. Numer wersji jest wyodrębniany automatycznie z pasującego ciągu odniesienia.

Jeśli ten atrybut nie jest ustawiony, używane jest domyślne wyrażenie regularne do połączenia odniesień do wersji.

- Istnieje kilka atrybutów służących do konfiguracji skryptów przechwytujących po stronie klienta. Każdy atrybut dotyczy jednego specyficznego rodzaju skryptu.

  Dostępne atrybuty/skrypty przechwytujące to

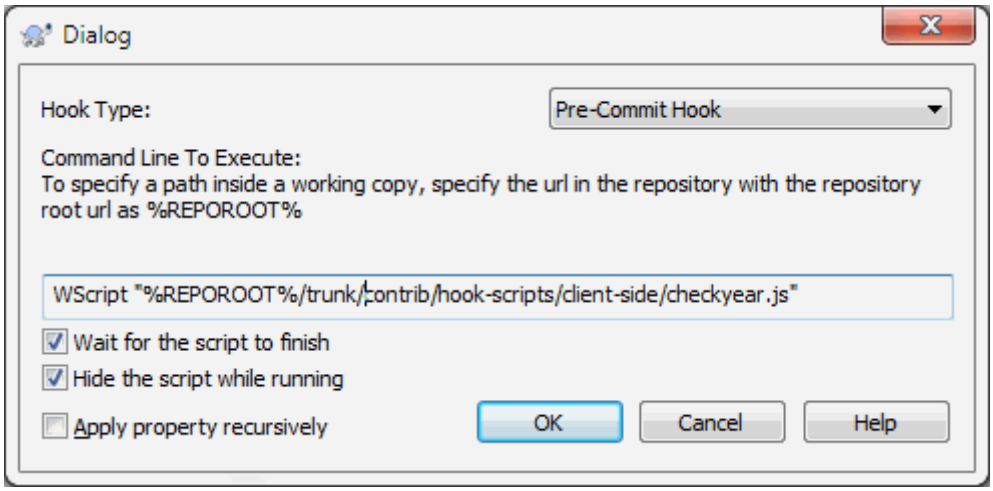
  - tsvn:startcommithook
  - tsvn:precommithook
  - tsvn:postcommithook
  - tsvn:startupdatehook
  - tsvn:preupdatehook
  - tsvn:postupdatehook
  - tsvn:prelockhook
  - tsvn:postlockhook

  Parametry są takie same jak przy skonfigurowaniu skryptu przechwytującego w oknie ustawień. Zobacz Sekcja 4.31.8, „Skrypty przechwytujące po stronie klienta” by poznać szczegóły.

  Jako że nie każdy użytkownik pobiera do takiego samgo położenia , możecie skonfigurować skrypt/narzędzie, które wykonuje się na kopii roboczej wskazując URL w repozytorium, używając do tego `%REPOROOT%`jako część adresu URL do kattlaogu głównego repozytorium. Na przykład, jeśli wasz skrypt przechwytujący znajduje się w kopii roboczej pod nazwą `contrib/hook-scripts/client-side/checkyear.js`, możesz wskazać ścieżkę do niego jako `%REPOROOT%/trunk/contrib/hook-scripts/client-side/checkyear.js`. W ten sposób nawet jeśli przeniesiecie repozytorium na inny serwer, nie musicie dostosowywać atrybutów skryptu.

  Zamiast `%REPOROOT%` możecie również wskazać `%REPOROOT+%`. Znak + jest używany by dodać dowolnie zagłębioną ścieżkę folderów konieczną do wyszukania skryptu. Jest to przydatne gdy chcemy wskazać ten skrypt tak, by po ewentualnym utworzeniu gałęzi sktypt został również znaleziony pomimo że url kopii roboczej jest inny. Korzystając z przykładu powyżej, należałoby wskazać ścieżkę do skryptu `%REPOROOT+%/contrib/hook-scripts/client-side/checkyear.js`.

  Poniższy zrzut ekranu pokazuje, jak w TortoiseSVN jest skonfigurowany skrypt wykonujący sprawdzenie bieżący rok praw autorskich w nagłówkach pliku źródłowego.

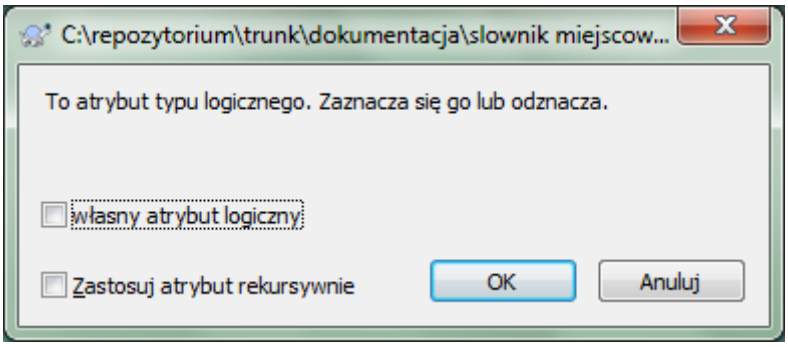


Rysunek 4.39. Okno dialogowe atrybutów skryptów przechwytujących

- Jeśli chcecie dodać nowy atrybut, można wybrać jeden z listy menu rozwijalnego lub wpisać dowolną nazwę atrybutu jaką chcecie. Jeśli projekt korzysta z niektórych niestandardowych atrybutów i chcecie, żeby te atrybuty pojawiły się na liście w menu rozwijalnym (aby uniknąć literówek przy wpisywaniu nazwy atrybutu), można utworzyć listę niestandardowych atrybutów za pomocą `tsvn:userfileproperties` i `tsvn:userdirproperties`. Zastosujcie te atrybuty w folderze. Gdy przechodzicie do edycji właściwości każdego elementu podrzędnego, niestandardowe atrybuty pojawią się na liście predefiniowanych nazw atrybutów.

  Można również określić, czy niestandardowe okno dialogowe służy do dodawania/edycji atrybutu. TortoiseSVN oferuje cztery różne okna dialogowe, w zależności od rodzaju atrybutu.

  bool
  : Jeśli wasz atrybut może przyjmować tylko dwa stany, np.. prawda i fałsz, możecie ustawić wasz atrybut jako typu `bool`.



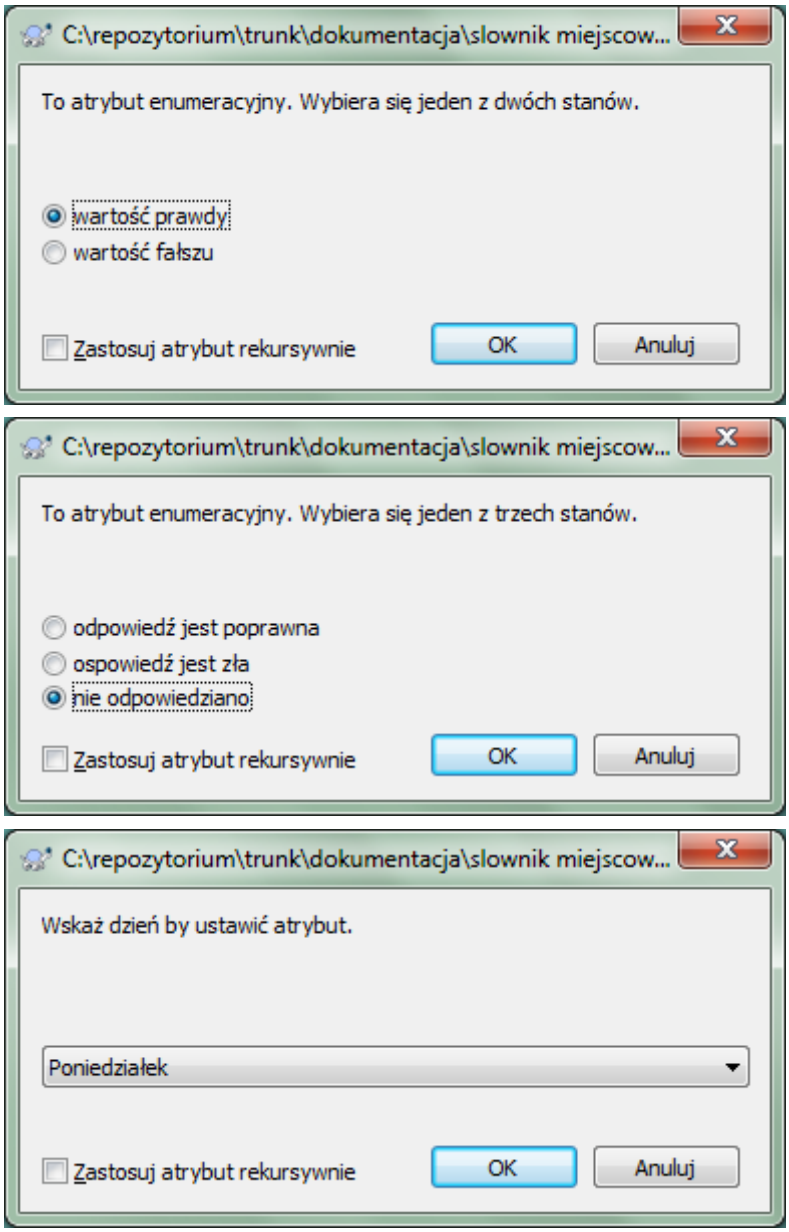
Rysunek 4.40. Dialog atrybutu typu logicznego zdefiniowanego przez użytkownika

  Zdefiniuj atrybut w poniższy sposób:

  ```
  nazwaatrybutu=bool;tekstEtykiety(WARTOSCTAK;WARTOSCNIE;TekstPolaWyboru)
  ```

  `tekstEtykiety` to tekst pokazywany w dialogu nad polem wyboru gdzie możesz wyjaśnić cel i sposób użycia atrybutu. Inne parametry powinny wyjaśniać się same.

  state
  : Jeśli atrybut może przyjmować jeden z wielu możliwych stanów, np. `tak`, `nie`, `może`, możecie zdefiniować atrybut jako typu `state`



Rysunek 4.41. Dialog atrybutu typu enumeracyjnego zdefiniowanego przez użytkownika

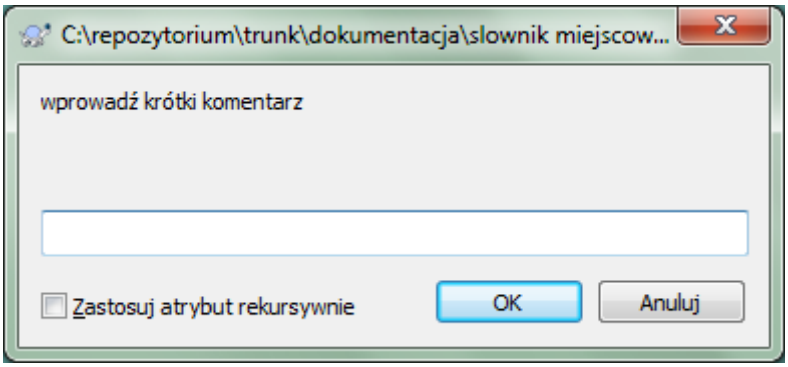
w ten sposób:

```
nazwaatrybutu=state;tekstEtykiety(WARTDOM;WART1;TEKST1;WART2;TEKST2;WART3;TEKST3;..
```

Parametry są takie same jak w atrybutach `bool`, wraz z `WARTDOM` określającą wartość domyślną wykorzystywaną, gdy atrybut nie jest ustawiony lub ma wartość spoza dozwolonego zbioru.

Dla trzech lub mniej wartości, dialog wyświetla odpowiednią liczbę przycisków opcji. Jeśli zdefiniowano więcej wartości, wykorzystuje się listę rozwijalną, z której użytkownik może wybrać wymagany stan.

singleline
: Dla atrybutów tekstowych mieszczących się w jednym wierszu, użyjcie typu atrybutu `singleline`:

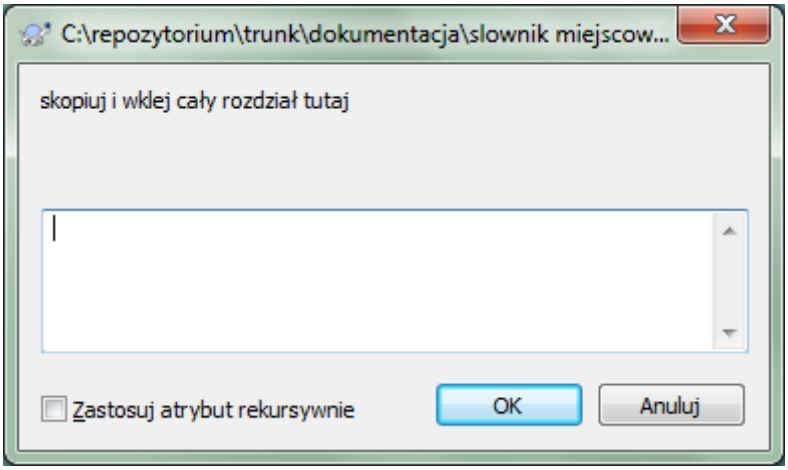


Rysunek 4.42. Okno atrybutów jednoliniowych typów użytkownika

`nazwaatrybutu=singleline;tekstetykiety(wyrreg)`

`wyrreg` określa wyrażenie regularne używane do sprawdzenia poprawności (dopasowanie) tekstu wprowadzonego przez użytkownika. Jeśli tekst nie pasuje do wyrreg, wyświetlony zostanie komunikat o błędzie a atrybut nie zostaje ustawiony.

multiline
Dla atrybutów tekstowych mieszczących się w wielu wierszach, użyjcie typu atrybutu `multiline`:



Rysunek 4.43. Okno atrybutów wieloliniowych typów użytkownika

`nazwaatrybutu=multiline;tekstetykiety(wyrreg)`

`wyrreg` określa wyrażenie regularne używane do sprawdzenia poprawności (dopasowanie) tekstu wprowadzonego przez użytkownika. Nie zapomnijcie o wpisaniu znaku nowej linii (\n) w wyrażeniu regularnym!

Zrzuty ekranu powyżej zostały wykonane dla następujących `tsvn:userdirproperties`

```
my:boolprop=bool; To atrybut typu logicznego. Zaznacza się go lub odznacza. (true;fals
my:stateprop1=state;To atrybut enumeracyjny. Wybiera się jeden z dwóch stanów. (true;t
my:stateprop2=state;To atrybut enumeracyjny. Wybiera się jeden z trzech stanów.(maybe;
my:stateprop3=state;Wskaż dzień by ustawić atrybut.(1;1;Poniedziałek;2;Wtorek;3;Środa;
my:singlelineprop=singleline;wprowadź krótki komentarz(.*)
my:multilineprop=multiline;skopiuj i wklej tutaj cały rozdział(.*)
```

TortoiseSVN można zintegrować z niektórymi narzędziami śledzenia błędów. Używa się do tego celu atrybutów projektu, które zaczynają się od `bugtraq:`. Czytajcie Sekcja 4.29, „Integracja z systemami śledzenia błędów / śledzenia problemów” dla dokładniejszych informacji.

Można go również zintegrować z niektórymi internetowymi przeglądarkami repozytoriów za pomocą atrybutów projektu, które zaczynają się od `webviewer:`. Przeczytajcie Sekcja 4.30, „Integracja z internetowymi przeglądarkami repozytoriów” dla uzyskania dalszych informacji.

Ustawienie właściwości projektu dla folderów

Te szczególne właściwości projektu muszą być ustawione na *folderach* aby system działał. Podczas korzystania z polecenia TortoiseSVN, które wykorzystuje te atrybuty, są one odczytywane z katalogu, na który kliknięto. Jeżeli atrybuty nie występują tam, TortoiseSVN będzie przeszukiwać poprzez drzewo folderów, aby je znaleźć, aż napotka na niewersjonowany folder lub korzeń drzewa (np. `C:\`). Jeśli możecie być pewni, że każdy użytkownik pobiera tylko np. z `trunk/` a nie z jakiegoś podfolderu, to wystarczy ustawić atrybuty na `trunk/`. Jeśli nie możemy być pewni, należy ustawić atrybuty rekurencyjnie dla każdego podkatalogu. Jeśli ustawiacie ten same atrybuty, ale używacie różnych wartości na różnych głębokościach w hierarchii projektu to będziecie mieli różne wyniki w zależności od miejsca gdzie klikniecie w strukturze folderów.

Tylko do atrybutów projektu, czyli `tsvn:`, `bugtraq:` i `webviewer:` można użyć pola wyboru Rekursywne, aby ustawić atrybut do wszystkich podkatalogów w hierarchii, bez jednoczesnego ustawienia na wszystkich plikach.

Przy dodawaniu nowych podfolderów do kopii roboczej za pomocą TortoiseSVN, wszelkie atrybuty projektu obecne w folderze nadrzędnym będą również automatycznie dodawane do nowego folderu podrzędnego.

Ograniczenia użycia przeglądarki repozytorium

Wczytywanie atrybutów zdalnie jest powolną operacją, więc niektóre funkcje opisane powyżej nie będą działać w przeglądarce repozytorium tak, jak w kopii roboczej.

- Kiedy dodajecie obiekt za pomocą przeglądarki repo, tylko standardowe atrybuty `svn:` oferowane są na liście predefiniowanych. Wszelkie inne nazwy właściwości muszą zostać wprowadzone ręcznie.
- Atrybutów nie można ustawić ani usunąć rekurencyjnie za pomocą przeglądarki repo.
- Atrybuty projektu *nie* będą propagowane automatycznie, gdy folder podrzędny jest dodawany za pomocą przeglądarki repo.
- `tsvn:autoprops` *nie* ustawi atrybutów na plikach, które są dodawane za pomocą przeglądarki repo.

Ostrzeżenie

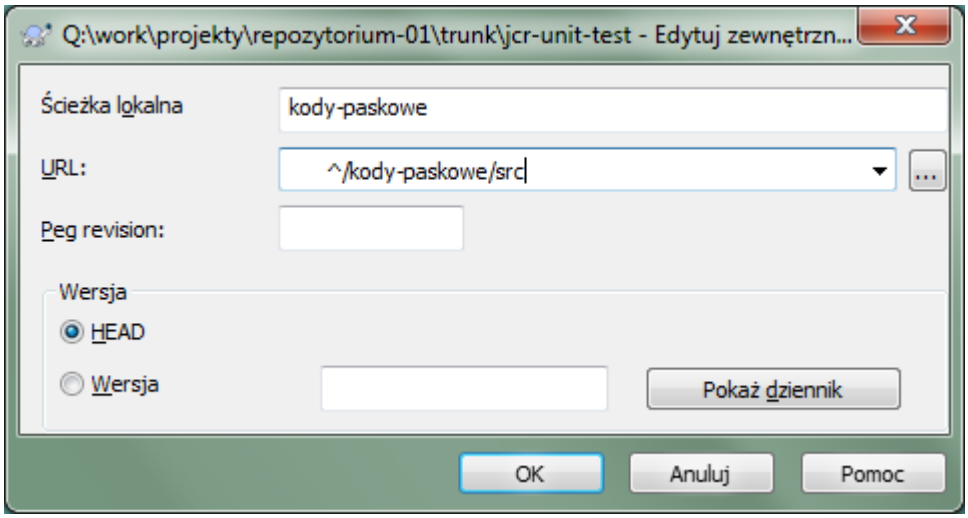
Chociaż atrybuty projektu TortoiseSVN są bardzo przydatne, jednak działają tylko z TortoiseSVN, a niektóre będą działać tylko w nowszych wersjach TortoiseSVN. Jeśli ludzie pracujący nad projektem wykorzystują różne klienty Subversion, ewentualnie mają stare wersje TortoiseSVN, możecie użyć przechwyceń repozytorium by egzekwować politykę projektu. atrybuty projektu mogą przyczynić się do realizacji polityki, nie mogą jej egzekwować.

4.18.3. Edytory atrybutów

Niektóre atrybuty muszą używać zbioru określonych wartości lub być sformatowane w szczególny sposób, aby można je było wykorzystać do automatyzacji. Do uzyskania poprawnego formatowania TortoiseSVN pokazuje

okna dialogowe edycji szczególnych atrybutów wyświetlające dozwolone wartości, lub dzieli atrybut na jego poszczególne składniki.

4.18.3.1. Zawartość zewnętrzna



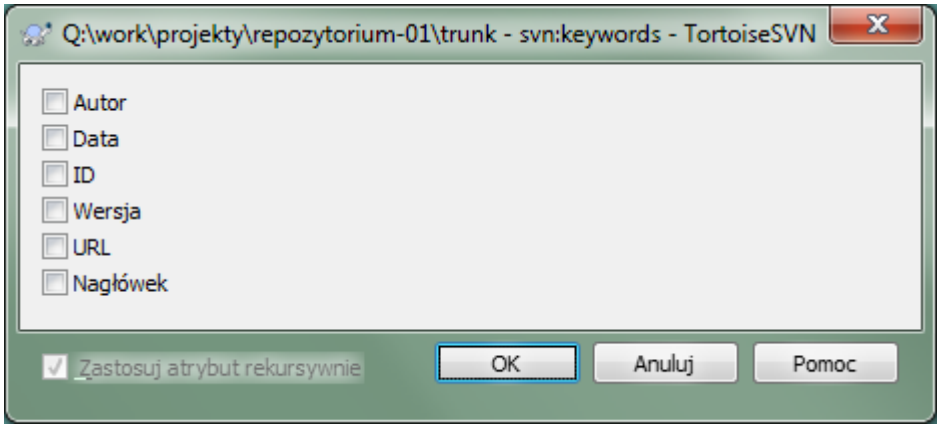
Rysunek 4.44. Strona atrybutu svn:externals

Atrybut `svn:externals` może być stosowany by wyciągnąć w innych projektach zawartość z tego samego lub zupełnie innego repozytorium sposób opisany w Sekcja 4.19, „Elementy zewnętrzne”.

Należy zdefiniować nazwę podfolderu, jako który zostanie pobrany folder sernętrzny, oraz adres URL elementu zewnętrznego w Subversion. Można pobrać zewnętrzny w swojej wersji HEAD, co spowoduje, że gdy element zewnętrzny zmieni się w repozytorium, wasza kopia robocza odczyta te zmiany podczas aktualizacji. Jednakże, jeśli trzeba by zewnętrzny wskazywał na szczególny stały punkt, możecie wskazać by użyto określonej wersji. W tym przypadku można również wskazać tą samą wersję jako wersję wieszakową. Jeśli element zewnętrzny zmieni nazwę w przyszłości, Subversion nie będzie w stanie zaktualizować tego elementu w waszej kopii roboczej. Przez wskazanie wersji wieszakowej wskazujecie Subversion by odnaleźć element mający tą samą nazwę wersji wieszakowej zamiast HEAD.

Przycisk Znajdź wersję HEAD pobiera wersję HEAD z każdego URL zewnętrznego i pokazuje tą wersję HEAD w prawej kolumnie. Popoznaniu wersji HEAD, prosty prawoklik na zewnętrznym daje wyświetla polecenie do zawieszenia wybranych zewnętrznych na ich wybranej wersji HEAD. W przypadku, gdy wersja HEAD nie jest jeszcze znana, polecenie spod prawokliku pobierze najpierw wersję HEAD.

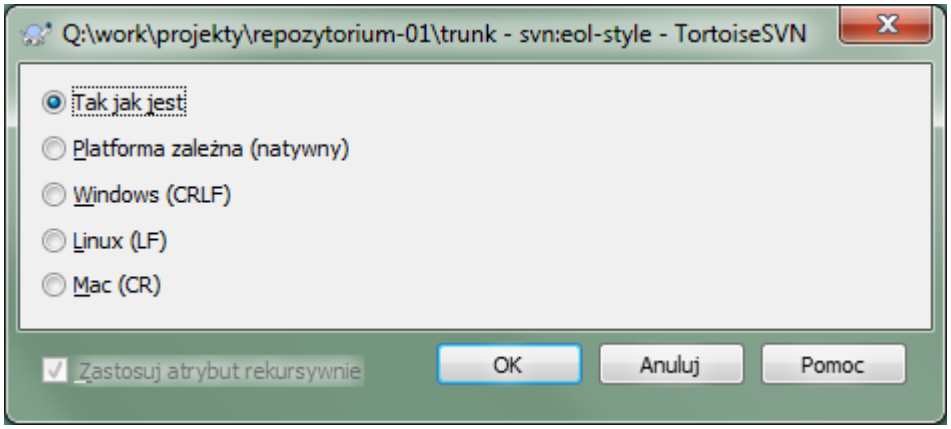
4.18.3.2. Słowa kluczowe SVN



Rysunek 4.45. Strona atrybutu svn:keywords

Wskażcie słowa kluczowe, które chcielibyście rozwinąć w pliku.

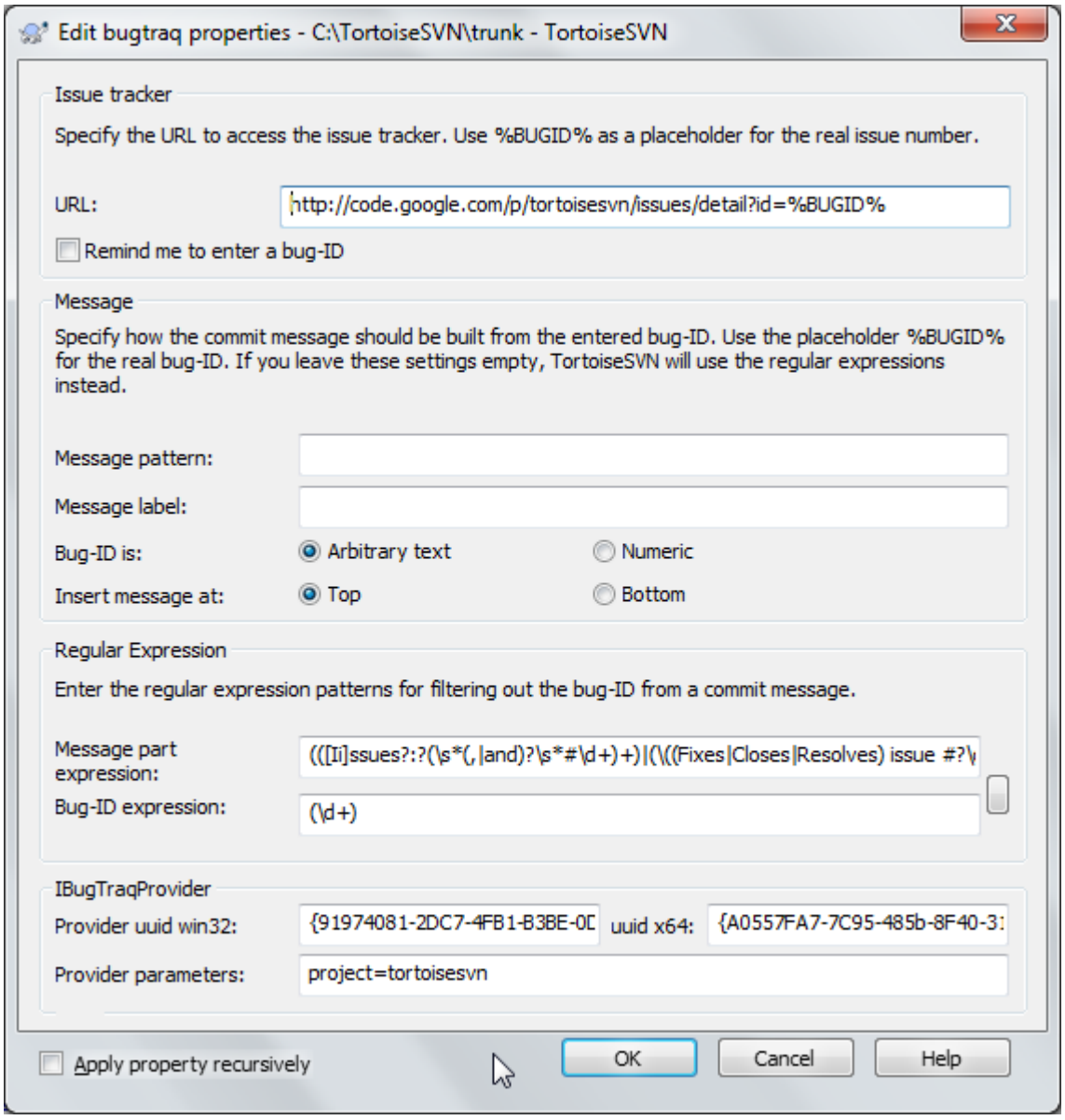
4.18.3.3. Typ EOL



Rysunek 4.46. Strona atrybutu svn:eol-style

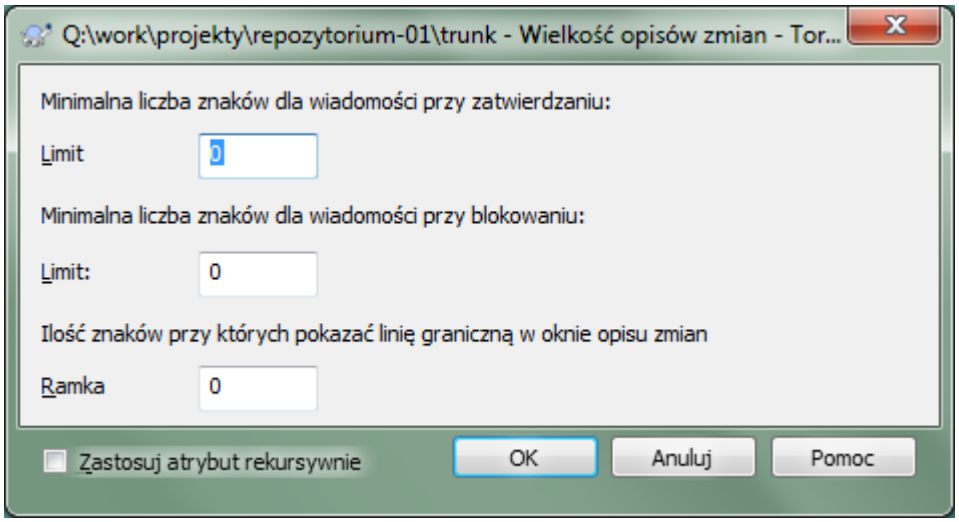
Wybierzcie typ znaku końca linii, który chcecie używać, a TortoiseSVN użyje poprawnej wartości atrybutu.

4.18.3.4. Integracja ze śledzeniem problemów



Rysunek 4.47. Strona atrybutu tsvn:bugtraq

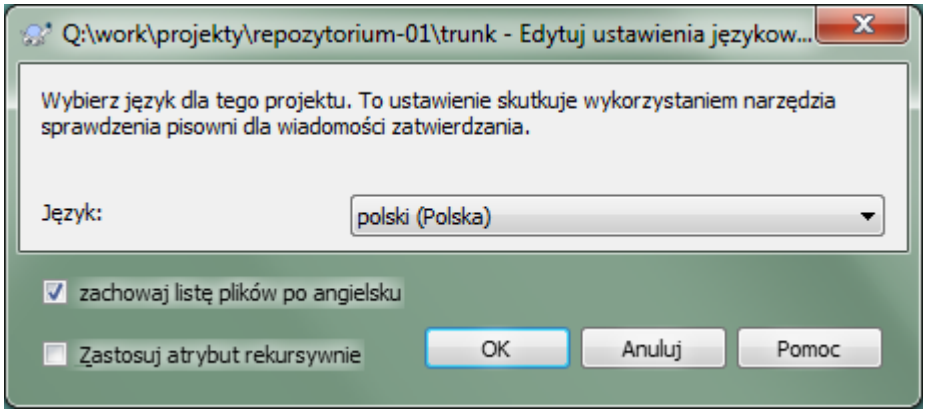
4.18.3.5. Rozmiary opisu zmian



Rysunek 4.48. Strona atrybutu rozmiarów opisu zmian

Te 3 atrybuty sterują formatowaniem komunikatów dziennika. Pierwsze 2 blokują OK w oknach zatwierdzenia oraz blokady dopóki komunikaty nie osiągną minimalnej długości. Pozycja granicy pokazuje znacznik na kolumnie określonej szerokości służący za wskaźnik w projektach posiadających ograniczenia długości komunikatu. Ustawienie wartości zero usuwa atrybut.

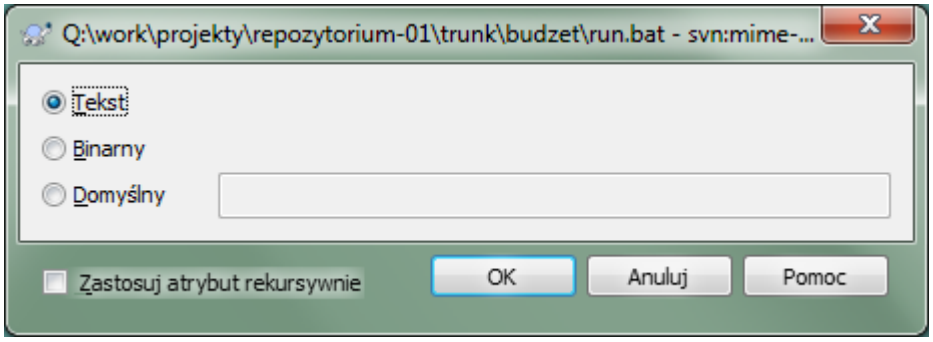
4.18.3.6. Język projektu



Rysunek 4.49. Strona atrybutu języka

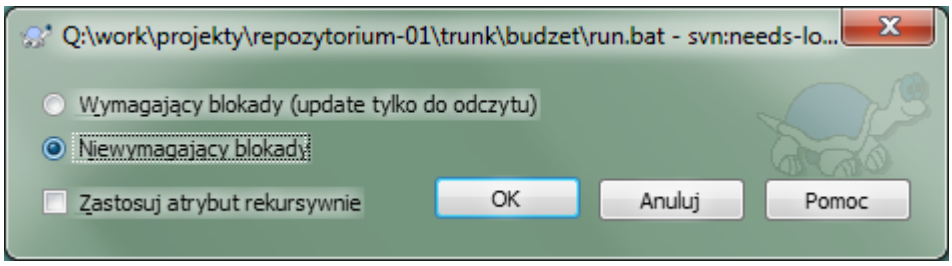
Wybierzcie język, w którym sprawdzane będą komunikaty dziennika w oknie zatwierdzenia. Pole wyboru na liście plików zaczyna działać po wciśnięciu prawego przycisku myszy w panelu komunikatu dziennika i wybraniu Wklej listę plików. Domyślnie status Subversion zostanie pokazany w waszym lokalnym języku. Gdy to pole wyboru jest zaznaczone, status zawsze podawany jest po angielsku, dla projektów wymagających komunikatów dziennika tylko po angielsku.

4.18.3.7. Typ MIME



Rysunek 4.50. Strona atrybutu svn:mime-type

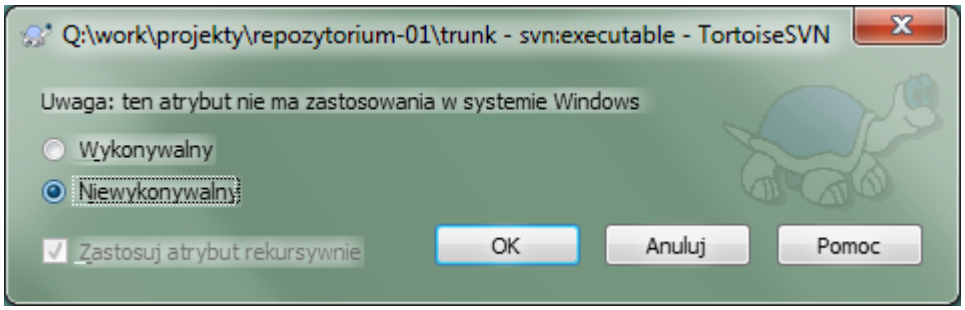
4.18.3.8. svn:needs-lock



Rysunek 4.51. Strona atrybutu svn:needs-lock

Ten atrybut steruje po prostu czy plik zostanie pobrany z atrybutem tylko do odczytu, jeśli nie ma na nim blokady w kopii roboczej.

4.18.3.9. svn:executable



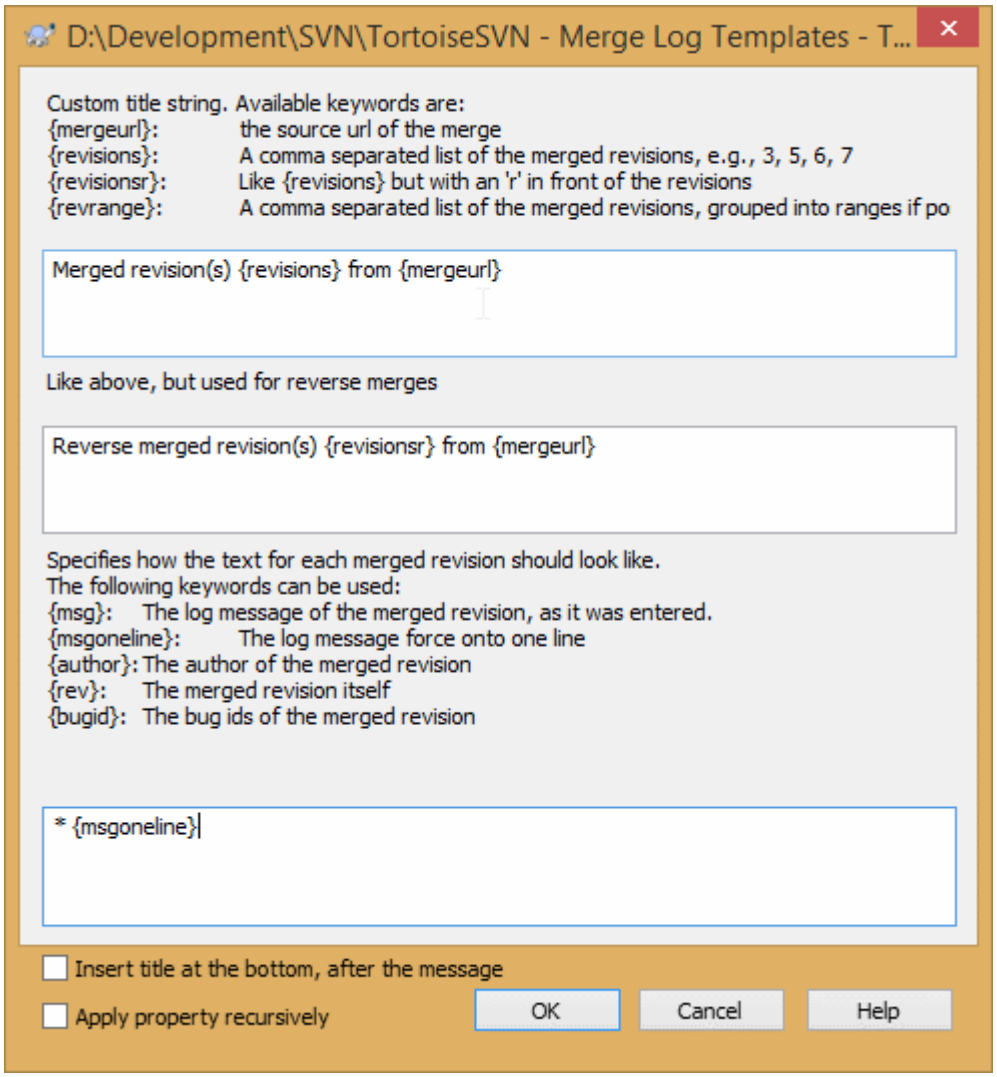
Rysunek 4.52. Strona atrybutu svn:executable

Ten atrybut określa, czy plik otrzyma status wykonywalny po pobraniu na Unix/Linux. Nie ma on wpływu na system Windows.

4.18.3.10. Szablony komunikatów dziennika scaleń

Gdy wersje są scalane do kopii roboczej, TortoiseSVN generuje komunikat dziennika ze wszystkich scalonych wersji. Są one dostępne przez przycisk Ostatnie komunikatyw oknie dialogowym zatwierdzenia.

Możecie dostosować tworzony komunikat korzystając z następujących atrybutów:



Rysunek 4.53. Okno atrybutu szablonów komunikatów dziennika scaleń

tsvn:mergelogtemplatetitle, tsvn:mergelogtemplatereversetitle
: Ten atrybut określa pierwszą część generowanegokomunikatu. Użyte mogą być następujące słowa kluczowe:

  {revisions}
  : Lista rozdzielonych przecinkami wersji, np. `3, 5, 6, 7`

  {revisionsr}
  : Podobnie jak `{revisions}`, jednak z każdą wersją poprzedzoną przez `r`, np., `r3, r5, r6, r7`

  {revrange}
  : Lista rozdzielonych przecinkami wersji, zgrupowanych w przedziały jeśli to możliwe, np. `3, 5-7`

  {mergeurl}
  : URL źródłowy scalenia, tj skąd pochodzą scalane wersje.

  Domyślna wartość ciągu znaków to `Merged revision(s) {revrange} from {mergeurl}:` ze znakiem nowej linii na końcu.

tsvn:mergelogtemplatemsg
: Ten atrybut wskazuje jak powinien wyglądać tekst dla każdej scalonej wersji. Można wykorzystać następujące słowa kluczowe:

  {msg}
  : Komunikat dziennika scalonej wersji, jaki został wpisany.

{msgoneline}
Posobnie jak `{msg}`, ale wszystkie znaki końca linii zastąpione są przez spacje, zatem cały komunikat pojawia się w jednej linii.

{author}
Autor scalonej wersji.

{rev}
Sama scalona wersja.

{bugids}
Identyfikatory błędów scalonej wersji, jeśli są.

tsvn:mergelogtemplatemsgtitlebottom
Ten atrybut wskazuje pozycję tytułu określonego w `tsvn:mergelogtemplatetitle` lub `tsvn:mergelogtemplatereversetitle`. Jeśli atrybut ustawiono na `yes` lub `true`, tytuł jest dopisywany u dołu zamiast u góry.

Ważne

Działa to tylko wtedy, gdy scalone rewizje są już w pamięci podręcznej dziennika. Jeśli zablokowano pamięć podręczną lub nie wyświetlono dziennika przed scaleniem, wygenerowany komunikat nie będzie zawierać żadnych informacji o scalonych wersjach.

4.19. Elementy zewnętrzne

Czasami dobrze jest zbudować kopię roboczą, która jest wykonana w wielu różnych pobraniach. Na przykład, możecie chcieć by różne pliki lub podkatalogi pochodziły z różnych miejsc w repozytorium, a może w ogóle z różnych repozytoriów. Jeśli chcecie, aby każdy użytkownik miał ten sam układ, można zdefiniować atrybut `svn:externals` by ułożyć określone zasoby w miejscach, gdzie są potrzebne.

4.19.1. Foldery zewnętrzne

Powiedzmy, że pobieracie kopię roboczą `/project1` to `D:\dev\project1`. Wybierzcie folder `D:\dev\project1`, kliknijcie prawym przyciskiem myszy i wybierzcie Menu systemu Windows → Właściwości z menu kontekstowego. Pojawi się okno Właściwości. Następnie przejdźcie do zakładki Subversion. Nie można ustawić właściwości. Kliknijcie Atrybuty.... W oknie atrybutów kliknijcie dwukrotnie na `svn:externals` jeśli już istnieje, albo kliknijcie na przycisk Nowy... po czym wybierzcie `zewnętrzne` z menu. Aby dodać nowy zewnętrzny, kliknijcie Nowy..., a następnie wypełnijcie wymagane informacje w pokazanym oknie dialogowym.

Ostrzeżenie

Adresy URL muszą być właściwie enkodowane lub nie będą działać, np. należy zastąpić każdą spację przez `%20`.

Jeśli chcecie by ścieżka lokalna zawierała spacje lub inne znaki specjalne, można ująć ją w cudzysłów, lub użyć znaku `\` (odwrócony ukośnik) jako znak ucieczki powłoki Uniksa poprzedzający znak specjalny. Oczywiście oznacza to również, że należy używać `/` (ukośnik) jako ogranicznik ścieżki. Zauważcie, że to zachowanie pojawiło się dopiero w Subversion 1.6 i nie będzie działać ze starszymi klientami.

Użyj jawnych numerów wersji

Należy poważnie rozważyć używanie jawnych numerów wersji we wszystkich definicjach zewnętrznych, jak to opisano powyżej. Działanie takie oznacza, że trzeba samemu decydować, kiedy wypuścić kolejną migawkę informacji zewnętrznych i jaką dokładnie migawkę wypuścić. Poza

> zdroworozsądkowym aspektem braku zdziwienia zmianami repozytoriów firm trzecich, nad którymi nie ma żadnej kontroli, użycie jawnych numerów wersji oznacza również, że cofnięcie czasowe kopii roboczej do poprzedniej wersji spowoduje, że wasze zewnętrzne definicje będą także przywrócone do wyglądu z tej wcześniejszej wersji, co z kolei oznacza, że zewnętrzne kopie robocze zostaną zaktualizowane, aby dopasować *ichemphasis> wygl*
>
> Okno edycji atrybutów `svn:externals` pozwala wybrać zewnętrzne i automatycznie ustawić je do wersji HEAD.

Jeśli zewnętrzny projekt leży w tym samym repozytorium, wszelkie zmiany w nim wprowadzone będą włączone do listy zatwierdzenia, jeśli użytkownik zatwierdza główny projekt.

Jeśli zewnętrzny projekt znajduje się w innym repozytorium, wszelkie zmiany wprowadzone do zewnętrznego projektu zostaną pokazane lub oznaczone podczas zatwierdzenia głównego projektu, jednak wymagane jest osobne zatwierdzenie tych zewnętrznych zmian.

Jeśli korzystacie z bezwzględnych adresów URL w `svn:externals` definicje i musicie przenieść swoją kopię roboczą (np. jeśli adres URL repozytorium się zmieni), to zewnętrzne nie ulegną zmianie i mogą już nie działać.

Aby uniknąć takich problemów, klienty Subversion w wersji 1.5 i wyższych wspierają w względne adresy URL zewnętrznych. Obsługiwane są cztery różne metody określania względnych adresów URL. W poniższych przykładach, że mamy dwa repozytoria: jedno na `http://example.com/svn/repos-1`, a drugie na `http://example.com/svn/repos-2`. Mamy pobrać z `http://example.com/svn/repos-1/project/trunk` do `C:\Working` a `svn:externals` jest ustawiony na linii głównej.

Względnie do katalogu nadrzędnego
: Adresy te zawsze zaczynają się od ciągu `../` na przykład:

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

  Wypakuje `http://example.com/svn/repos-1/widgets/foo` do `C:\Working\common\foo-widget`.

  Należy pamiętać, że adres URL jest w stosunku do adresu URL katalogu z atrybutu `svn:externals`, a nie do folderu, w którym zewnętrzne są zapisane na dysku.

Względnie do katalogu głównego
: Te adresy URL zawsze zaczynają się od ciągu `^/` na przykład:

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

  Wypakuje `http://example.com/svn/repos-1/widgets/foo` do `C:\Working\common\foo-widget`.

  Możecie łatwo odwołać się do innych repozytoriów z tym samym `SVNParentPath` (wspólny katalog zawierający kilka repozytoriów). Na przykład:

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

  Zostanie rozwinięty do `http://example.com/svn/repos-2/hammers/claw` do `C:\Working\common\claw-hammer`.

Względnie do schematu
: Adresy URL zaczynające się od `//` kopiują tylko schemat z URL. Przydaje się to gdy ta sama nazwa hosta musi być dostępna z różnych systemów w zależności od lokalizacji w sieci, np. klienty w intranecie używają `http://` podczas gdy klienty zewnętrzne używają `svn+ssh://`. Na przykład:

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

Zostanie rozwinięty do `http://example.com/svn/repos-1/widgets/foo` lub `svn+ssh://example.com/svn/repos-1/widgets/foo` w zależności od metody użytej do pobrania `C:\Working`.

Względnie do nazwy serwera hosta
: Adresy URL rozpoczynające się od ciągu `/` kopiują część adresu URL ze schematem i nazwą hosta, na przykład:

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

Spowoduje to wypakowanie `http://example.com/svn/repos-1/widgets/foo` do `C:\Working\common\foo-widget`. Ale jeśli pobierzecie kopię roboczą z innego serwera `svn+ssh://another.mirror.net/svn/repos-1/project1/trunk` wtedy zewnętrzne odniesienie wypakuje `svn+ssh://another.mirror.net/svn/repos-1/widgets/foo`.

Możecie również, jeśli jest to wymagane, wskazać wersję wieszakową i obowiązującą dla URL. By dowiedzieć się więcej o wersjach wieszakowej i obowiązującej, przeczytajcie *odpowiedni rozdział* [http://svnbook.red-bean.com/en/1.8/svn.advanced.pegrevs.html] podręcznika Subversion.

Ważne

Gdy wskażecie jako folder docelowy folder podrzędny jak w przykładach powyżej, upewnijcie się, że *wszystkie* foldery pośrednie są również wersjonowane. Zatem w przykładach powyżej folder `common` powinien być pod kontrolą wersji!

O ile zewnętrzne będą zachowywać się poprawnie w większości sytuacji gdy foldery pomiędzy nimi nie są wersjonowane, mogą wystąpić operacje, które nie będą działać zgodnie z oczekiwaniami. A ikony nakładek statusu w eksploratorze również nie będą pokazywać poprawnego stanu.

Jeśli potrzebujecie więcej informacji, jak TortoiseSVN obsługuje atrybuty, przeczytajcie Sekcja 4.18, „Ustawienia projektu”.

Aby dowiedzieć się o różnych metod dostępu do wspólnych podprojektów przeczytajcie Sekcja B.6, „Dołączanie wspólnego podprojektu”.

4.19.2. Pliki zewnętrzne

Od Subversion 1.6 można dodać jeden zewnętrzny plik do kopii roboczej przy użyciu tej samej składni, jak w przypadku folderów. Istnieją jednak pewne ograniczenia.

- Ścieżka pliku zewnętrznego musi być bezpośrednim potomkiem folderu, w którym ustawiono atrybut `svn:externals`.
- Adres URL dla pliku zewnętrznego musi leżeć w tym samym repozytorium, jak adres URL gdzie ten plik zostanie wstawiony; międzyrepozytoryjne pliki zewnętrzne nie są obsługiwane.

Plik zewnętrzny zachowuje się pod wieloma względami jak każdy inny wersjonowany plik, ale nie można go przesuwać ani usuwać za pomocą normalnych poleceń; zamiast tego musi być zmodyfikowany atrybut `svn:externals`.

4.19.3. Tworzenie zewnętrznych mechanizmem przeciągnij i upuść

Jeśli macie już kopię roboczą plików lub folderów które chcecie dołączyć jako zewnętrzne do innej kopii roboczej, możecie po prostu dodać je używając przeciągnięcia i upuszczenia w eksploratorze windows.

Po prostu przeciągnijcie prawym przyciskiem myszy plik lub folder z pierwszej kopii roboczej tam gdzie mają one być użyte jako zewnętrzne. Pojawi się menu kontekstowe po zwolnieniu przycisku myszy: SVN Dodaj jako zewnętrzne tutaj gdy kliknie się na ten element menu, atrybut `svn:externals` zostanie dodany automatycznie. Wszystko co trzeba potem zrobić to zatwierdzić zmiany atrybutów i zaktualizować by pobrać te zewnętrzne poprawnie dołączone do kopii roboczej.

4.20. Odgałęzianie / etykietowanie

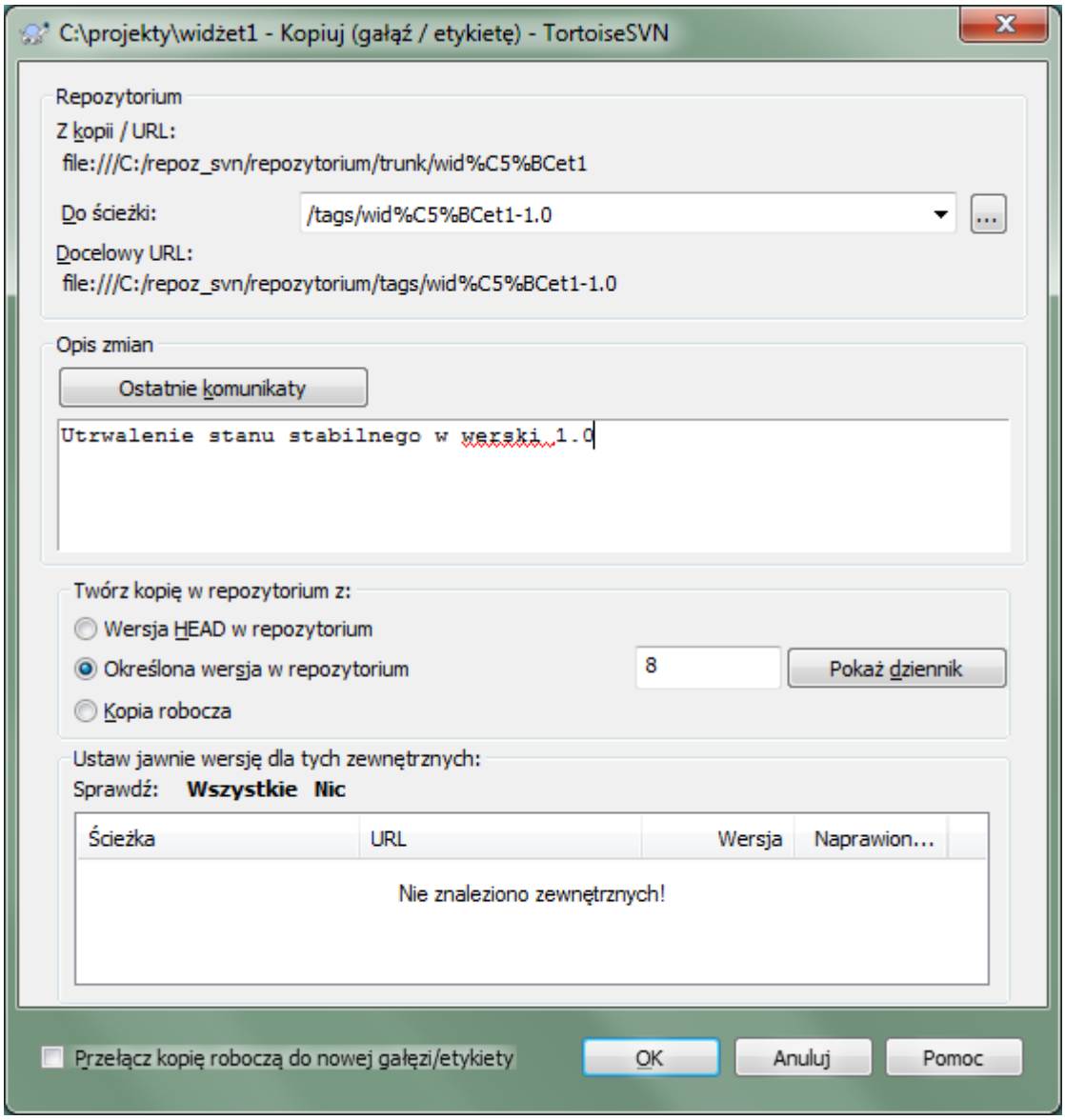
Jedną z cech systemów kontroli wersji jest możliwość wyodrębnienia zmian do osobnej linii rozwoju. Linia ta jest znana jako *gałąź*. Gałęzie są często używane do wypróbowania nowych funkcji bez naruszania głównej linii rozwojowej z błędami kompilacji i algorytmu. Jak tylko nowa funkcja jest wystarczająco stabilna, to gałąź rozwojowa jest *scalana* na powrót do głównej gałęzi (trunk).

Inną cechą systemów kontroli wersji jest możliwość oznaczania poszczególnych wersji (np. wersji wydanych), dzięki czemu można w każdej chwili odtworzyć niektórych kompilacji lub środowiska. Proces ten jest znany jako *etykietowanie*.

Subversion nie posiada specjalnych poleceń do rozgałęzienia lub etykietowania, ale używa zamiast tego tzw „tanich kopii”. Tanie kopie są podobne do twardych linków w systemie Unix, co oznacza, że zamiast tworzenia pełnej kopii w repozytorium, tworzony jest wewnętrzny link, wskazując na konkretne drzewo/wersję. W wyniku gałęzie i etykiety są bardzo szybko tworzone i nie zajmują prawie żadnego dodatkowego miejsca w repozytorium.

4.20.1. Tworzenie gałęzi lub etykiety

Jeśli projekt został zaimportowany do zalecanej struktury katalogów, utworzenie wersji gałęzi lub etykiety jest bardzo proste:



Rysunek 4.54. Okno dialogowe gałęzi/etykiety

Wybierzcie folder, w kopii roboczej, który chcesz skopiować do gałęzi lub etykiety, a następnie wybierzcie polecenie TortoiseSVN → Gałąź/etykieta....

Docelowym domyślnym adresem URL dla nowej gałęzi będzie URL źródła, na którym jest oparta kopia robocza. Będziecie musieli edytować ten adres URL do nowej ścieżki dla gałęzi/etykiety. Stąd zamiast

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

zechcecie raczej teraz użyć czegoś takiego

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

Jeśli nie pamiętacie konwencji nazewnictwa użytej ostatnio, kliknijcie przycisk po prawej stronie, aby otworzyć przeglądarkę repozytorium, dzięki czemu można wyświetlić istniejącą strukturę repozytorium.

foldery pośrednie

Po wskazaniu docelowego URL, wszystkie foldery muszą w nim już istnieć, w przeciwnym razie wyświetli się komunikat błedu. W powyższym przykładzie musi istnieć URL `http://svn.collab.net/repos/ProjectName/tags/` by utworzyć tetykietęag `Release_1.10`.

Jednak jeśli chcecie utworzyć gałąź/etykietę dla adresu URL posiadającego nie istniejące jeszcze foldery pośrednie , możecie zaznaczyć opcję `Utwórz foldery pośrednie` na dole okna. Gdy opcja jest zaznaczona, wszystkie foldery pośrednie zostaną automatycznie utworzone.

Należy zauważyć, opcja ta jest domyślnie wyłączona by uniknąć literówek. Na przykład, jeśli wpisano docelowy URL `http://svn.collab.net/repos/ProjectName/Tags/Release_1.10` zamiast `http://svn.collab.net/repos/ProjectName/tags/Release_1.10`, wystąpi błąd przy wyłączonej opcji, natomiast przy opcji zaznaczonej zostanie automatycznie utworzony folder `Tags`, i skończy się to na istnieniu dwóch folderów `Tags` i `tags`.

Teraz musicie wybrać źródło kopii. Są dostępne trzy opcje:

Wersja HEAD w repozytorium
: Nowa gałąź jest kopiowana bezpośrednio w repozytorium z wersji HEAD. Nie trzeba danych przenosić danych z kopii roboczej, a gałąź zostaje utworzona bardzo szybko.

Określona wersja w repozytorium
: Nowa gałąź jest kopiowana bezpośrednio w repozytorium, ale można wybrać starszą wersję. Jest to przydatne, jeśli zapomnieliście zrobić etykietę kiedy wydawaliście swój projekt w ubiegłym tygodniu. Jeśli nie pamiętacie numeru wersji, kliknijcie przycisk po prawej stronie, aby wyświetlić dziennik zmian, a następnie wybierzcie stamtąd numer wersji. Znowu dane nie są przesyłane z kopii roboczej, a gałąź stworzy się bardzo szybko.

Kopia robocza
: Nowa gałąź jest identyczną kopią lokalnej kopii roboczej. Jeśli zaktualizowaliście pliki do starszej wersji w KR, lub w przypadku dokonania lokalnych zmian, dokładnie to trafia do kopii. Oczywiście tego rodzaju skomplikowana etykieta może powodować przesyłanie danych z KR do repozytorium, jeśli jeszcze ich tam nie ma.

Jeśli chcecie, aby kopia robocza była przełączona do nowo utworzonej gałęzi automatycznie, użyjcie pola wyboru Przełącz kopię roboczą na nową gałąź/etykietę. Ale jeśli to zrobicie, upewnijcie się najpierw, że kopia robocza nie zawiera zmian. Jeśli tak, to zmiany te zostaną połączone do gałęzi KR po przełączeniu.

Jeśli kopia robocza posiada inne projekty ujęte w atrybutach `svn:externals`, te zewnętrzne będą wypisane na dole okna dialogowego gałęzi/etykiety. Dla każdego zewnętrznego pokazana jest ścieżka docelowa i URL źródła.

Jeśli chcecie upewnić się, że nowa etykieta zawsze jest w stanie spójnym, sprawdźcie czy wszystkie zewnętrzne mają przypięte. Jeśli nie sprawdzicie zewnętrznych i te zewnętrzne wskazują na wersję HEAD, która może się zmienić w przyszłości, podczas pobierania nowej etykiety nastąpi pobranie wersji HEAD zewnętrznych i etykieta może się już nie skompilować. Więc to jest zawsze dobry pomysł, aby ustawić zewnętrzne na ustaloną wersję podczas tworzenia etykiety.

Zewnętrzne są zawsze przypinane do ich bieżącej wersji HEAD lub wersji BASE kopii roboczej, zależnie od źródła gałęzi/etykiety:

Źródło Kopii	Przypięta Wersja
Wersja HEAD w repozytorium	wersja HEAD repozytorium zewnętrznego
Określona wersja w repozytorium	wersja HEAD repozytorium zewnętrznego
Kopia robocza	wersja BASE kopii roboczej zewnętrznego

Tabela 4.1. Przypięta Wersja

zewnętrzny w zewnętrznych

Jeśli projekt zawarty jako zewnętrzny sam również zawierał zewnętrzne, to one nie zostaną zaetykietowane! Tylko zewnętrzne które są bezpośrednim potomkiem mogą być etykietowane.

Naciśnijcie OK by zatwierdzić nową kopię do repozytorium. Nie zapomnijcie wpisać opisu zmian. Należy pamiętać, że kopia jest tworzona *wewnątrz repozytorium*.

Pamiętajcie, że jeśli zdecydowaliście się przełączyć swoją kopię roboczą do nowo utworzonej gałęzi, tworzenie gałęzi lub etykiety *nie ma* wpływu na kopię roboczą. Nawet po utworzeniu gałęzi z KR, zmiany te są zatwierdzane do nowej gałęzi, nie do linii głównej, więc KR może być jeszcze oznaczone jako zmodyfikowane w stosunku do linii głównej.

4.20.2. Inne sposoby tworzenia gałęzi lub etykiety

Można również utworzyć gałąź lub etykietę bez kopii roboczej. Aby to zrobić, otwórzcie przeglądarkę repozytorium. Można tam przeciągnąć foldery w nowe miejsce. Musicie przytrzymać klawisz **Ctrl** podczas przeciągania, aby utworzyć kopię, w przeciwnym wypadku folder zostanie przeniesiony, a nie skopiowany.

Można też przeciągnąć folder prawym przyciskiem myszy. Po zwolnieniu przycisku myszy można wybrać z menu kontekstowego, czy należy go przenieść czy skopiować. Oczywiście, aby utworzyć gałąź lub etykietę należy skopiować folder, a nie przesuwać.

Jeszcze innym sposobem jest użycie okna dziennika. Możecie pokazać okno dziennika do np. linii głównej, wybierzcie wersję (wersję HEAD na samej górze albo starszą wersję), kliknijcie prawym przyciskiem myszy i wybierzcie utwórz gałąź/etykietę z wersji....

4.20.3. Pobierać czy przełączać...

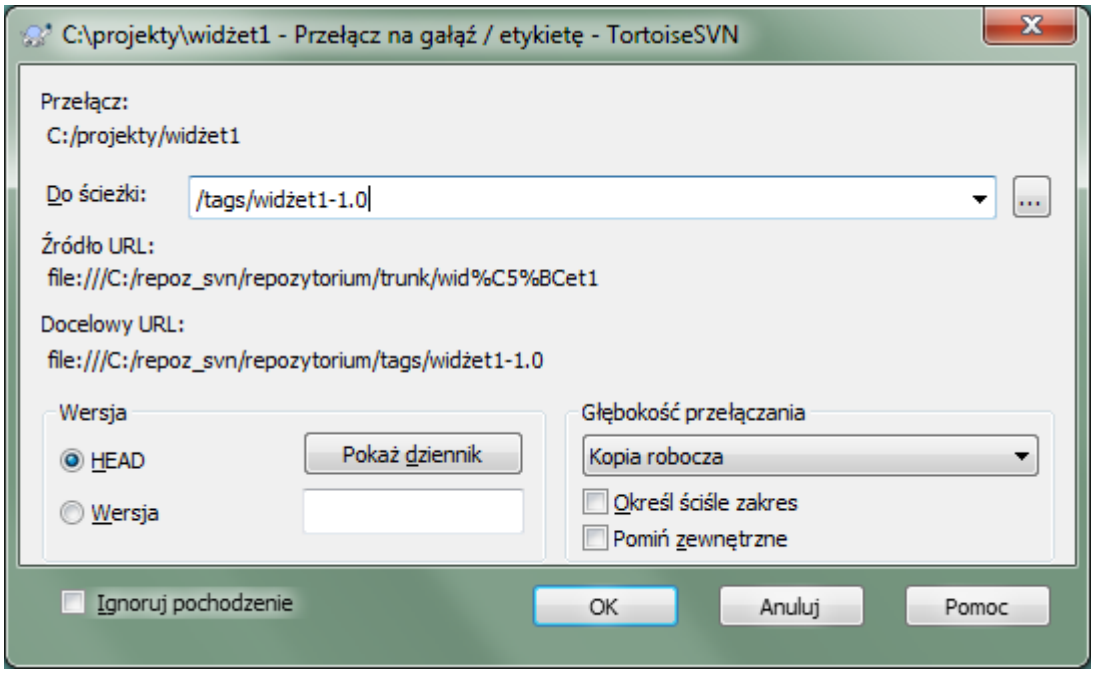
...oto jest (nie do końca) pytanie. Podczas pobierania wczytywana całość gałęzi z repozytorium do folderu roboczego, TortoiseSVN → Przełącz... przekazuje tylko zmienione dane do kopii roboczej. Lepiej dla obciążenia sieci, lepiej dla cierpliwości. :-)

Aby móc pracować ze świeżo wygenerowaną gałęzią lub etykietą macie kilka sposobów, aby sobie z tym poradzić. Można:

- TortoiseSVN → Przełącz aby wykonać świeże pobranie w pustym folderze. Możecie, jak chcecie, pobrać do dowolnej lokalizacji na dysku lokalnym i utworzyć dowolną liczbę kopii roboczych z repozytorium.
- Przełącie bieżącą kopię roboczą do nowo utworzonej kopii w repozytorium. Ponownie zaznaczcie folder najwyższego poziomu projektu i użyjcie TortoiseSVN → Przełącz... z menu kontekstowego.

  W następnym oknie dialogowym wpiszcie adres URL gałęzi, którą właśnie utworzyliście. Wybierzcie Wersja HEAD, a następnie przycisk na OK. Kopia robocza jest przełączona do nowej gałęzi/etykiety.

  Przełączenie działa jak aktualizacja, bo nigdy nie odrzuca zmian lokalnych. Wszelkie zmiany wprowadzone do kopii roboczej, które jeszcze nie zostały zatwierdzone zostaną scalone, gdy przełączacie. Jeśli nie chcecie, aby tak się stało, musicie albo zatwierdzić zmiany przed przełączeniem albo przywrócić kopię roboczą do wersji już zatwierdzonej (zazwyczaj HEAD).
- Jeśli chcecie pracować na linii głównej i gałęzi, ale nie kosztem świeżego pobrania, możecie użyć eksploratora Windows, aby wykonać kopię pobrania linii głównej w innym folderze, a następnie TortoiseSVN → Przełącz... kopiuje do swojej nowej gałęzi.



Rysunek 4.55. Okno dialogowe przełączenia

Chociaż Subversion się nie czyni rozróżnienia pomiędzy etykietami i gałęziami, sposób w jaki są zwykle używane różni się trochę.

- Etykiety są zazwyczaj wykorzystywane do tworzenia statycznego obrazu projektu na określonym etapie. Jako takie nie są zwykle używane dla rozwoju - to gałęzie są od tego. Jest to powodem zalecenia struktury repozytorium `/trunk /branches /tags` w pierwszej kolejności. Praca na wersji etykiety nie jest *dobrym pomysłem*, ale ponieważ lokalne pliki nie są zabezpieczone przed zapisem nie ma nic, aby powstrzymać Was przed pomyłką. Jednakże, jeśli spróbujecie zatwierdzić coś na ścieżce w repozytorium, która zawiera `/tags/`, TortoiseSVN wyświetli ostrzeżenie.

- Być może trzeba wprowadzić dalsze zmiany do wydania które już zaetykietowaliście. Poprawny sposób obsługi jest taki, aby utworzyć najpierw nową gałąź z etykiety i zatwierdzać na gałęzi. Wykonajcie zmiany na tej gałęzi, a następnie utwórzcie nową etykietę z tej nowej gałęzi, np. `Version_1.0.1`.

- Jeśli zmodyfikowaliście kopię roboczą utworzoną z gałęzi i zatwierdziliście, to następnie wszystkie zmiany przechodzą do nowej gałęzi a *nie* linii głównej. Tylko zmiany są przechowywane. Reszta pozostaje tanią kopią.

4.21. Scalenie

W przypadku, gdy gałęzie są używane do utrzymywania oddzielnych linii rozwoju, na pewnym etapie będziemy chcieli, aby scalić zmiany dokonane na jednej gałęzi z powrotem do linii głównej lub odwrotnie.

Ważne jest, aby zrozumieć, jak odgałęzienia i scalenia działają w Subversion, zanim zaczniecie ich używać, ponieważ mogą stać się bardzo skomplikowane. Zalecane jest, aby zapoznać się z rozdziałem *Rozgałęzianie i scalanie* [http://svnbook.red-bean.com/en/1.8/svn.branchmerge.html] w podręczniku Subversion, który daje pełny opis i liczne przykłady, jak ich używać.

Kolejnym punktem do zaznaczenia jest, że połączenie *zawsze* ma miejsce w kopii roboczej. Jeśli chcecie scalić zmiany *do* gałęzi, trzeba mieć pobraną kopię roboczą dla tej gałęzi i wywołać kreatora scaleń z kopii roboczej za pomocą TortoiseSVN → Scalaj....

Ogólnie jest to dobry pomysł, aby wykonać połączenie w niezmodyfikowanej kopii roboczej. Jeśli dokonano innych zmian w KR, najpierw je zatwierdźcie. Jeśli scalenie nie wykonało się zgodnie z oczekiwaniami, możecie cofnąć zmiany, a polecenie Wycofaj zmiany usunie *wszystkie* zmiany, w tym dokonane przed scaleniem.

Istnieją trzy typowe przypadki użycia dla scalenia, które są traktowane w nieco odmienny sposób, co opisano poniżej. Na pierwszej stronie kreatora scalenia jest prośba o wybranie metody, której potrzebujecie.

Scalenie grupy wersji
: Metoda ta dotyczy przypadku, gdy po dokonaniu jednej lub więcej wersji do gałęzi (lub linii głównej) i chcecie przenieść te zmiany w do innej gałęzi.

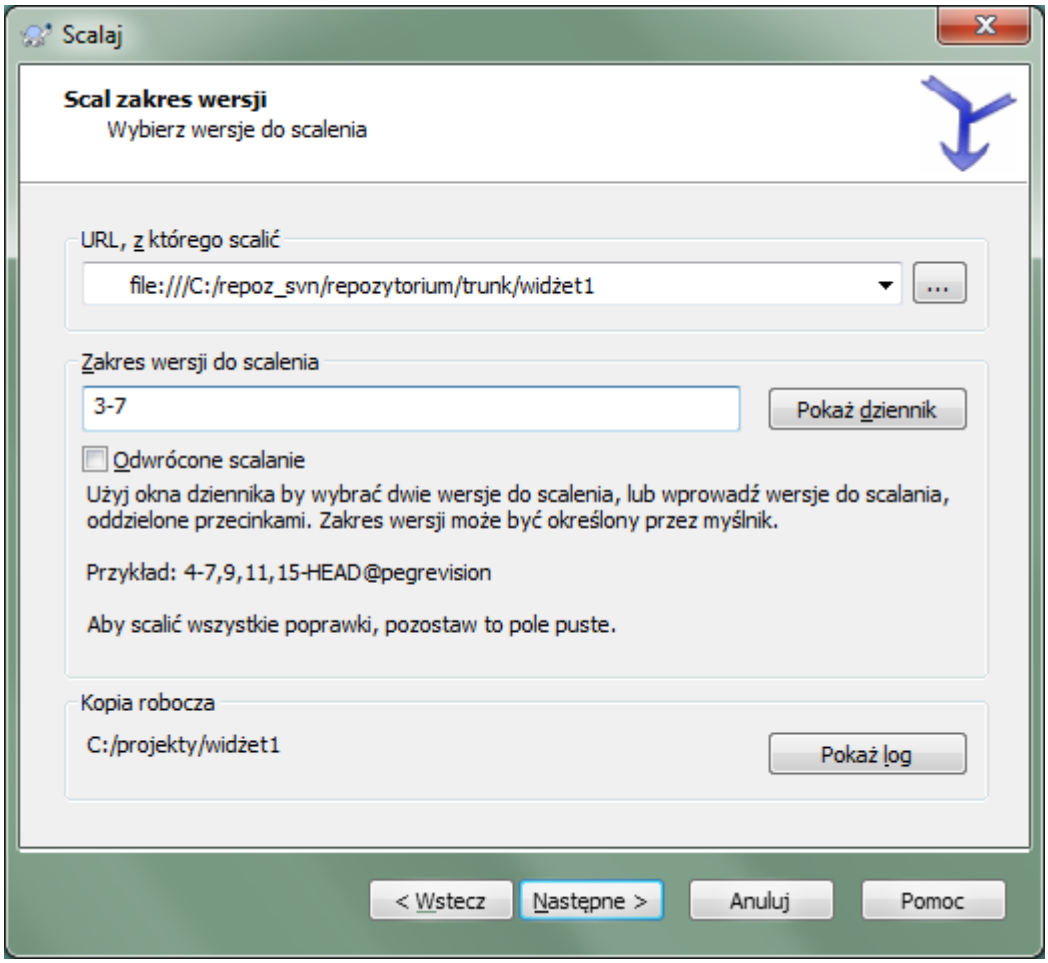
  Wymagacie by Subversion wykonało: „ Obliczenie zmian niezbędnych do uzyskania [OD] wersji 1 gałęzi A [DO] wersji 7 z gałęzi A, i zastosowanie tych zmian do kopii roboczej (linii głównej lub gałęzi B). ”

  Jeśli pozostawisz pusty zakres rewizji, Subversion używa funkcjonalności śledzenia zmian aby wyliczyć prawidłowy zakres rewizji. Funkcjonalność jest znana jako reintegracja lub łączenie automatyczne.

Scalenie dwóch różnych drzew
: Jest to bardziej ogólny przypadek metody reintegracji. Prosicie Subversion o wykonanie działań: „ Obliczyć zmiany niezbędne do uzyskania [OD] wersji HEAD linii głównej [DO] wersji head gałęzi i zastosować te zmiany do kopii roboczej (linii głównej). ” W efekcie linia główna wygląda teraz dokładnie jak gałąź.

  Jeśli serwer / repozytorium nie obsługuje śledzenia scaleń, to jest jedyny sposób, aby połączyć gałąź z powrotem do linii głównej. Innym przypadkiem wykorzystania występuje wtedy, gdy używacie gałęzi dostawcy i trzeba scalić zmiany wynikające z nowej łaty dostawcy do kodu linii głównej. Aby uzyskać więcej informacji przeczytajcie rozdział o *gałęzie dostawców* [http://svnbook.red-bean.com/en/1.8/svn.advanced.vendorbr.html] w Podręczniku Subversion.

4.21.1. Scalenie grupy wersji



Rysunek 4.56. Kreator scalenia - Wybierz zakres wersji

W polu Z: należy wpisać pełny adres URL folderu gałęzi lub etykiety zawierającego żądane zmiany do dołączenia w kopii roboczej. Możecie też kliknąć ... by przejrzeć repozytorium i znaleźć odpowiednią gałąź. Jeśli scalaliście już z tej gałęzi, to wtedy po prostu skorzystajcie z listy rozwijanej, która pokazuje historię poprzednio używanych adresów URL.

Jeśli scalacie z gałęzi przemianowanej lub usuniętej, będziecie musieli cofnąć się do wersji, w której gałąź wciąż istniała. W tym przypadku trzeba również wskazać tą wersję jako wersję wieszakową w zakresie wersji do scalenia (patrz poniżej), w przeciwnym wypadku scalenie nie powiedzie się gdy nie będzie w stanie odnaleźć tej ścieżki w HEAD.

W polu Zakres wersji do scalenia wpiszcie listę wersji, które chcecie scalić. To może być jedna wersja, wykaz konkretnych wersji oddzielonych przecinkami lub zakres wersji oddzielonych myślnikiem, lub dowolna ich kombinacja.

Jeśli chcecie, aby wskazać wersję wieszakową dla scalenia, należy dodać wersję wieszakową na końcu wersji, np. `5-7,10@3`. W powyższym przykładzie, wersje 5,6,7 i 10 zostaną scalone z 3, która jest wersją wieszakową.

Ważne

Istnieje istotna różnica w sposobie w jaki zakres zmian zostaje określony za pomocą TortoiseSVN w porównaniu z klientem linii poleceń. Najprostszym sposobem wizualizacji jest wyobrazić sobie o ogrodzeniu jako płocie i sztachetach.

Z klientem linii poleceń można określić zmiany na połączenie za pomocą dwóch „płotów” wersji, które określają punkty *przed* i *po*.

Z TortoiseSVN należy określić zestaw zmian do scalenia przy użyciu „sztachet”. Powód tego staje się jasny, kiedy podczas korzystania z okna dziennika do określenia wersji do scalenia każda wersja pojawia się jako zestaw zmian.

Gdy scalacie wersje kawałkami, według metody pokazanej w książce Subversion powinniście połączyć wersje 100-200 teraz i 200-300 następnym razem. Zgodnie z TortoiseSVN powinniście scalić 100-200 teraz oraz 201-300 następnym razem.

Różnica ta wniosła dużo ognia na listach dyskusyjnych. Zdajemy sobie sprawę, że istnieje różnica w stosunku do klienta linii poleceń, ale wierzymy, że dla większości użytkowników GUI łatwiejszy jest do zrozumienia sposób, który wdrożyliśmy.

Najprostszym sposobem, aby wybrać potrzebny zakres wersji jest kliknięcie na Pokaż dziennik, ponieważ wypisuje listę ostatnich zmian z ich opisami. Jeśli chcecie scalić zmiany z jednej wersji, zaznaczcie tą wersję. Jeśli chcecie scalić zmiany z kilku wersji, wybierzcie ten zakres (za pomocą zwykłego modyfikatora **Shift**). Kliknijcie na OK a wykaz numerów wersji do scalenia zostanie automatycznie wypełniony.

Jeśli chcecie scalić zmiany z powrotem *spoza* kopii roboczej, aby przywrócić zmiany, które już zostały zatwierdzone, wybierzcie wersje do wycofania i upewnijcie się, że pole wyboru Odwrócone scalanie jest zaznaczone.

Jeśli już scalono pewne zmiany z tej gałęzi, mam nadzieję, że będziecie musieli wpisać notatkę dla ostatniej scalonej wersji w opisie zmiany podczas zatwierdzenia zmiany. W takim przypadku można użyć Pokaż dziennik na kopii roboczej do śledzić ten opis zmiany. Pamiętając, że myślimy o wersji jako zestawie zmian, należy użyć wersji kolejnej po punkcie kończącym ubiegłe scalenie jako punkt startowy dla tego scalenia. Na przykład, jeśli macie już scalone ostatnim razem wersje 37 do 39, to punktem wyjściowym do tego scalania powinna być wersja 40.

Jeśli używacie funkcji śledzenia scaleń Subversion, nie musicie pamiętać, które wersje zostały już połączone - Subversion będzie to notować dla Was. Jeśli pozostawić pusty zakres wersji, wszystkie wersje, które jeszcze nie zostały scalone zostaną uwzględnione. Czytajcie Sekcja 4.21.5, „Śledzenie scalania”, aby dowiedzieć się więcej.

Kiedy używane jest śledzenie scaleń, okno dziennika pokaże uprzednio scalone wersje, i wersje poprzedzające punkt wspólnego przodka, tj. zanim gałąź została skopiowana, jako wyszarzone. Pole wyboru Ukryj niescalalne

wersje pozwala na odfiltrowanie tych wersji całkowicie tak aby było widać tylko wersje, które *mogą* zostać scalone.

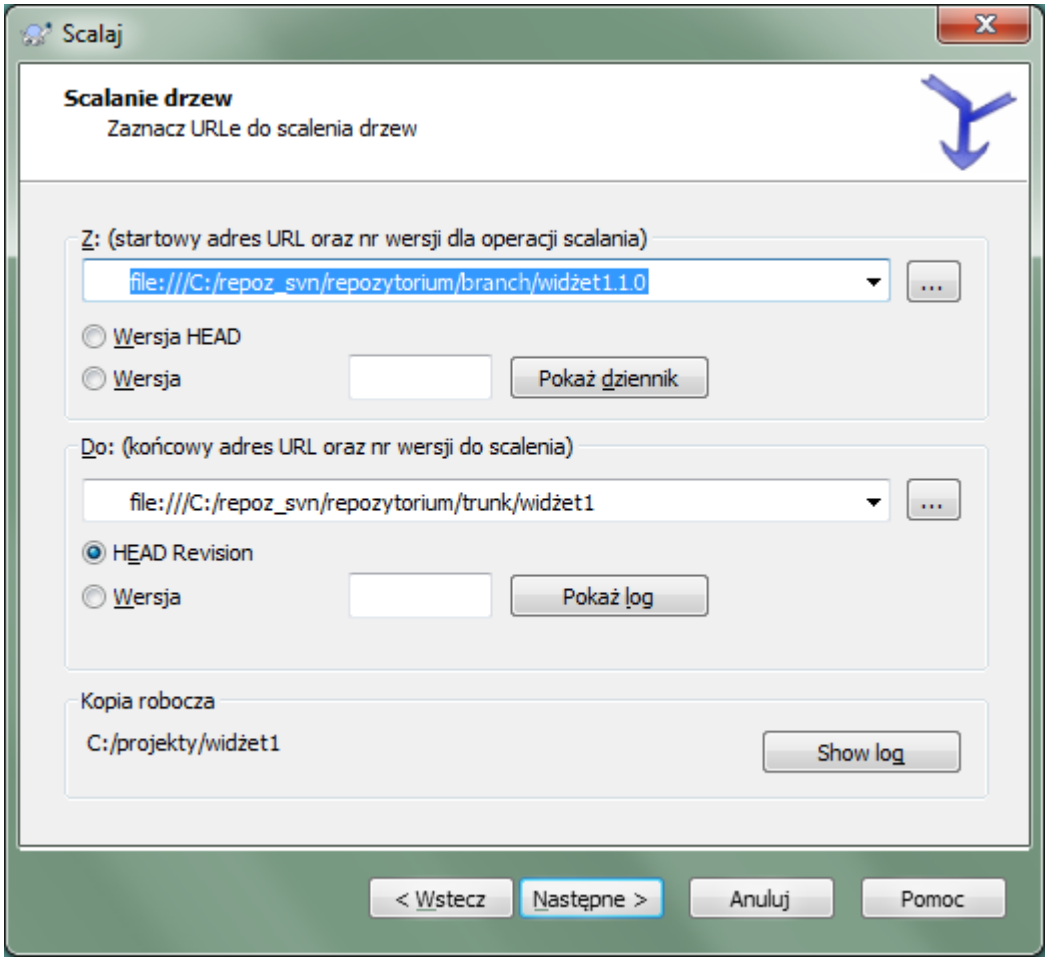
Jeśli inni ludzie być może zatwierdzili zmiany, bądźcie ostrożni używając wersji HEAD. Może ona nie odnosić się do wersji o której myślicie, jeśli ktoś zatwierdził po Waszej ostatniej aktualizacji.

Jeśli pozostawisz pusty zakres rewizji lub wybierzesz wszystkie rewizje z pola rozwijanego, Subversion złączy wszystkie jeszcze nie połączone rewizje. Funkcjonalność jest znana jako reintegracja lub łączenie automatyczne.

Istnieją pewne warunki, które stosuje się do scalenia reintegrowanego. Po pierwsze, serwer musi obsługiwać śledzenie scaleń. Kopia robocza musi mieć głębokość nieskończoną (żadnych rzadkich aktualizacji) i nie może mieć żadnych lokalnych modyfikacji, przełączonych elementów ani elementów, które zostały uaktualnione do wersji innej niż HEAD. Wszystkie zmiany linii głównej dokonane w trakcie rozwoju gałęzi muszą być scalone do całej gałęzi (lub oznaczone jako scalone). Zakres wersji do scalenia zostanie obliczony automatycznie.

Wciśnijcie Następne i przejdźcie do Sekcja 4.21.3, „Opcje scalania”.

4.21.2. Scalenie dwóch różnych drzew



Rysunek 4.57. Kreator scalenia - Scalanie drzew

Jeśli korzystacie z tej metody, aby scalić gałąź funkcji z powrotem do linii głównej, trzeba, aby uruchomić kreatora scalania w kopii roboczej linii głównej.

W polu Z: należy wpisać pełny adres URL folderu z *linii głównej*. To może źle zabrzmieć, ale pamiętajcie, że linia główna jest punktem początkowym, do którego chcecie dodać zmiany gałęzi. Możecie też kliknąć ... by przeglądać repozytorium.

W polu Do: należy wpisać pełny adres URL folderu gałęzi funkcji.

W obu polach Od wersji i Do wersji wprowadź numer ostatniej wersji, w której oba drzewa były zsynchronizowane. Jeżeli jesteście pewni, że nikt inny nie robi zatwierdzeń można użyć wersji HEAD, w obu przypadkach. Jeśli jest szansa, że ktoś inny mógł wykonać zatwierdzenie od czasu synchronizacji, skorzystajcie z określonego numeru wersji, aby uniknąć utraty najnowszych zatwierdzeń.

Możecie również użyć Pokaż dziennik do wybrania wersji.

4.21.3. Opcje scalania

Ta stron kreatora pozwala ustawić opcje zaawansowane, przed rozpoczęciem procesu scalania. W większości przypadków wystarczy używać ustawień domyślnych.

Możecie określić głębokość użytą do scalenia, czyli jak daleko w kopii roboczej powinny się wykonać scalenia. Warunki głębokości są opisane w Sekcja 4.3.1, „Głębokość pobierania”. Głębokość domyślna to Kopia robocza, która używa istniejących ustawień głębokości, i jest to prawie zawsze to, co trzeba.

W większości przypadków chcecie by scalenie uwzględniało historię pliku, więc zmiany w stosunku do wspólnego przodka zostają scalane. Czasami trzeba scalić pliki, które są być może powiązane, ale nie w repozytorium. Na przykład może być zaimportowano wersje 1 i 2 biblioteki strony trzeciej do dwóch odrębnych katalogów. Mimo, że są logicznie powiązane, Subversion nie ma wiedzy o tym, a widzi tylko zaimportowane pliki. Jeśli spróbujecie połączyć różnice pomiędzy tymi dwoma drzewami, to zobaczycie całkowite usunięcie, a następnie pełne dodanie. Aby Subversion używał różnic opartych na ścieżce, a nie różnic opartych na historii, zaznacz pole Ignoruj pochodzenie. Czytajcie więcej na ten temat w podręczniku Subversion, *Dostrzeganie lub ignorowanie pochodzenia* [http://svnbook.red-bean.com/en/1.8/svn.branchmerge.advanced.html#svn.branchmerge.advanced.ancestry].

Możecie określić sposób, w jaki zmiany znaku zakończenia linii i białe znaki są obsługiwane. Opcje te zostały opisane w Sekcja 4.11.2, „Opcje końca linii i białych znaków”. Domyślnym zachowaniem jest traktować wszystkie białe znaki i końca linii różnice rzeczywiste zmiany do wykonania scalenia.

Pole wyboru oznaczone Wymuś scalanie służy do uniknięcia konfliktu drzewa, gdzie przychodzące usunięcie wpływa na plik zmodyfikowany lokalnie lub w ogóle nie wersjonowany. Jeżeli plik zostanie usunięty, to nie ma sposobu aby go odzyskać, dlatego ta opcja nie jest zaznaczona domyślnie.

Jeśli używacie śledzenia scaleń i chcecie oznaczyć wersję jako scaloną, bez faktycznego wykonania scalania, zaznaczcie pole wyboru Tylko zapisz fakt scalania. Istnieją dwa możliwe powody zrobienia tego. Możliwe, że scalenie jest zbyt skomplikowane dla algorytmów scalenia, więc wprowadziliście zmiany ręcznie, a następnie zaznaczcie zmiany jako scalone, by algorytm śledzenia scaleń był o tym poinformowany. A może chcecie, aby zapobiec by scalać szczególną wersję. Oznaczenie jej jako już scalonej uniemożliwi scalenia przeprowadzane przez klienty obsługujące śledzenie scaleń.

Teraz wszystko jest ustawione, musicie jedynie kliknąć na przycisk Scalanie. Jeśli chcecie tylko obejrzeć wyniki, to Testuj scalanie symuluje operację scalania, ale *nie* zmienia w ogóle kopii roboczej. Pokazuje tylko listę plików, które będą zmieniane przez prawdziwe scalenie, a zaznacza pliki, gdzie konflikty *mogą* wystąpić. Ponieważ śledzenie scaleń czyni scalenie znacznie bardziej skomplikowanym, nie można zagwarantować, aby dowiedzieć się wcześniej, czy scalenie zakończy się bez konfliktów, więc pliki oznaczone jako skonfliktowane w teście scalą się w rzeczywistości bez problemu.

Okno dialogowe postępu scalenia pokazuje każdy etap scalania z zakresem zaangażowanych wersji. Może to oznaczać jedną wersję więcej, niż się spodziewaliśmy. Na przykład, jeśli zażądano scalenia wersji 123, okno dialogowe postępu zaraportuje „ Scalenie wersji 122 do 123 ”. Aby to zrozumieć trzeba pamiętać, że Scal jest ściśle związana z Porównaj. Proces scalania działa, generując listę różnic między dwoma punktami w repozytorium i stosując te różnice do kopii roboczej. Dialog postępu po prostu pokazuje początkowy i końcowy punkt różnicy.

4.21.4. Przeglądanie wyników scalania

Scalanie zostało zakończone. To dobry pomysł, aby przyjrzeć się scaleniu i sprawdzić czy jest ono zgodnie z oczekiwaniami. Scalanie jest zwykle dość skomplikowane. Konflikty pojawiają się często wtedy, gdy gałąź odpłynęła daleko od linii głównej.

Podpowiedź

Gdy wersje są scalane do kopii roboczej, TortoiseSVN generuje komunikat dziennika ze wszystkich scalonych wersji. Są one dostępne przez przycisk Ostatnie komunikatyw oknie dialogowym zatwierdzenia.

By dostosować ten wygenerowany komunikat, ustaw odpowiednie atrybuty projektu w kopii roboczej. Więcej w Sekcja 4.18.3.10, „Szablony komunikatów dziennika scaleń”

Dla klientów i serwerów Subversion przed 1.5, żadne informacje o scaleniu nie są przechowywane i scalone wersje muszą być śledzone ręcznie. Po przetestowaniu zmian i dojściu do zatwierdzenia tej zmiany, Wasz opis zmiany do dziennika powinien *zawsze* zawierać numery wersji, które zostały przeniesione w scaleniu. Jeśli chcecie zastosować inne scalanie w późniejszym czasie trzeba będzie wiedzieć, co już zostało scalone, jeśli nie chcecie nanieść zmiany więcej niż raz. Aby uzyskać więcej informacji na ten temat, patrzcie *Najlepsze praktyki scalania* [http://svnbook.red-bean.com/en/1.4/svn.branchmerge.copychanges.html#svn.branchmerge.copychanges.bestprac] w księdze Subversion.

Jeśli serwer i wszystkie klienty używają Subversion 1.5 lub wyższej, Funkcja śledzenia scaleń rejestruje scalone wersje i unika scalania wersji więcej niż raz. To czyni życie znacznie prostszym gdyż można po prostu scalić w pełny zakres wersji za każdym razem, i wiedzieć, że tylko nowe wersje zostaną rzeczywiście scalone.

Zarządzanie gałęziami jest bardzo ważne. Jeśli chcecie utrzymać tą gałąź na bieżąco z linią główną, należy się upewnić, by scalać tak często, by gałąź i linia główna nie odpłynęły zbyt daleko od siebie. Oczywiście, należy wciąż unikać wielokrotnego scalenia zmian, jak wyjaśniono powyżej.

Podpowiedź

Jeśli właśnie scalono gałąź funkcji z powrotem do linii głównej, linia główna zawiera teraz całość nowego kodu funkcji, a gałąź jest już nieaktualna. Możecie ją teraz usunąć z repozytorium jeśli trzeba.

Ważne

Subversion nie może scalić pliku z folderem ani odwrotnie - tylko foldery do folderów i pliki do plików. Jeśli klikniesz na plik i otworzyć okno dialogowe scalania, to trzeba podać ścieżkę do pliku w tym oknie dialogowym. Po wybraniu folderu i pojawieniu się okna dialogowego, należy określić adres URL folderu do scalenia.

4.21.5. Śledzenie scalania

Subversion 1.5 wprowadza udogodnienia dla śledzenia scaleń. Podczas scalania zmian z jednego drzewa do drugiego, scalone numery wersji są przechowywane i informacje te mogą być wykorzystywane do różnych celów.

- Możesz uniknąć zagrożenia scaleniem tej samej wersji dwa razy (problem powtarzającego się scalenia). Gdy już wersja jest oznaczona jako scalona, przyszłe scalenia, które ją obejmują zakresem, przeskoczą ją.
- Podczas scalania gałęzi z powrotem do linii głównej, okno dziennika może pokazać zatwierdzenia gałęzi jako część dziennika linii głównej, dając lepsze możliwości śledzenia zmian.
- Po wyświetleniu dialogu dziennika z poziomu okna scalania, wersje już scalone są wyświetlane w kolorze szarym.
- Podczas wyświetlania informacji adnotującej plik, możecie wybrać czy pokazać oryginalnego autora scalonych wersji, a nie osobę, która wykonała scalenie.
- Możecie zaznaczyć wersje jako *nie scalać*, umieszczając je na liście wersji scalonych bez rzeczywistego zrobienia scalenia.

Informacje śledzenia scalania są przechowywana w atrybucie `svn:mergeinfo` przez klienta podczas wykonywania scalania. Gdy scalanie zostało zatwierdzone serwer zapisuje te informacje w bazie danych, a gdy żądacie scalania, opisów zmian lub adnotacji, serwer może odpowiednio reagować. Aby system działał prawidłowo należy upewnić się, że serwer, repozytorium i wszystkie klienty są uaktualniane. Wcześniejszy klient nie będzie przechowywać atrybutu `svn:mergeinfo` a wcześniejsze serwery nie dostarczą informacji wymaganych przez nowe klienty.

O śledzeniu scaleń w Subversion dowiecie się więcej z *Dokumentacji scalenia połaczeń wersji* [http://svn.apache.org/repos/asf/Subversion/trunk/notes/merge-tracking/index.html].

4.21.6. Obsługa Konfliktów po Scaleniu

Ważne

Teksty w oknach rozwiązywania konfliktów zapewniane sa przez bibliotekę SVN i przez to nie mogły (jeszcze) zostać przetłumaczone jak okna TortoiseSVN. Przepraszamy.

Scalenie nie zawsze przebiega gładko. Czasami występuje konflikt. TortoiseSVN pomaga w tym procesie, pokazując okno dialogowe *scalenia konfliktu*.

Rysunek 4.58. Okno Scalenia Konfliktu

Jest prawdopodobne, że niektóre zmiany zostaną scalone płynnie, podczas gdy inne lokalne modyfikacje wywołają konflikty ze zmianami zatwierdzonymi już do repozytorium. Wszystkie zmiany, które mogą być scalone są scalane. Okno Scalenia Konfliktu daje różne sposoby obsługi linii, które pozostają w konflikcie.

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

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

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

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

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

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

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

Oznacz jako rozwiązany
: Oznacza konflikty jako rozwiązane. Ten przycisk jest wyłączony do czasu użycia przycisku Edytuj by edytować konflikt ręcznie i zapisać te zmiany do pliku. Gdy zmiany są zapisane, przycsk zostanie włączony.

Edytuj
: Uruchamia edytor scalenia by można było rozwiązać konflikty ręcznie. Nie zapomnijcie zapisać plik by przycisk Oznacz jako rozwiązany został włączony.

If there's a tree conflict, please first see Sekcja 4.6.3, „Konflikty drzewa” about the various types of tree conflicts and how and why they can happen.

By rozwiązać konflikt drzewa po scaleniu, wyświetlane jest okno z różnymi opcjami rozwiązania konfliktu:

Rysunek 4.59. The Merge Tree Conflict Dialog

Ponieważ jest wiele możliwości powstania konfliktu drzewa, w oknie pokazywane są przyciski rozwiązań zależnych od określonego konfliktu. Napisy i etykiety na przyciskach informują jakie opcje rozwiązania konfliktu prezentuje. Jeśli nie jesteście pewni, zamknijcie okno lub skorzystajcie z przycisku Odłóż by rozwiązać konflikt później.

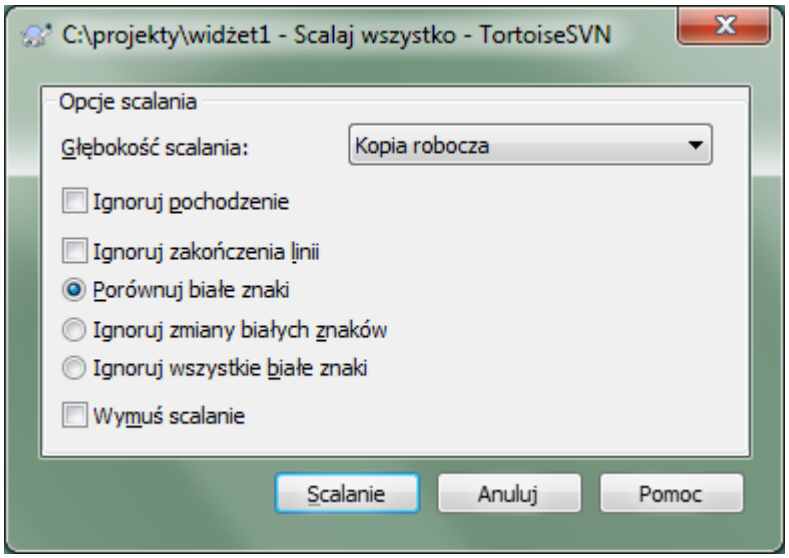
4.21.7. Utrzymanie gałęzi funkcji

Podczas opracowywania nowej funkcji przy użyciu odrębnej gałęzi, dobrym pomysłem jest, aby wypracować politykę reintegracji, gdy funkcja jest gotowa. Jeśli inne prace wykonywane są na `linii gównej` w tym samym czasie, może się okazać, że różnice stają się z czasem znaczące, a scalenie z powrotem staje się koszmarem.

Jeśli funkcja ta jest stosunkowo prosta i rozwój nie potrwa długo, możecie przyjąć proste podejście, polegające na trzymaniu gałęzi w separacji do zakończenia tej funkcji, a następnie scalić zmiany gałęzi z powrotem do linii głównej. W kreatorze scalania będzie to proste działanie Scal zakres wersji, z zakresem wersji jako rozpiętością wersji gałęzi.

Jeśli funkcja będzie dopracowywana dłużej i trzeba brać pod uwagę zmiany w `linii głównej`, musicie mieć gałąź synchronizowaną. Oznacza to,okresowe scalenia zmian linii głównej do gałęzi, tak, by gałąź zawierała wszystkie zmiany linii głównej *oraz* z nowej funkcji. Proces synchronizacji wykorzystuje Scal zakres wersji. Gdy funkcja jest zakończona, to możesz połączyć ją z powrotem do `linii głównej` za pomocą Reintegruj gałąź lub Scal dwa różne drzewa.

Inną (szybką) drogą by scalić wszystkie zmiany z linii głównej do gałęzi funkcji jest użycie TortoiseSVN → Scalaj wszystko... z rozszerzonego menu kontekstowego (należy przytrzymać klawisz **Shift** podczas prawokliku na pliku).



Rysunek 4.60. Okno dialogowe Scalaj-Wszystko

To okno jest naprawdę proste. Wszystko co trzeba zrobić to ustawienie opcji scalenia, jak to opisano w sekcji Sekcja 4.21.3, „Opcje scalania”. Reszta wykonywana jest przez TortoiseSVN automatycznie przy użyciu śledzenia scaleń.

4.22. Blokowanie

Subversion na ogół działa najlepiej bez blokowania, przy użyciu metody „kopia-modyfikacja-scalanie” opisanej wcześniej w Sekcja 2.2.3, „Rozwiązanie kopia-modyfikacja-scalanie”. Jest jednak kilka przypadków, kiedy może zaistnieć potrzeba wykonania niektórych postaci polityki blokad.

- Używacie „niescalalnych” plików, na przykład plików graficznych. Jeśli dwóch ludzi zmienia ten sam plik, scalenie nie jest możliwe, więc jedno z was straci swoje zmiany.
- Twoja firma zawsze stosowała system kontroli wersji z blokowaniem w przeszłości i zapadła decyzja kierownictwa, że „blokady są najlepsze”.

Po pierwsze należy się upewnić, że serwer Subversion został uaktualniony do wersji co najmniej 1.2. Wcześniejsze wersje nie obsługują w ogóle blokady. Jeśli używacie dostępu `file://`, to oczywiście tylko klient musi zostać zaktualizowany.

Trzy Znaczenia „Blokady”

W tej sekcji, jak również prawie wszędzie w tej książce, słowa „blokada” i „blokowanie” opisuje mechanizm wzajemnego wykluczania użytkowników by uniknąć niszczących się wzajemnie zatwierdzeń. Niestety występują dwa inne rodzaje „blokad”, którymi Subversion, a przez to i ta książka, musi czasem się zająć.

Drugim są `blokady kopii roboczej`, używane wewnętrznie przez Subversion do zapobiegania starciom pomiędzy wieloma klientami Subversion działającymi na tej samej kopii roboczej Często blokady takie powstają gdy polecenie jak update/commit/... zostanide przerwane błędem. Blokady te mogą być usunięte przez uruchomienie polecenia porządkowania kopii roboczej, co opisano w Sekcja 4.17, „Uporządkuj”.

I trzeci, pliki i foldery zostają zablokowane gdy korzysta z nich inny proces, na przykład gdy macie dokument otwarty właśnie w Wordzie, zostaje on zablokowany i nie można uzyskać do niego dostępu z TortoiseSVN.

Można z reguły zapomnieć o tych innych blokadach, dopóki coś się nie wyłoży i nie trzeba się będzie tym zająć. W tej książce „blokada” odnosi się do pierwszego znaczenia, chyba że wynika toz kontekstu lub jest bezpośrednio powiedziane.

4.22.1. Jak działa blokowanie w Subversion

Domyślnie nic nie jest zablokowane i każdy, kto ma dostęp do zatwierdzeń może dokonać zmian w dowolnym pliku w dowolnym momencie. Inni będą aktualizować swoje kopie robocze okresowo i zmiany w repozytorium zostaną scalone ze zmianami lokalnymi.

Jeśli *Uzyskasz blokadę* na pliku, to tylko Wy będziecie mogli wykonywać zatwierdzenia tego pliku. Zatwierdzenia wszystkich innych użytkowników będą blokowane do momentu zwolnienia blokady. Zablokowany plik nie może być modyfikowany w jakikolwiek sposób w repozytorium, więc nie można go usunąć ani zmienić jego nazwy, chyba że uczyni to właściciel blokady.

Ważne

Blokada jest przypisana nie do konkretnego użytkownika, ale do konkretnego użytkownika i kopii roboczej. Blokada na jednej kopii roboczej uniemożliwia również temu samemu użytkownikowi zatwierdzenie zablokowanego pliku z innej kopii roboczej.

Na przykład, wyobraźcie sobie, że użytkownik Jan ma kopię roboczą na swoim komputerze biurowym. Tam rozpoczyna pracę na obrazie, a zatem uzyskuje blokadę na tym pliku. Gdy opuszcza swoje biuro, jeszcze nie zakończył pracy z tym plikiem, więc nie zwalnia blokady na nim. Po powrocie do domu Jan tam także ma kopię roboczą i decyduje się popracować trochę nad projektem. Ale nie można zmienić ani zatwierdzić tego pliku obrazu, ponieważ blokada tego pliku znajduje się na jego kopii roboczej w biurze.

Jednak inni użytkownicy nie muszą wiedzieć, że nałożyliście blokadę. O ile nie sprawdzają stanu blokad regularnie, pierwszym razem, gdy się o takiej dowiedzą, będzie brak możliwości zatwierdzenia, co w większości przypadków nie jest zbyt przydatne. Aby ułatwić zarządzanie blokadami, wprowadzono nowy atrybut Subversion `svn:needs-lock`. Gdy ten atrybut jest ustawiony (na jakąkolwiek wartość) na pliku, gdy plik jest pobierany

lub aktualizowany, lokalna kopia ma ustawianą właściwość tylko do odczytu *chyba*, że kopia robocza utrzymuje blokadę na pliku. Mechanizm ten działa jako ostrzeżenie, że nie należy zmieniać tego pliku o ile wcześniej nie ustawiło się blokady. Pliki wersjonowane i tylko do odczytu są oznaczone specjalną nakładką w TortoiseSVN, aby wskazać, że trzeba uzyskać blokadę przed ich edycją.

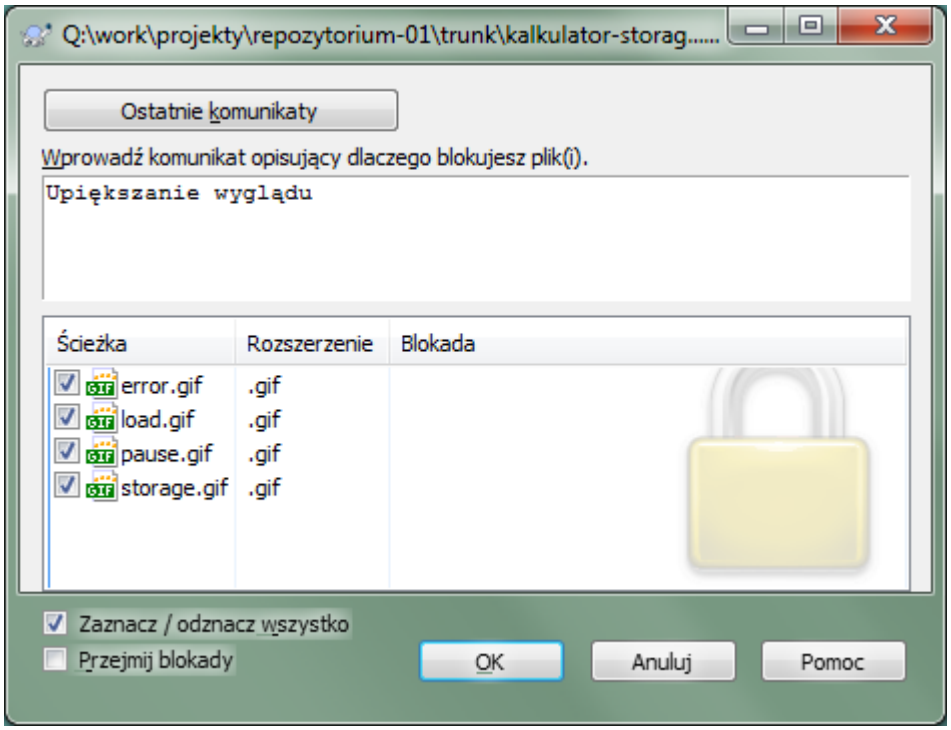
Blokady są zapamiętywane dla konkretnej lokalizacji kopii roboczej tak samo jak właściciela. Jeśli macie kilka kopii roboczych (w domu, w pracy), to można posiadać blokadę tylko w *jednej* z tych kopii roboczych.

Jeśli jeden z Waszych współpracowników nakłada blokadę i potem idzie na urlop, nie zwalniając jej, co można zrobić? Subversion zapewnia środki do wyłamywania blokad. Zwolnienie blokady przez kogoś innego nazywane jest dalej *Łamaniem* blokady, zaś nałożenie siłą własnej blokady, którą trzyma już ktoś inny jest określany jako *Przejęcie* blokady. Oczywiście nie są to rzeczy, które należy robić delikatnie, jeśli chce się pozostać w przyjaźni ze współpracownikami.

Blokady są rejestrowane w repozytorium, zaś token blokady jest tworzony w lokalnej kopii roboczej. Jeśli istnieje rozbieżność, na przykład, jeśli ktoś złamał blokadę, lokalny token blokady staje się nieważny. Repozytorium jest zawsze ostatecznym odniesieniem.

4.22.2. Ustawianie blokady

Wybierzcie w kopii roboczej plik(i), dla którego chcecie uzyskać blokadę, a następnie wybierzcie polecenie TortoiseSVN → Zablokuj....



Rysunek 4.61. Okno dialogowe blokady

Pojawi się okno dialogowe, pozwalające na wprowadzenie komentarza, tak aby inni mogli zrozumieć, dlaczego zablokowaliście plik. Komentarz jest opcjonalny i obecnie używany tylko w repozytoriach opartych na svnserve. Jeśli (i *tylko*, jeśli) musicie przejąć blokadę od kogoś innego, zaznaczcie pole Przejmij blokadę, a następnie kliknijcie OK.

Można ustawić atrybut projektu `tsvn:logtemplatelock`, aby zapewnić szablon opisu zmian dla użytkowników wypełnić go jako komunikat blokady. Zapoznajcie się z Sekcja 4.18, „Ustawienia projektu” by uzyskać instrukcje dotyczące ustawiania atrybutów.

Jeśli wybierzecie folder, a następnie użyjecie TortoiseSVN → Zablokuj… okno blokady otworzy się dla *każdego* pliku w *każdym* podkatalogu wybranym do zablokowania. Jeśli naprawdę chcecie zablokować całą hierarchię,

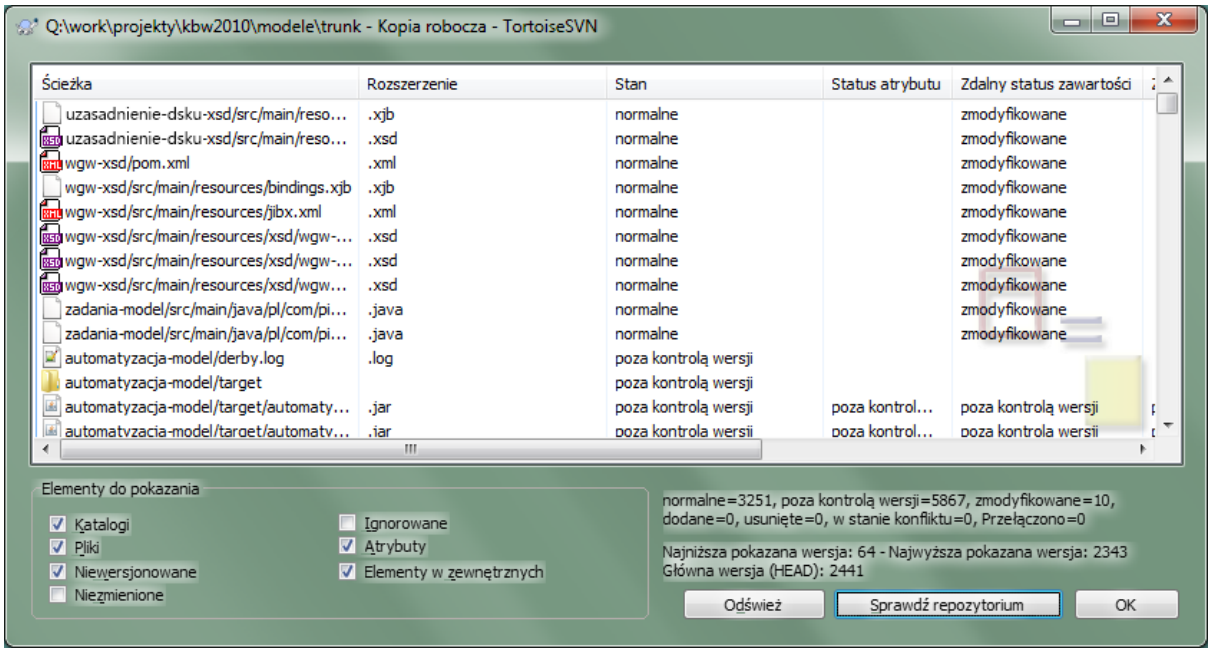
w ten sposób to zrobicie, ale można stać się bardzo niepopularnym wśród swoich współpracowników, jeśli zablokujecie im cały projekt. Używajcie z rozwagą ...

4.22.3. Zwalnianie blokady

Aby upewnić się, że nie zapomnicie zwolnić zbędnej blokady, zablokowane pliki są wyświetlane w oknie dialogowym zatwierdzeń jako domyślnie zaznaczone. Jeśli zatwierdzenie zostanie wykonane, blokady trzymane na wybranych plikach zostaną usunięte, nawet jeśli pliki nie zostały zmodyfikowane. Jeśli nie chcecie, aby zwolnić blokadę na pewne pliki, możecie odznaczyć je (jeśli nie są one modyfikowane). Jeśli chcecie zachować blokadę na pliku, który został zmodyfikowany, musicie zaznaczyć pole wyboru Trzymaj blokady przed zatwierdzeniem zmian.

Aby zwolnić blokadę ręcznie, wybierzcie plik(i) w kopii roboczej, dla którego chcecie, aby zwolnić blokadę, a następnie wybierzcie polecenie TortoiseSVN → Zwolnij blokadę... Nie trzeba nic więcej wprowadzać zanim TortoiseSVN połączy się z repozytorium i zwolni blokadę. Możecie również użyć tego polecenia na folderze, aby zwolnić wszystkie blokady rekurencyjnie.

4.22.4. Sprawdzenie stanu blokady



Rysunek 4.62. Okno dialogowe Sprawdź zmiany

Aby zobaczyć blokady trzymane przez Was i innych, można użyć TortoiseSVN → Sprawdź zmiany…. Lokalnie przechowywane tokeny blokady pokazują się natychmiast. Aby sprawdzić blokadę utrzymywaną przez innych (i zobaczyć czy któraś z waszych blokad jest złamana lub przejęta), trzeba kliknąć na Sprawdź repozytorium.

Z menu kontekstowego tutaj można również uzyskać i zwolnić blokadę, a także złamanie i przejąć blokady utrzymywane przez innych.

Unikanie łamania i przejmowania blokad

W przypadku złamania lub przejęcia cudzej blokady bez powiadomienia właściciela, można potencjalnie spowodować utratę czyjejś pracy. Jeśli pracujecie na niescalalnych typach plików i przejmujecie cudze blokady, po zwolnieniu blokady mogą oni swobodnie zatwierdzić własne zmiany i nadpisać Wasze. Subversion nie traci danych, ale Wy tracicie ochronę pracy zespołowej, którą daje blokada.

4.22.5. Oznaczanie nieblokowanych plików jako tylko do odczytu

Jak wspomniano powyżej, najbardziej efektywnym sposobem na używania blokad jest ustawianie atrybutu `svn:needs-lock` na plikach. Zapoznajcie się z Sekcja 4.18, „Ustawienia projektu” by uzyskać instrukcje dotyczące ustawiania atrybutów. Pliki z tym atrybutem zawsze będą pobierane i uaktualniane z właściwością tylko do odczytu, chyba, że kopia robocza trzyma blokadę.

Przypominamy, TortoiseSVN używa w tym celu specjalnej nakładki na ikonę.

Jeśli prowadzicie polityki, w których każdy plik musi być zablokowany, może się okazać, że łatwiej używać funkcji auto-props Subversion, by ustawić atrybut automatycznie przy każdym dodaniu nowych plików. Czytajcie Sekcja 4.18.1.5, „Automatyczne ustawienie atrybutu” dla uzyskania dalszych informacji.

4.22.6. Skrypty przechwytujące blokowania

Po utworzeniu nowego repozytorium przez Subversion 1.2 lub nowsze, cztery szablony przechwyceń są tworzone w folderze `hooks` repozytorium. Nazwane są od czasu wykonania: przed i po uzyskaniu blokady (before lock, after lock) oraz przed i po zwolnieniu blokady (before unlock, after unlock).

Jest to dobry pomysł, aby zainstalować na serwerze skrypty przechwytujące `post-lock` i `post-unlock`, wysyłające e-mail wskazujący plik, który został zablokowany. Przy pomocy działania takiego skryptu, wszyscy użytkownicy zostaną powiadomieni, jeśli ktoś zablokuje/odblokowuje plik. Możeszcie znaleźć przykład skryptu przechwytującego `hooks/post-lock.tmpl` w folderze repozytorium.

Możecie również użyć przechwyceń, aby uniemożliwić złamanie lub przejęcie blokad, czy może ograniczyć je do mianowanych administratorów. A może chcecie wysłać email do właściciela, kiedy jedna z ich blokad zostanie złamana lub przechwycona.

Czytajcie Sekcja 3.3, „Skrypty przechwytujące po stronie serwera” by dowiedzieć się więcej.

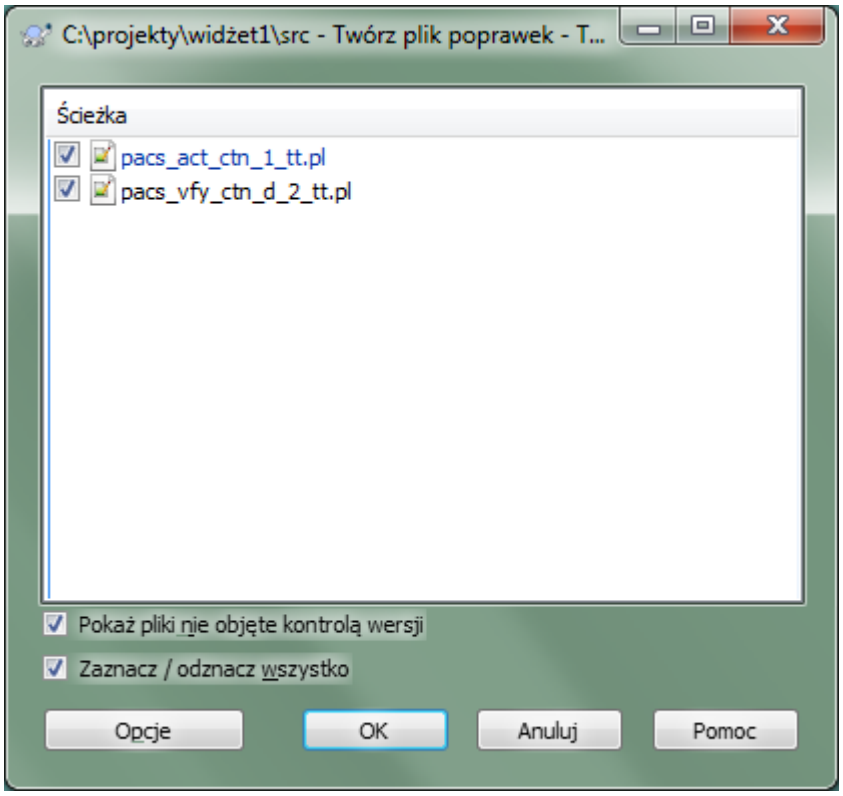
4.23. Tworzenie i stosowanie poprawek

Dla projektów open source (jak ten) każdy ma dostęp do odczytu repozytorium, i każdy może przyczynić się do rozwoju projektu. Jak zatem kontrolować te udziały? Jeśli każdy może zatwierdzić zmiany, projekt zostanie trwale zdestabilizowany i prawdopodobnie na stałe uszkodzony. W takiej sytuacji zmiany są zarządzane przez złożenie pliku *poprawki* do zespołu programistów, którzy mają dostęp do zapisu. Mogą oni najpierw przejrzeć poprawki, a potem albo dodać go do repozytorium lub odrzucić z powrotem do autora.

Pliki poprawki są po prostu plikami różnicowymi w formacie Unified-Diff pokazującymi różnice między kopią roboczej a wersją bazy.

4.23.1. Tworzenie pliku poprawki

Najpierw musicie wprowadzić *i przetestować* zmiany. Wtedy zamiast TortoiseSVN → Zatwierdź... na folderze nadrzędnym należy wybrać TortoiseSVN → Twórz plik poprawek...



Rysunek 4.63. Okno dialogowe tworzenia pliku poprawek

można teraz wybrać pliki, które chcecie zawrzeć w poprawce, tak samo jak przy pełnym zatwierdzeniu. Utworzy to jeden plik zawierający zestawienie wszystkich dokonanych zmian do wybranych plików od ostatniej aktualizacji z repozytorium.

Kolumny w tym oknie można dostosować w ten sam sposób, jak kolumny w dialogu Sprawdź zmiany. Czytajcie Sekcja 4.7.3, „Status lokalny i zdalny” dla dalszych szczegółów.

Klikając na przycisk Opcje można określić, jak poprawka jest tworzona. Na przykład można określić, że zmiany w zakończeń linii lub białych znaków nie są włączone do końcowej poprawki.

Można stworzyć oddzielne poprawki zawierające zmiany dla różnych zestawów plików. Oczywiście, jeśli tworzycie plik poprawki, następnie dodajecie kolejne zmiany w *tym samym* pliku, po czym utworzycie nową poprawkę, druga poprawka będzie zawierała *oba* zestawy zmian.

Wystarczy zapisać plik do używając wybranej przez siebie nazwy. Pliki poprawek mogą mieć dowolne rozszerzenia, ale zwyczajowo powinny korzystać z rozszerzenia `.patch` lub `.diff`. Teraz jesteśmy gotowi do złożenia pliku poprawki.

Podpowiedź

nie zapisujcie pliku poprawki z rozszerzeniem `.txt` jeśli chcecie wysłać go pocztą elektroniczną do kogoś innego. Pliki tekstowe są często zniekształcane przez programy pocztowe i często zdarza się że białe znaki i znaki nowej linii są automatycznie przekształcane i pakowane. Gdy tak się dzieje, poprawka nie zostanie prawidłowo zastosowana. Należy użyć `.patch` lub `.diff` jako rozszerzenia podczas zapisywania pliku poprawki.

Można również zapisać poprawkę do schowka zamiast do pliku. Jest to wygodne, gdyż można treść wkleić do e-maila aby inni ją ocenili. Albo, jeśli macie dwie kopie robocze na jednym komputerze i chcecie przenieść zmiany z jednej do drugiej, poprawka w schowku stanowi wygodny sposób na zrobienie tego.

Jeśli wolicie, możecie utworzyć plik poprawki z okien dialogowych Zatwierdź zmiany oraz Sprawdź zmiany. Wystarczy wybrać pliki i skorzystać z menu kontekstowego by stworzyć poprawkę z tych plików. Jeśli chcecie zobaczyć okno Opcje musicie przytrzymać **shift** podczas kliknięcia prawym przyciskiem myszy.

4.23.2. Stosowanie pliku poprawki

Pliki poprawek są stosowane do kopii roboczej. Należy to zrobić na tym samym poziomie folderu, jaki został wykorzystany do stworzenia patcha. Jeśli nie jesteście pewni, gdzie on jest, wystarczy spojrzeć na pierwszą linię pliku patch. Na przykład, jeśli pierwszy przetworzony plik miał ścieżkę `doc/source/english/chapter1.xml` i pierwsza linia w pliku poprawki brzmi `Index: english/chapter1.xml` to trzeba zastosować poprawkę do folderu `doc/source/`. Jednak pod warunkiem, że znajdujecie się w poprawnej kopii roboczej, jeśli wybierzecie niewłaściwy poziom folderu, TortoiseSVN wyświetli uwagę i zaproponuje odpowiedni poziom.

Aby zastosować plik poprawki do kopii roboczej, trzeba mieć co najmniej dostęp do odczytu repozytorium. Powodem tego jest fakt, że program musi połączyć odniesienia zmian z powrotem do wersji na których zostały one wykonane przez zdalnego dewelopera.

Z menu kontekstowego na tym folderze kliknijcie na TortoiseSVN → Zastosuj poprawkę... Spowoduje to wyświetlenie okna dialogowego otwarcia pliku umożliwiającego wybranie pliku poprawki do zastosowania. Domyślnie tylko pliki z rozszerzeniami `.patch` lub `.diff` są wyświetlane, ale można wybrać „Wszystkie pliki”. Jeśli wcześniej zapisaliście patch do schowka, można użyć Otwórz ze schowka... w oknie otwarcia pliku. Warto podkreślić, że ta opcja pojawia się tylko wtedy wprowadzono poprawkę do schowka przy użyciu TortoiseSVN → Twórz plik poprawek.... Skopiowanie poprawki do schowka z innej aplikacji nie spowoduje wyświetlania przycisku.

Alternatywnie, jeśli patch ma rozszerzenie `.patch` lub `.diff`, można kliknąć prawym przyciskiem myszy na nim bezpośrednio i wybrać TortoiseSVN → Zastosuj poprawkę.... W tym przypadku pojawi się monit o podanie lokalizacji kopii roboczej.

Te dwie metody tylko wykonują to samo na dwa sposoby. Używając pierwszej metody wybiera się KR i przechodzi do pliku poprawek. Przy drugiej wybiera się plik poprawki i przechodzi do KR.

Po wybraniu pliku poprawki i lokalizacji kopii roboczej, TortoiseMerge wykonuje próbę scalenia zmian z pliku poprawki z kopią roboczą. Małe okno wyświetla listę plików, które zostały zmienione. Kliknijcie dwukrotnie na każdym z kolei, by przejrzeć zmiany i zapisać scalone pliki.

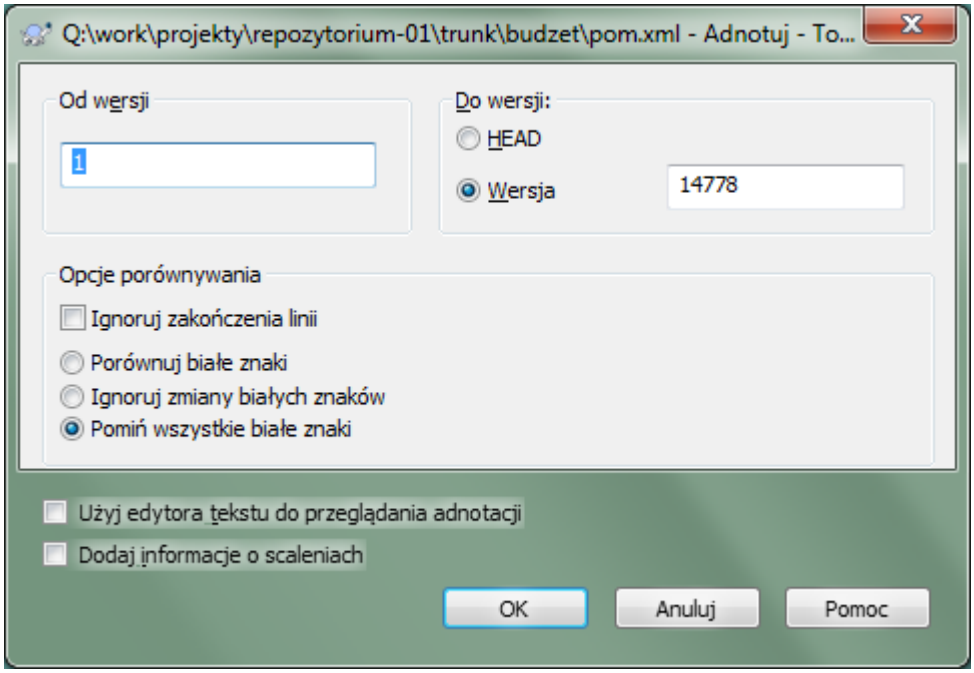
Poprawka zdalnego dewelopera została zastosowana do kopii roboczej, więc trzeba ją zatwierdzić, aby wszyscy uzyskali dostęp do zmian z repozytorium.

4.24. Kto zmienił którą linię?

Czasem trzeba wiedzieć nie tylko, jak się zmieniły linie, ale także kto dokładnie zmienił konkretne linie w pliku. To wtedy przydaje się polecenie TortoiseSVN → Adnotuj..., czasem zwane również *komentuj*.

To polecenie wypisuje dla każdej linii w pliku, autora i wersję w której linia została zmieniona.

4.24.1. Adnotacje dla plików



Rysunek 4.64. Okno dialogowe komentarza / adnotacji

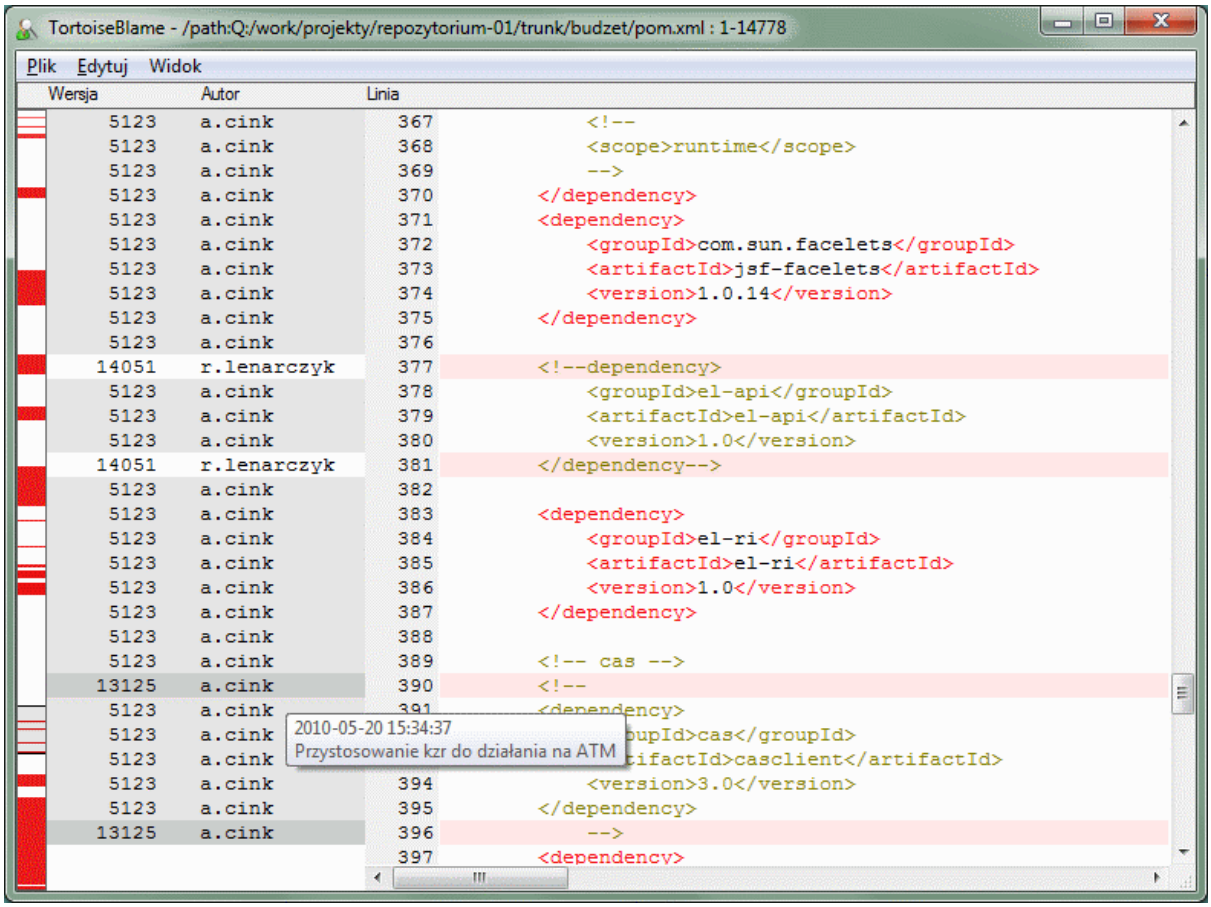
Jeśli nie jesteście zainteresowani zmianami z wcześniejszych wersji można ustawić wersję, od której adnotacje powinny się zaczynać. Ustawcie ją na `1`, jeśli chcecie adnotacji dla *wszystkich* wersji.

Domyślnie plik adnotacji jest wyświetlany przy użyciu z*TortoiseBlame*, który podświetla różne wersje, aby ułatwić ich czytanie. Jeżeli chcecie wydrukować lub edytować plik adnotacji, wybierzcie Użyj edytora tekstu do przeglądania adnotacji.

Możecie określić sposób, w jaki są obsługiwane znaki końca linii i białe znaki. Opcje te zostały opisane w Sekcja 4.11.2, „Opcje końca linii i białych znaków”. Domyślnym zachowaniem jest traktowanie wszystkich różnic białych znaków i końców linii jako rzeczywiste zmiany, ale jeśli chcecie zignorować zmiany wcięcia i znaleźć właściwego autora kodu, można wybrać tutaj odpowiednią opcję.

Można dołączyć również informacje o scaleniach jeśli chcesz, jednak opcja ta może potrwać znacznie dłużej, ze względu na pobieranie danych z serwera. Kiedy linie są scalone z innego źródła, informacje adnotacji pokazują wersję w której zmiana została wprowadzona w oryginalnym źródle, jak i wersję, kiedy zostały scalone do tego pliku.

Gdy wciśnięto OK TortoiseSVN rozpoczyna pobieranie danych potrzebnych do utworzenia pliku adnotacji. Po zakończeniu procesu adnotacji wynik jest zapisywany w pliku tymczasowym i można przejrzeć wyniki.



Rysunek 4.65. TortoiseBlame

TortoiseBlame, który jest częścią TortoiseSVN, sprawia, że plik adnotacji łatwiejszy do odczytania. Po najechaniu kursorem na linii w kolumnie informacji adnotacji, wszystkie linie z tej samej wersji są pokazane na ciemniejszym tle. Wiersze z innych zmian, które zostały zmienione przez samego autora są pokazane na jasnym tle. Kolorystyka może nie działać tak wyraźnie, jeśli macie ustawione wyświetlanie na trybie 256 kolorów.

Jeśli się kliknie lewym przyciskiem myszy na linii, wszystkie linie o takiej samej wersji zostaną podświetlone, a linie z innych wersjami tego samego autora są podświetlone na o ton jaśniej. Wyróżnienie jest przylegające, co pozwala na ruch myszy bez utraty podświetleń. Kliknijcie na tej wersji raz jeszcze, aby wyłączyć podświetlenie.

Komentarze wersji (opisy zmian z dziennika) są wyświetlane w oknie podpowiedzi gdy wskaźnik myszy przesuwa się nad kolumnę informacji o adnotacji. Jeśli chcecie, aby skopiować opis zmiany dla tej wersji, wystarczy użyć menu kontekstowego, które pojawia się po kliknięciu prawym przyciskiem myszy na kolumnie informacji adnotacji.

Możecie szukać w raporcie adnotacji za pomocą Edytuj → Znajdź.... To pozwala na wyszukiwanie numerów wersji, autorów i samej zawartości pliku. Opisy zmian nie są dołączone do wyszukiwania - należy użyć okna dziennika by w nich wyszukiwać.

Możecie również przejść do określonego numeru linii za pomocą Edytuj → Idź do linii….

Kiedy mysz jest nad kolumnami informacji adnotacji, dostępne jest menu kontekstowe, które pomaga w porównywaniu wersji i badaniu historii, za pomocą numeru wersji z linii pod myszą jako odniesieniem. Menu kontekstowe → Adnotuj poprzednią wersję generuje raport adnotacji dla tego samego pliku, ale z poprzednią wersją jako górną granicą. Daje to raport adnotacji dla stanu plik tuż zanim przeglądana linia wyglądała jak we wskazanej wersji. menu kontekstowe → Pokaż zmiany uruchamia przeglądarkę różnic, pokazując, co się

zmieniło we wskazywanej wersji. menu kontekstowe → Pokaż dziennik wyświetla okno dziennika zmian począwszy od wskazanej wersji.

Jeśli potrzebujecie lepszego wizualnego wskaźnika gdzie znajdują się najstarsze i najnowsze zmiany, wybierzcie Widok → Koloruj linie wg wieku. Zostanie użyty gradient koloru, aby pokazać nowsze linie w kolorze czerwonym i starsze linie w kolorze niebieskim. Kolorystyka domyślnie jest dość jasna, ale można to zmienić go za pomocą ustawień TortoiseBlame.

Jeśli używacie Śledzenia Scaleń i żądacie informacji o scaleniach podczas uruchamiania adnotacji, linie scalone są wyświetlane nieco inaczej. Jeżeli linia zmieniła się w wyniku scalenia z inną ścieżką, TortoiseBlame pokaże wersję i autora ostatniej zmiany w oryginalnym pliku, a nie wersję, gdzie scalenie miało miejsce. Linie te są wyróżnione, pokazując wersję i autora kursywą. Wersja scalenia jest pokazywana oddzielnie w dymku po najechaniu kursorem myszy na kolumnach informacji adnotacji. Jeśli nie chcecie by scalone linie pokazane były w ten sposób, odznaczcie pole wyboru Dodaj informacje o scaleniach przy uruchamianiu adnotacji.

Jeśli chcecie zobaczyć ścieżki składowe scalenia, wybierzcie Widok → Scalanie ścieżek. Pokazana zostanie ścieżka gdzie linia została ostatnio zmieniona, z wyłączeniem zmian wynikających ze scalenia.

Wersja przedstawiona w informacji adnotacji reprezentuje ostatnią wersję, gdy zawartość tej linii została zmieniona. Jeśli plik został utworzony przez skopiowanie innego pliku, cofamy się do momentu zmiany samej linii i adnotacja pokaże ostatnią wersję w oryginalnym pliku źródłowym, a nie wersję, w której wykonano kopię. Dotyczy to również ścieżek pokazanych w informacji scalenia. Ścieżka pokazuje położenie w repozytorium, w którym ostatnia zmiana została wprowadzona do tej linii.

Ustawienia TortoiseBlame można uzyskać za pomocą TortoiseSVN → Ustawienia... na karcie TortoiseBlame. Zapoznaj się z Sekcja 4.31.9, „Ustawienia TortoiseBlame”.

4.24.2. Różnice adnotacji

Jednym z ograniczeń raportu adnotacji jest to, że pokazuje tylko, jak plik wyglądał w szczególnej wersji i ostatnią osobę, która zmieniła każdą linię. Czasami chcesz wiedzieć, jaka zmiana została wprowadzona, a także, kto ją wykonał. Po kliknięciu prawym przyciskiem na linii w TortoiseBlame otrzymujecie wejście menu kontekstowego, do pokazania zmian w tej wersji. Ale jeśli chcecie zobaczyć zmiany *i* informacje adnotacji jednocześnie potrzebujecie połączenia raportów porównania i adnotacji.

Okno dziennika wersji zawiera kilka opcji pozwalających na wykonanie tego.

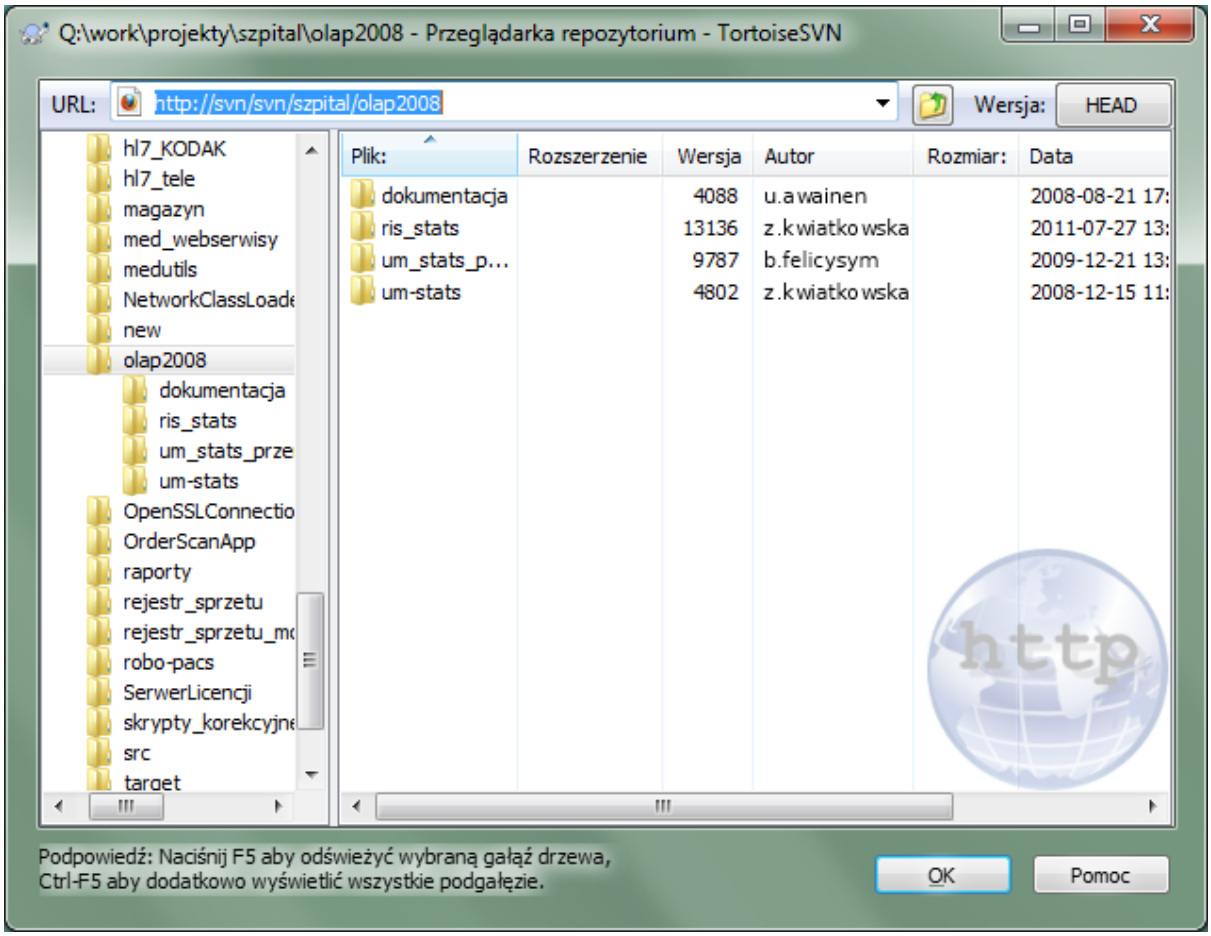
Adnotuj wersje
: W górnym okienku wybierzcie 2 wersje, a następnie wybierzcie menu kontekstowe → Adnotuj wersje. W ten sposób pobrane zostaną dane adnotacji dla 2 wersji, a następnie użyjcie porównywarki by zestawić dwa pliki adnotacji.

Adnotuj zmiany
: Wybierzcie jedną wersję w górnym okienku, a następnie wskażcie jeden plik w dolnym okienku i wybierzcie → Adnotuj zmiany. W ten sposób pobrane zostaną dane adnotacji dla wybranej i poprzedniej wersji, a następnie użyjcie porównywarki by zestawić dwa pliki adnotacji.

Porównaj i adnotuj z roboczą BASE
: Pokażcie dziennik dla pojedynczego pliku, i w górnym okienku wybierz jedną wersję, następnie wybierzcie menu kontekstowe → Porównaj i adnotuj z roboczą BASE. W ten sposób pobrane zostaną dane adnotacji dla wybranej wersji oraz pliku roboczego BASE, po czym użyjcie porównywarki dla skonfrontowania dwóch plików adnotacji.

4.25. Przeglądarka repozytorium

Czasem trzeba pracować bezpośrednio naw repozytorium, bez konieczności posiadania kopii roboczej. Do tego właśnie służy *Przeglądarka repozytorium*. Tak jak eksplorator i nakładki ikon pozwalają na przeglądanie kopii roboczej, tak przeglądarka repozytorium pozwala na przeglądanie struktury i stanu repozytorium.



Rysunek 4.66. Przeglądarka repozytorium

Z przeglądarki repozytorium można wykonać polecenia, takie jak kopiowanie, przenoszenie, zmiana nazwy ... bezpośrednio w repozytorium.

Przeglądarka repozytorium wygląda bardzo podobnie do eksploratora Windows, oprócz tego, że pokazuje zawartość repozytorium dla szczególnej wersji, a nie pliki na komputerze. W lewym okienku można zobaczyć drzewo katalogów, a w prawym wyświetlana jest zawartość wybranego katalogu. W górnej części okna przeglądarki repozytorium możecie wpisać adres URL repozytorium, i wersję, które chcecie przeglądać.

Foldery dołączone z atrybutem `svn:externals` są również przedstawione w przeglądarce repozytorium. Te foldery są wyświetlane z małą strzałką na nich by pokazać, że nie są one częścią struktury repozytorium, a odwołaniami.

Podobnie jak w eksploratorze Windows, można kliknąć na nagłówki kolumn w prawym okienku, jeśli trzeba zmienić porządek sortowania. I tak jak w eksploratorze jest menu kontekstowe dostępne w obu okienkach.

Menu kontekstowe dla pliku pozwala:

- Otwiera wybrany plik, albo przy użyciu domyślnej dla tego typu pliku przeglądarki, albo w wybranym przez Was programie.
- Edycja wybranego pliku. Spowoduje pobranie tymczasowej kopii roboczej i uruchomi edytor odpowiedni dla tego typu pliku. Po zamknięciu programu edytora, jeśli zmiany zostały zapisane, pojawi się okno dialogowe zatwierdzenia, pozwalające na wprowadzenie komentarza i zatwierdzenie zmiany.
- Pokazuje dziennik zmian dla tego pliku, lub pokazuje wykres wszystkich wersji, dzięki czemu można zobaczyć, skąd plik pochodzi.
- Adnotuje plik, aby zobaczyć kto i kiedy zmienił którą linię.

- Pobiera jeden plik. Stwarza to „rzadką” kopię roboczą, która zawiera tylko ten jeden plik.
- Usuwa lub zmienia nazwę pliku.
- Zapisuje niewersjonowaną kopię pliku na dysku twardym komputera.
- Kopiuje adres URL pokazany na pasku adresu do schowka.
- Wykonuje kopię pliku, albo do innej części repozytorium, albo do kopii roboczej wykonanej z tego samego repozytorium.
- Przegląda/edytuje atrybuty pliku.
- Tworzy skrót by można było szybko ponownie uruchomić przeglądarkę repozytorium dla bieżącej lokalizacji.

Menu kontekstowe dla folderu pozwala:

- Pokaż dziennik zmian dla tego folderu, lub pokazuje wykres wszystkich wersji, dzięki czemu można zobaczyć, skąd folder pochodzi.
- Eksportuje folder do lokalnej niewersjonowanej kopii na dysku twardym.
- Pobiera folder by utworzyć lokalną kopię roboczą na dysku twardym.
- Tworzy nowy folder w repozytorium.
- Dodaje niewersjonowane pliki lub foldery bezpośrednio do repozytorium. Jest to faktycznie operacja Import Subversion.
- Usuwa lub zmienia nazwę folderu.
- Tworzy kopię folderu albo w innej części repozytorium albo do kopii roboczej wykonanej z tego samego repozytorium. Może to być wykorzystane do utworzenia gałęzi/etykiety bez potrzeby wykonywania pobrania do kopii roboczej.
- Przegląda/edytuje atrybuty folderu.
- Zaznacza folder do porównania. Zaznaczony folder jest wyświetlany pogrubioną czcionką.
- Porównuje folder z wcześniej zaznaczonym, albo jako plik różnicowy, albo jako lista zmienionych plików z których każdy może być później porównany przy użyciu domyślnej porównywarki. Może to być szczególnie użyteczne podczas porównywania dwóch etykiet lub linii głównej z gałęzią by określić co zostało zmienione.

Po wybraniu dwóch folderów w prawym okienku, można zobaczyć różnice jako plik różnicowy, lub listę plików, które mogą być wizualnie porównane przy użyciu domyślnej porównywarki.

Po wybraniu wielu folderów w prawym okienku, można pobrać wszystkie z nich naraz do dowolnego folderu nadrzędnego.

Jeśli wybierzecie 2 etykiety, które są skopiowane z tego samego źródła (zazwyczaj `/trunk/`), można użyć Menu kontekstowe → Pokaż dziennik..., aby wyświetlić listę wersji pomiędzy dwoma punktami etykiet.

Zewnętrzne elementy (z odwołaniami za pomocą `svn:externals` są również pokazane w przeglądarce repozytorium, a można nawet przejść do zawartości folderu. Elementy zewnętrzne są oznaczone czerwoną strzałką nad elementem.

Można użyć **F5** aby jak zwykle odświeżyć widok. Spowoduje to odświeżenie wszystkiego, co jest aktualnie wyświetlane. Jeśli chcecie załadować z wyprzedzeniem lub odświeżyć informacje dla węzłów, które nie zostały jeszcze otwarte, użyjcie **Ctrl-F5**. Po tym, rozszerzenie dowolnego węzła nastąpi natychmiast, bez opóźnień ładowania informacji z sieci.

Można również użyć w przeglądarce repozytorium operacji przeciągnij i upuść. Jeśli przeciągniecie folder z eksploratora do przeglądarki repo, zostanie on zaimportowany do repozytorium. Zauważ, że jeśli przeciągniecie wiele elementów, będą mogły być zaimportowane w oddzielnych zatwierdzeniach.

Jeśli chcecie przenieść element w repozytorium, wystarczy go przeciągnąć lewym przyciskiem myszy do nowej lokalizacji. Jeśli chcecie utworzyć kopię zamiast przenosić element, stosuje się **Ctrl**-przeciągnięcie lewym przyciskiem myszy. Podczas kopiowania, kursor ma nadpisany symbol „plus”, tak samo jak w eksploratorze.

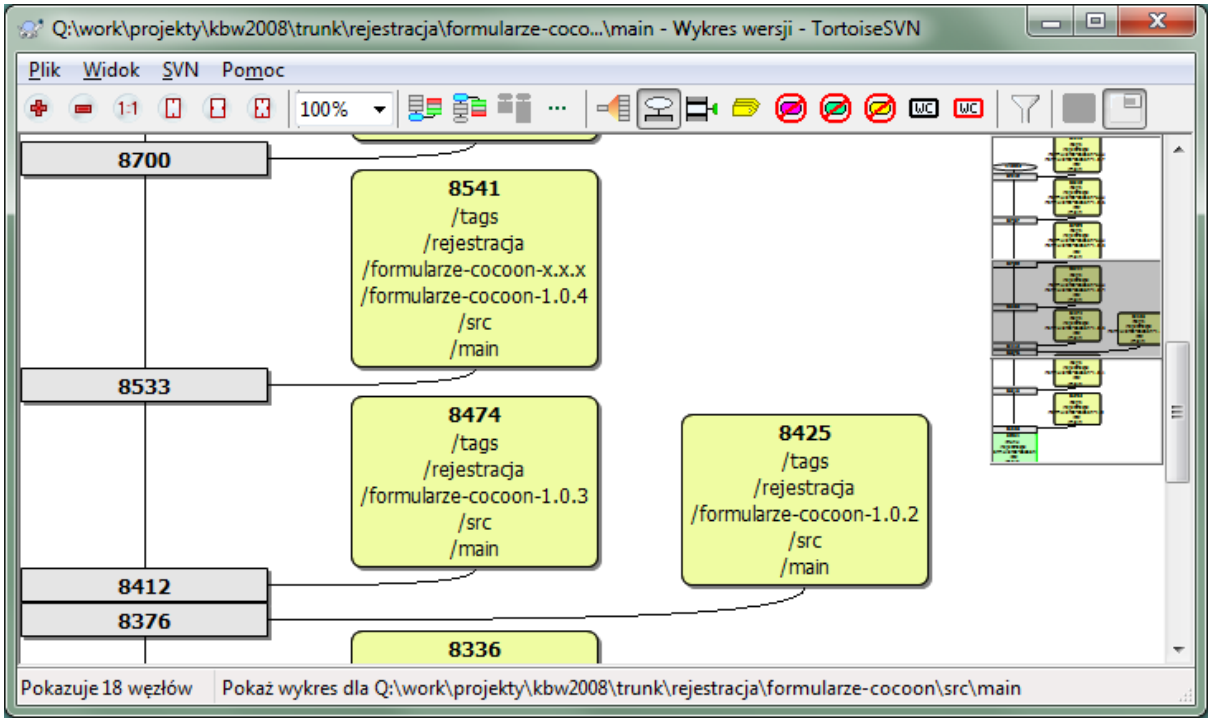
Jeśli chcecie skopiować/przenieść plik lub folder do innej lokalizacji, a także nadać mu nową nazwę w tym samym czasie, można go przeciągnąć prawym przyciskiem myszy lub **Ctrl**-przeciągnąć prawym przyciskiem myszy zamiast przeciągać lewym przyciskiem myszy. W takim przypadku jest wyświetlane okno zmiany nazwy, w którym można wpisać nową nazwę pliku lub folderu.

W przypadku wprowadzenia zmian w repozytorium za pomocą jednej z tych metod, zostanie wyświetlone okno dialogowe wprowadzenia opisu zmiany. Jeśli przeciągniecie coś przez pomyłkę, jest to również okazja, aby anulować akcję.

Czasami podczas próby otwarcia ścieżki otrzymacie komunikat o błędzie w miejsce szczegółów elementu. To może się zdarzyć, jeśli określono nieprawidłowy adres URL, nie macie uprawnień dostępu, lub jeśli jest jakiś inny problem z serwerem. Gdy chcecie skopiować wiadomość, aby zawrzeć ją w e-mailu, po prostu kliknijcie prawym przyciskiem myszy i użyjcie Menu kontekstowe → Kopiuj komunikaty błędów do schowka, lub po prostu użyj **Ctrl+C**.

URLe/repozytoria oznaczone zakładkami są pokazywane poniżej folderów bieżącego repozytorium w widoku drzewa z lewej strony. Można dodać tu elementy klikając prawym klawiszem myszy na dowolnym pliku i wybierając Menu kontekstowe → Dodaj do Zakładek. Kliknięcie na zakładce otworzy podgląd do tego repozytorium oraz pliku/folderu.

4.26. Wykresy wersji



Rysunek 4.67. Wykres wersji

Czasami trzeba wiedzieć, gdzie gałęzie i etykiety zostały pobrane z linii głównej a idealnym sposobem by obejrzeć tego typu informacje jest postać wykresu lub drzewa. Należy wtedy użyć TortoiseSVN → Pokaż wykres wersji...

To polecenie analizuje historię zmian i próbuje stworzyć drzewo pokazujące miejsca, w których zostały pobrane kopie, i kiedy gałęzie/etykiety zostały usunięte.

Ważne

W celu wygenerowania wykresu, TortoiseSVN musi pobrać wszystkie opisy zmian z głównego repozytorium. Oczywiście może to trwać wiele minut, nawet z repozytorium zawierającego kilka tysięcy wersji, w zależności od szybkości serwera, przepustowość sieci, itd. Jak wypróbujecie to na czymś w rodzaju projektu *Apache*, który ma obecnie ponad 500.000 wersji, można sobie poczekać czas jakiś.

Dobrą wiadomością jest to, że jeżeli używacie buforowania dziennika, tylko cierpieć tą zwłokę to raz. Po tym, dane dziennika odbywa się lokalnie. Buforowanie dziennika jest włączone w ustawieniach TortoiseSVN.

4.26.1. Węzły wykresu wersji

Każdy węzeł wykresu przedstawia zmiany wersję w repozytorium, gdzie coś się zmieniło w oglądanym drzewie. Różne rodzaje węzłów można odróżnić na podstawie kształtu i koloru. Kształty są stałe, a kolory można ustawić za pomocą TortoiseSVN → Ustawienia

Elementy dodane i skopiowane
: Elementy, które zostały dodane lub utworzone przez skopiowanie innego pliku/folderu wyświetlane są przy użyciu zaokrąglonego prostokąta. Domyślnym kolorem jest zielony. Etykiety i linie główne są traktowane jako szczególny przypadek i używa się innego koloru, w zależności od TortoiseSVN → Ustawienia.

Elementy usunięte
: Elementy usunięte np. gałąź, która nie jest już wymagana, są wyświetlane przy użyciu ośmiokąt (prostokąt z obciętymi narożnikami). Domyślny jest kolor czerwony.

Elementy ze zmienioną nazwą
: Elementy ze zmienioną nazwą są również wyświetlone przy użyciu ośmiokąta, ale domyślnym kolorem jest niebieski.

Wersja wskazówki gałęzi
: Wykres jest zwykle ograniczony do pokazywania punktów gałęzi, ale często przydaje się możliwość obejrzenia odpowiednich wersji HEAD również dla każdej gałęzi. Jeśli wybierzesz Pokaż wersje HEAD, każdy węzeł wersji HEAD pojawi się jako elipsa. Należy pamiętać, że HEAD odnosi się tutaj do ostatniej wersji zatwierdzonej na tej ścieżce, nie zaś wersji HEAD repozytorium.

Wersja kopii roboczej
: Jeśli wywołuje się wykres wersji z kopii roboczej, można wybrać wyświetlanie wersji BASE na wykresie używając Pokaż wersję kopii roboczej, co zaznacza węzeł BASE pogrubionym konturem.

Zmieniona kopia robocza
: Jeśli wywołuje się na wykres wersji z kopii roboczej, można wybrać, aby pokazać dodatkowy węzeł reprezentujący zmodyfikowaną kopię roboczą za pomocą Pokaż modyfikacje kopii roboczej. Jest to eliptyczny węzeł z pogrubionym domyślnie czerwonym konturem.

Zwykły element
: Wszystkie inne elementy pokazywane są przy użyciu zwykłego prostokąta.

Zauważcie, że domyślnie wykres pokazuje tylko punkty, w których elementy zostały dodane, skopiowane lub usunięte. Pokazanie wszystkich zmian projektu generuje bardzo duży wykres dla przypadków nietrywialnych. Jeśli naprawdę chcecie zobaczyć *wszystkie* wersje, gdzie dokonano zmian, jest możliwość wykonania tego w menu Widok i na pasku narzędzi.

Widok domyślny (wyłączone grupowanie) rozmieszcza węzły tak, aby ich pozycje pionowe były w ścisłym porządku wersji, jest zatem wizualna wskazówka odnośnie kolejności, w jakiej wykonano rzeczy. W przypadku, gdy dwa węzły są w tej samej kolumnie porządek jest oczywisty. Kiedy dwa węzły znajdują się w sąsiadujących kolumnach przesunięcie jest o wiele mniejsze, ponieważ nie ma potrzeby, aby zapobiec nakładaniu się węzłów, stąd kolejność jest tu nieco mniej oczywista. Takie optymalizacje są niezbędne do utrzymania złożonych wykresów do

rozsądnych rozmiarach. Należy pamiętać, że porządkowanie używa *krawędzi* węzła na *starszej* stronie jako punkt odniesienia, czyli dolnej krawędzi węzła gdy wykres jest pokazany od najstarszych węzłów na dole. Krawędź odniesienia jest istotna, ponieważ kształty węzłów nie są tej samej wysokości.

4.26.2. Zmiana widoku

Ponieważ wykres wersji jest często dość skomplikowany, istnieje kilka funkcji, które mogą być wykorzystywane do dostosowania widoku wedle upodobań. Są one dostępne w menu Widok i pasku narzędzi.

Grupowanie gałęziami
: Domyślne zachowanie (wyłączone grupowanie) ma wszystkie wiersze posortowane ściśle według wersji. W rezultacie, długowieczne gałęzie z nielicznymi zatwierdzeniami zajmują całą kolumnę dla tylko kilku zmian i wykres staje się bardzo szeroki.

  Ten tryb grupuje zmiany według gałęzi, tak że nie ma globalnego przeglądu wersji: Kolejne wersje na gałęzi wyświetlane są (często) kolejnych liniach. Gałęzie podrzędne jednak są ułożone w taki sposób, że późniejsze gałęzie pojawiają się w tej samej kolumnie powyżej wcześniejszych gałęzi dla utrzymania niewielkiej szerokości wykresu. W rezultacie, dany wiersz może zawierać zmiany z różnych wersji.

Najstarsze na górze
: Zwykle wykres przedstawia najstarszą wersję na dole, a drzewo rośnie w górę. Użyj tej opcji do zmiany kierunku wzrostu na z góry w dół.

Wyrównaj drzewa na górze
: Gdy wykres jest podzielony na kilka mniejszych drzew, drzewa mogą pojawić się zarówno w naturalnym porządku wersji, lub wyrównane do dolnej części okna, w zależności od tego, czy używacie opcji Grupowanie gałęziami. Użyjcie tej opcji do wskazania wzrostu wszystkich drzew od góry.

Redukuj krzyżowanie
: Ta opcja jest zazwyczaj włączona i unika pokazuje wykres z wieloma komplikującymi widok przecięciami linii. Jednak może to również powodować że kolumny ustawiają się w mniej logicznych miejscach, na przykład w ukośną linię, a nie w kolumnie, a wykres może zajmować większą powierzchnię. Jeśli jest to problem, możecie wyłączyć tę opcję z menu Widok.

Zaznacz zmiany w ścieżkach
: Długie nazwy ścieżek mogą zająć dużo miejsca i sprawiają, że prostokąt węzła jest bardzo duży. Użyjcie tej opcji, aby pokazać tylko zmienione części ścieżki, zastępując część wspólną kropkami. Na przykład jeśli tworzycie gałąź `/branches/1.2.x/doc/html` z `/trunk/doc/html` to gałąź mogłaby być przedstawiona w zwartej formie `/branches/1.2.x/..`, gdyż ostatnie dwa poziomy, `doc` i `html` nie zmieniły się.

Pokaż wszystkie wersje
: Robi dokładnie to, czego oczekujecie i pokazuje każdą wersję, w której coś (w rozrysowywanym drzewie) uległo zmianie. Dla długiej historii może to dać naprawdę ogromny wykres.

Pokaż wersje HEAD
: Gwarantuje to, że najnowsza wersja dla każdej gałęzi jest zawsze widoczna na wykresie.

Dokładne źródła kopii
: Kiedy jest tworzona gałąź/etykieta, domyślnym zachowaniem jest pokazanie gałęzi jako pobranej z ostatniego węzła, w którym dokonano zmiany. Ściśle mówiąc jest to niewłaściwe, gdyż gałęzie są wykonane najczęściej z obecnej wersji HEAD, a nie określonej wersji. Jest zatem możliwe, aby pokazać bardziej poprawną (ale mniej przydatną) wersję, która została wykorzystana do utworzenia kopii. Zauważcie, że ta wersja może być młodsza od wersji HEAD źródła gałęzi.

Zwiń etykiety
: Jeżeli projekt ma wiele etykiet, pokazując każdą z nich jako oddzielny węzeł na wykresie zajmuje dużo miejsca i ukrywa bardziej interesujące struktury gałęzi rozwojowych. W tym samym czasie być może trzeba mieć łatwy dostęp do treści etykiety, dzięki czemu można porównać wersje. Ta opcja ukrywa węzły etykiet a pokazuje je za to w dymku dla węzła, z którego zostały skopiowane. Ikona etykiety z prawej strony z węzła źródłowego wskazuje, że zostały wykonane etykiety. To znacznie upraszcza widok.

Zauważcie, że jeśli etykieta jest sama używany jako źródło kopii, być może nowej gałęzi opartej na etykiecie, to etykieta pojawi się jako osobny węzeł, a nie zwinięta.

Ukryj usunięte ścieżki
: Ukrywa ścieżki, które już nie występują w wersji HEAD repozytorium, np. usunięte gałęzie.

  Jeśli wybraliście opcję Zwiń etykiety, wtedy usunięte gałęzie, z których zostały wzięte etykiety nadal będą wyświetlane, w przeciwnym razie etykiety również znikną. Ostatnia wersja, która została zaetykietowana pojawi się w kolorze używanym dla usuniętych węzłów zamiast pokazywać oddzielną wersję usunięcia.

  Jeśli wybierzesz opcję Ukryj etykiety to te gałęzie ponownie znikną, ponieważ nie są potrzebne do pokazania etykiet.

Ukryj nieużywane gałęzie
: Ukrywa gałęzie w przypadku braku zmian zatwierdzonych dla odpowiedniego pliku lub podkatalogu. To nie musi oznaczać, że gałąź nie była używana, tylko, że nie dokonano żadnych zmian na *tej* jej części.

Pokaż wersję kopii roboczej
: Oznaczenia wersji na wykresie, która odpowiada wersji aktualizującej element, dla którego pobrano wykres. Jeśli właśnie nastąpiła aktualizacja, będzie to HEAD, ale jeśli inni zatwierdzili zmiany od waszej ostatniej aktualizacji KR może być kilka wersji niżej. Węzeł jest oznaczony, przez nadanie mu pogrubionego konturu.

Pokaż modyfikacje kopii roboczej
: Jeśli KR zawiera lokalne zmiany, opcja ta zwraca go jako oddzielny węzeł eliptyczny, powiązany do węzła z którego KR otrzymała ostatnią aktualizację. Domyślnym kolorem konturu jest czerwony. Być może trzeba odświeżyć wykres używając **F5** by uchwycić ostatnie zmiany.

Filtr
: Czasami wykres wersji zawiera więcej wersji niż chcesz zobaczyć. Opcja ta otwiera okno dialogowe, które pozwala na ograniczenie wyświetlanego zakresu wersji i ukrycie poszczególnych ścieżek według nazwy.

  Po ukryciu szczególnej ścieżki, której węzeł ma węzły potomne, dzieci pojawią się jako oddzielne drzewo. Jeśli chcecie, aby ukryć wszystkie dzieci, użyjcie pola wyboru Usuń całe poddrzewo(-a).

Paski drzewa
: W przypadku, gdy wykres zawiera kilka drzew, czasem jest przydatne skorzystanie z alternatywnych kolorów tła, aby pomóc rozróżnić drzewa.

Pokaż przegląd
: Wyświetla mały obrazek całego wykresu, z bieżącym oknie widoku jako prostokątem, który można przeciągać. Pozwala to na poruszanie się łatwiej po wykresie. Należy pamiętać, że dla wykresów bardzo dużego grafu przegląd może stać się bezużyteczny ze względu na ekstremalny współczynnik powiększenia i dlatego nie będzie wyświetlany w takich przypadkach.

4.26.3. Wykorzystanie wykresu

Aby łatwiej poruszać się po dużym wykresie, należy użyć okna przeglądu. Pokazuje ono cały wykres w małym okienku z podświetloną aktualnie wyświetlaną częścią. Możecie przeciągać zaznaczony obszar by zmienić wyświetlany region.

Data aktualizacji, autor i komentarze wyświetlane są w oknie podpowiedzi gdy wskaźnika myszy przesunie się nad polem zmian.

Jeśli wybierzecie dwie zmiany (Użyjcie **Ctrl**-kliknięcia lewym przyciskiem myszy), można użyć menu kontekstowego, aby pokazać różnice pomiędzy tymi wersjami. Możecie wybrać, wyświetlanie różnic w punktach utworzenia gałęzi, ale zwykle będziecie chcieli pokazać różnice w punktach końcowych gałęzi, tj. w wersji HEAD.

Możecie zobaczyć różnice pliku różnicowym Unified-Diff, który pokazuje wszystkie różnice w pojedynczym pliku, przy minimalnym kontekście. Jeśli zdecydujecie się na Menu kontekstowe → Porównaj wersje zostanie wyświetlona lista zmienionych plików. Kliknięcie dwukrotne na nazwie pliku pozwala załadować obie wersje pliku i porównać je za pomocą narzędzia do wizualnego porównywania.

Jeśli klikniesz prawym przyciskiem myszy na wersji można użyć Menu kontekstowe → Pokaż dziennik, aby wyświetlić historię.

Można również scalić zmiany w wybranej wersji(-ach) do innej kopii roboczej. Okno wyboru folderu pozwala wybrać kopię roboczą do scalenia, ale po tym nie ma okna dialogowego potwierdzenia, ani okazji przetestowania scalenia. Jest to dobry pomysł, aby scalić do niezmodyfikowanej kopii roboczej, dzięki czemu można cofnąć zmiany, jeśli nie wyjdzie! Jest to przydatna funkcja, jeśli chcesz scalić wybrane wersje z jednej gałęzi do drugiej.

Nauka czytania wykresu wersji

Początkujący użytkownicy mogą być zaskoczeni faktem, że wykres wersji pokazuje coś, co nie pasuje do modelu myślowego użytkownika. Jeśli wersja zmienia wiele kopii lub gałęzi pliku lub folderu, na przykład, niech będzie wiele węzłów dla pojedynczej wersji. Jest to dobra praktyka, aby rozpocząć od lewej opcji na pasku narzędzi i dostosować wykres krok po kroku, aż przybliży się do modelu myślowego.

Wszystkie opcje filtrowania usiłują stracić tak mało informacji, jak to możliwe. Może to powodować że niektóre węzły zmienią kolor, na przykład. Gdy wynik jest niepoprawny, wycofajcie ostatnią operację filtrowania i spróbujcie zrozumieć, co jest specjalnego w danej wersji lub gałęzi. W większości przypadków początkowo oczekiwany wynik operacji filtrowania będzie niedokładny lub mylący.

4.26.4. Odświeżanie widoku

Jeśli chcecie sprawdzić ponownie serwer by wczytać nowe informacje, możesz po prostu odświeżyć widok przy użyciu **F5**. Jeśli używacie bufora dziennika (domyślnie włączony), zostanie sprawdzone repozytorium odnośnie nowszych zatwierdzeń i pobierane są tylko nowe. Jeśli bufor dziennika był w trybie offline, to nastąpi również próba powrotu do trybu online.

Jeśli używasz bufora dziennika i uważasz, że treści wiadomości lub autor może ulec zmianie, należy użyć okna dziennika, aby odświeżyć potrzebne wiadomości. Ponieważ wykres wersji działa z korzenia repozytorium, trzeba by unieważnić całą zawartość bufora dziennika a uzupełnianie może potrwać *bardzo* długo.

4.26.5. Przycinanie drzew

Duże drzewo może być trudne w nawigacji i czasami chcesz ukryć jego części, lub rozbić go na las mniejszych drzew. Jeśli po najechaniu myszką na punkt, w którym węzeł łącza wchodzi lub opuszcza węzeł zostanie wyświetlony jeden lub więcej rozwijalnych przycisków, które pozwalają zrobienie tego.

Kliknijcie na minus, aby zwinąć dołączone poddrzewa.

Kliknijcie na przycisk plus aby rozwinąć zwinięte drzewo. Kiedy drzewo jest zwinięte, ten przycisk jest widoczny by wskazać ukryte poddrzewa.

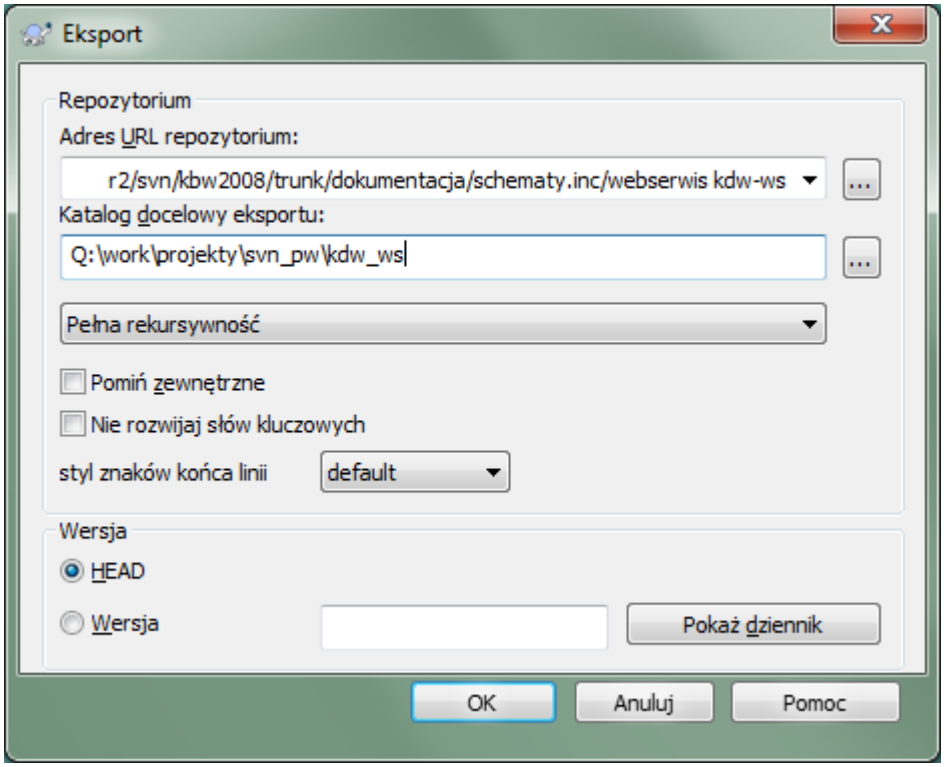
Kliknijcie na przycisku z krzyżykiem, aby podzielić dołączone poddrzewa i pokazać je jako oddzielne drzewa na wykresie.

Kliknijcie przycisk koła aby ponownie połączyć podzielone drzewa. Gdy drzewo zostało podzielone, ten przycisk jest widoczny, aby wskazać, że istnieje osobne poddrzewo.

Kliknijcie na tle wykresu by wywołać główne menu kontekstowe, które oferuje opcje Rozwiń wszystko i Połącz wszystko. Jeśli żadna gałąź nie została zwinięta lub rozdzielona, menu kontekstowe nie pojawi się.

4.27. Eksport kopii roboczej Subversion

Niekiedy możecie życzyć sobie czystej kopii drzaewa roboczego bez folderu `.svn`, np. by utwozyć zzipowane archiwum lub wysłać na serwer sieciowy. Zamiast tworzenia kopię i usuwania z niej ręcznie folderu `.svn`, TortoiseSVN oferuje polecenie TortoiseSVN → Eksport.... Eksportowanie z URL i eksportowanie z kopii roboczejtraktowane są trochę inaczej.



Rysunek 4.68. Okno dialogowe Eksport-z-URL

Podczas wykonania tego polecenia w folderze bez informacji o wersji, TortoiseSVN zakłada, że w wybrany folder jest celem, i otwiera okno na wpisanie adresu URL i wersji jako źródła eksportu. To okno posiada opcje eksportu tylko katalogu najwyższego poziomu, pominięcia odnośników zewnętrznych i zastąpienia stylu końca linii dla plików, które mają ustawiony atrybut `svn:eol-style`.

Oczywiście można eksportować również bezpośrednio z repozytorium. Otwórzcie przeglądarkę repozytorium, aby przejść do odpowiedniego poddrzewa w repozytorium, a następnie użyjcie Menu kontekstowe → Eksport. Otrzymacie okno Export z adresu URL opisane powyżej.

Jeśli wykonujecie to polecenie na kopii roboczej, zostaniecie zapytani o miejsce zapisu *czystej* kopii roboczej bez folderu `.svn`. Domyślnie eksportowane są tylko pliki wersjonowane, ale możecie użyć pola wyboru Eksportuj także pliki nie objęte kontrolą wersji by dodać wszystkie niewersjonowane pliki istniejące w KR ale nie przechowywane w repozyrorium. Wskazania przy użyciu `svn:externals` na zewnętrzne mogą być w razie potrzeby pominięte.

Innym sposobem eksportu z kopii roboczej jest przeniesienie prawym przyciskiem folderu kopii roboczej na inne miejsce i wybranie Menu Kontekstowe → SVN Eksportuj wersjonowane pliki tutaj, Menu Kontekstowe

→ SVN Eksportuj wszystko tutaj lub Menu Kontekstowe → SVN Eksportuj zmienione elementy tutaj. Ta druga opcja kopiuje również niewersjonowane pliki. Trzecia opcja eksportuje tylko zmienione elementy, pozostawiając strukturę folderów.

Przy eksporcie z kopii roboczej, jeśli folder docelowy zawiera już folder o takiej samej nazwie jak ten, z którego są eksportowane, będziecie mieć możliwość zastąpienia istniejącej zawartości lub utworzenia nowego folderu z automatycznie wygenerowaną nazwą, np. `Docelowy (1)`.

Eksportowanie pojedynczego pliku

Okno dialogowe eksportu nie pozwala na eksport pojedynczych plików, choć Subversion jest w stanie.

Aby eksportować pojedyncze pliki z TortoiseSVN, musicie użyć przeglądarki repozytorium (Sekcja 4.25, „Przeglądarka repozytorium”). Wystarczy przeciągnąć plik(i), który chcesz wyeksportować z przeglądarki repozytorium tam, gdzie chcecie je w eksploratorze, lub skorzystać z menu kontekstowego w przeglądarce repozytorium, aby wyeksportować pliki.

Eksportowanie drzewa zmian

Jeśli chcecie wyeksportować kopię struktury drzewa projektu, ale zawierającą tylko pliki, które uległy zmianie w szczególnej wersji, lub pomiędzy dwiema wersjami, użyjcie funkcji porównania wersji opisanej w Sekcja 4.11.3, „Porównanie folderów”.

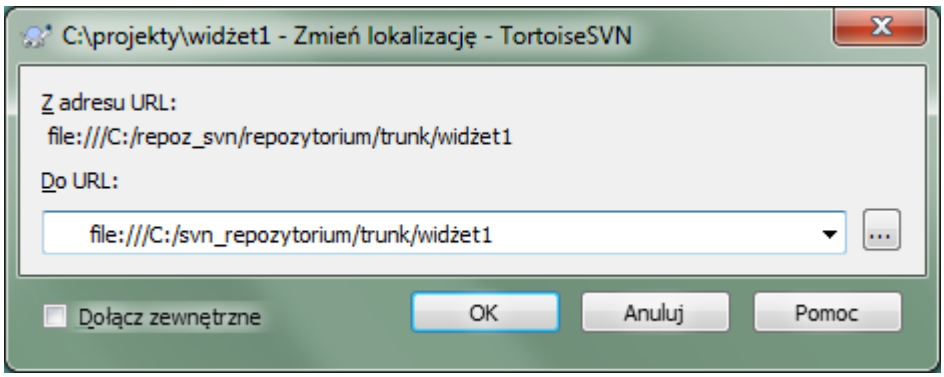
Jeśli chcecie wyeksportować strukturę drzewa zawierającą tylko pliki zmienione lokalnie, przyjrzyjcie się SVN Eksportuj zmienione elementy tutaj opisane powyżej.

4.27.1. Usunięcie kopii roboczej z kontroli wersji

Czasem macie kopię roboczą którą potrzeba zmienić z powrotem w zwykły folder bez katalogu `.svn`. Wszystko co trzeba zrobić, to usunąć katalog `.svn` z korzenia kopii roboczej.

Innąopcją jest eksport folderu do siebie samego. W Eksploratorze Windows przenieście prawym przyciskiem myszy folder główny z panelu plików do siebie samego w panelu folderów. TortoiseSVN wykrywa ten specjalny przypadek i pyta czy życzycie sobie odłączyć wersjonowanie od kopii roboczej. Po odpowiedzi *tak*, folder kontrolny zostanie usunięty i otrzymujecie zwykłe,niewersjonowane drzewo katalogów.

4.28. Relokacja kopii roboczej



Rysunek 4.69. Okno dialogowe relokacji

Jeśli repozytorium ma z jakiegoś powodu zmienić swoje położenie (IP/URL). Być może utknęliście i nie możecie zatwierdzić i nie chcecie pobrać kopii roboczej ponownie z nowej lokalizacji by przenosić wszystkie zmienione dane z powrotem do nowej kopii roboczej, TortoiseSVN → Zmień lokalizację jest poleceniem, którego szukacie.

W zasadzie robi ono bardzo niewiele: przepisuje wszystkie adresy URL, które są związane z każdym plikiem i folderem z nowego adresu URL.

> **Uwaga**
>
> Operacja ta działa tylko na *korzeniu* kopii roboczej. Stąd wejście w menu kontekstowym pojawia się tylko dla korzenia swojej kopii.

Możecie być zaskoczeni, że TortoiseSVN łączy się z repozytorium w ramach tej operacji. Całość pracy to wykonanie kilku prostych sprawdzeń, aby upewnić się, że nowy adres URL naprawdę odnosi się do tego samego repozytorium, co istniejąca kopia robocza.

> **Ostrzeżenie**
>
> *Jest to bardzo rzadko wykonywana operacja*. Polecenie relokacji jest używane *tylko*, jeśli zmienił się adres URL katalogu głównego repozytorium. Możliwe przyczyny to:
>
> - Zmienił się adres IP serwera.
> - Zmienił się protokół (np.. z http:// do https://).
> - Zmieniła się ścieżka do korzenia repozytorium w konfiguracji serwera.
>
> Innymi słowy, trzeba relokować, gdy kopia robocza odwołuje się do tego samego miejsca w tym samym repozytorium, ale samo repozytorium zostało przeniesione.
>
> To nie ma zastosowania, jeżeli:
>
> - Chcecie przenieść się do innego repozytorium Subversion. W takim wypadku należy wykonać czyste pobranie z nowej lokalizacji repozytorium.
> - Aby przełączyć się na inną gałąź lub katalog w tym samym repozytorium, należy użyć TortoiseSVN → Przełącz.... Czytajcie Sekcja 4.20.3, „Pobierać czy przełączać...” by uzyskać więcej informacji.
>
> Jeśli korzystacie z relokacji w jednym z powyższych przypadków, to *uszkodzi kopię roboczą* i otrzymacie wiele niewyjaśnionych komunikatów o błędach podczas aktualizacji, zatwierdzania itp. Kiedy to się już stało, jedynym ratunkiem jest nowe pobranie.

4.29. Integracja z systemami śledzenia błędów / śledzenia problemów

Bardzo często w rozwoju oprogramowania zmiany powinny być związane z konkretnym błędem lub ID problemu. Użytkownicy systemów śledzenia błędów (trackerów problemów) chcieliby powiązać zmiany wykonane w Subversion z określonym ID w ich systemie śledzenia błędów. Większość trackerów przewiduje zatem dołączenie skryptu przechwytującego pre-commit, który przetwarza opisy zmian by znaleźć ID błędu, z którym wiąże się zatwierdzenie. Jest to nieco podatne na błędy, ponieważ opiera się na poprawnym wprowadzeniu opisu zmiany przez użytkownika, by skrypt pre-commit mógł przetworzyć go poprawnie.

TortoiseSVN wspomaga użytkownika na dwa sposoby:

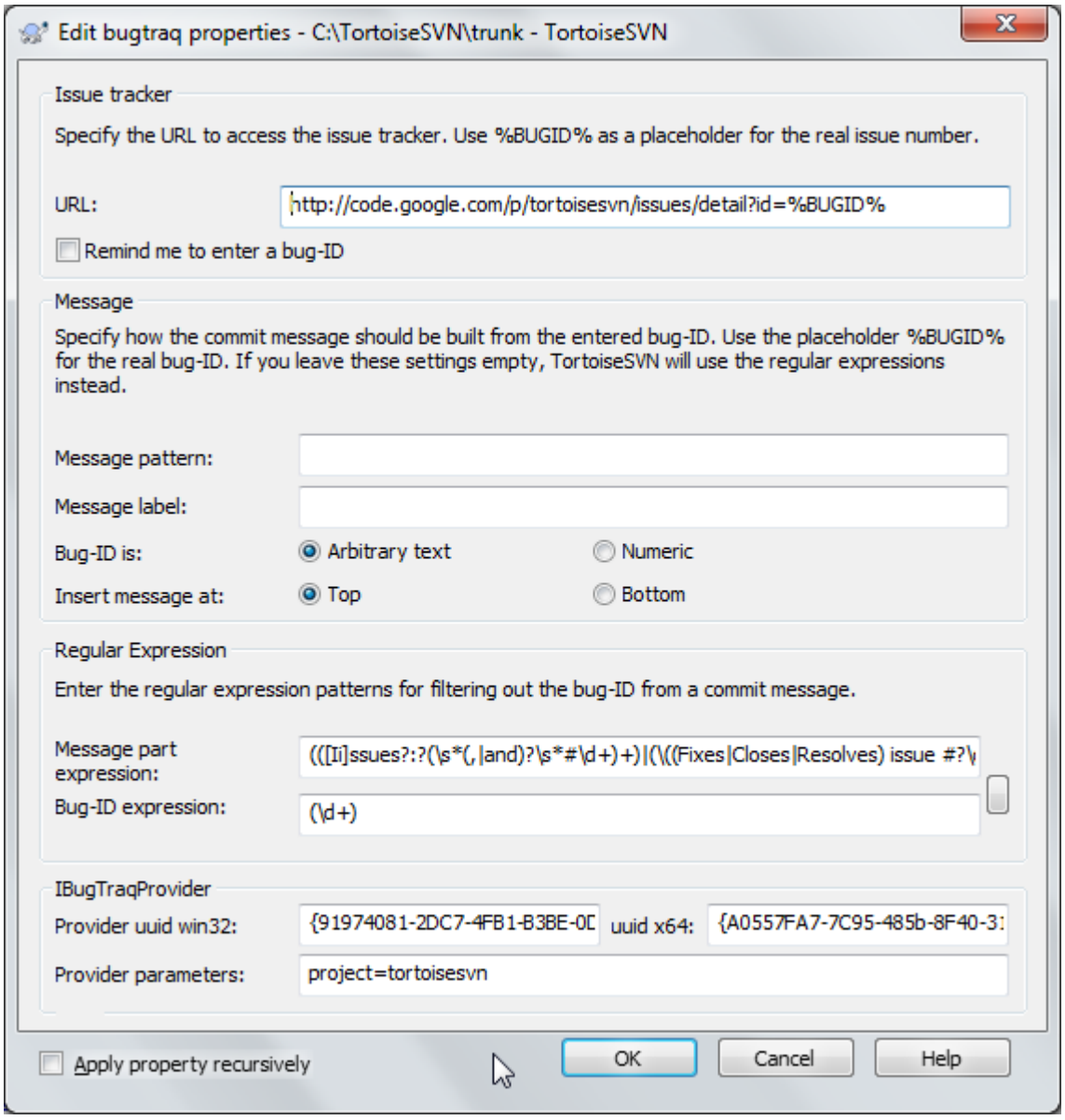
1. Gdy użytkownik wprowadzi opis zmiany, dobrze zdefiniowana linia, zawierająca numer wydania związanego z zatwierdzeniem może zostać dodana automatycznie. Zmniejsza to ryzyko, że użytkownik wpisze numer wydania w taki sposób, że narzędzie śledzenia błędów nie będzie w stanie poprawnie go przeanalizować.

   Lub TortoiseSVN może wyróżnić część wpisanego opisu zmiany, który jest uznawany przez tracker problemów. W ten sposób użytkownik wie, że komunikat w dzienniku może być przetworzony prawidłowo.

2. Gdy użytkownik przegląda opisy zmian, TortoiseSVN tworzy połączenie z każdego ID błędu w opisie zmiany, który uruchamia przeglądarkę do wskazanego błędu.

4.29.1. Dodawanie numerów wydań do opisów zmian

Możecie zintegrować wybrane narzędzie śledzenia błędów w TortoiseSVN. Aby to zrobić, musicie zdefiniować kilka właściwości, które zaczynają się `bugtraq:`. Muszą one być ustawione na folderach: (Sekcja 4.18, „Ustawienia projektu”)



Rysunek 4.70. Okno dialogowe atrybutów bugtraq

kiedy edytujecie atrybuty bugtraq używany jest specjalny edytor atrybutu w celu ułatwienia ustawienia odpowiednich wartości.

Istnieją dwa sposoby integracji TortoiseSVN z trackerami problemów. Jednym z nich jest oparty na prostych ciągach znaków, drugi jest oparty na *wyrażeniach regularnych*. Atrybuty używane przez oba przypadki są:

bugtraq:url
: W atrybucie tym należy ustawić adres URL narzędzia śledzenia błędów. Należy prawidłowo enkodowanym URI i zawierać `%BUGID%`. `%BUGID%` jest zastępowany numerem, który został wpisany. Pozwala to TortoiseSVN wyświetlić link w oknie dziennika, więc jeśli szukacie w dzienniku wersji można przejść bezpośrednio do narzędzia śledzenia błędów. Nie musicie podawać tego atrybutu, ale TortoiseSVN pokaże tylko numer wydania, a nie link do niego. Np. projekt TortoiseSVN używa `http://issues.tortoisesvn.net/?do=details&id=%BUGID%`.

Można również używać względnych adresów URL zamiast bezwzględnych. Jest to przydatne, gdy tracker problemów jest w tej samej domenie/serwerze jak repozytorium kodu źródłowego. W przypadku, gdy nazwa domeny się nie zmienia, nie musicie dostosować atrybutu `bugtraq:url`. Istnieją dwa sposoby na określenie względnego adresu URL:

Jeśli zaczyna się od ciągu `^/`, zakłada się, że w jest liczony względem głównego repozytorium. Na przykład, `^/../?do=details&id=%BUGID%` zostanie rozpisany do `http://tortoisesvn.net/?do=details&id=%BUGID%`, jeśli repozytorium znajduje się na `http://tortoisesvn.net/svn/trunk/`.

Adres URL rozpoczynający się od ciągu `/` przekłada się na adres względem do hosta serwera. Na przykład `/?do=details&id=%BUGID%` zostanie rozpisany do`http://tortoisesvn.net/?do=details&id=%BUGID%`, jeśli repozytorium znajduje się w dowolnym miejscu na `http://tortoisesvn.net`.

bugtraq:warnifnoissue
: Ustawcie go na `true`, jeśli chcecie by TortoiseSVN ostrzegał o pustym numerze wydania w polu tekstowym. Prawidłowe wartości to `true/false`. *Jeśli nie zdefiniowana, zakładana jest `false`.*

4.29.1.1. Numer zgłoszenia w polu tekstowym

W prostym podejściu, TortoiseSVN pokazuje użytkownikowi oddzielne pole gdzie mona wprowadzić ID błędu. Następnie osobna linia jest dołączona na początku/końcu do opisu zmiany wprowadzonego przez użytkownika.

bugtraq:message
: Ten atrybut uaktywnia system śledzenia błędów w trybie *pola wprowadzania*. Jeśli ten atrybut jest ustawiony, to TortoiseSVN poprosi, aby wprowadzić numer wydania podczas zatwierdzania zmiany. Jest używany, aby dodać wiersz na końcu opisu zmiany. Musi on zawierać `%BUGID%`, który zastępowany jest numerem zatwierdzenia. Gwarantuje to, że dziennik zatwierdzeń zawiera odniesienie do numeru wydania, które jest zawsze w spójnym formacie i może być analizowane przez narzędzia śledzenia błędów aby skojarzyć numer wydania z danym zatwierdzeniem. Jako przykład można użyć `Issue : %BUGID%`, ale to zależy od wybranego narzędzia.

bugtraq:label
: Ten tekst jest wyświetlany przez TortoiseSVN w oknie dialogowym zatwierdzenia jako etykieta pola edycji, w którym należy wpisać numer wydania. Jeśli nie jest ustawiony, będzie wyświetlany `ID błędu / Nr zgłoszenia:`. Należy pamiętać jednak, że okno nie będzie dopasowane do tego zapisu, dlatego należy utrzymać rozmiar etykiety poniżej 20-25 znaków.

bugtraq:number
: Jeśli jest ustawiony na `true` tylko cyfry są dozwolone w polu tekstowym numeru wydania. Wyjątkiem jest przecinek, więc można wpisać kilka numerów oddzielonych przecinkiem. Prawidłowe wartości to `true/false`. *Jeśli nie zdefiniowany, zakładana jest `true`.*

bugtraq:append
: Ten atrybut określa, czy ID błędu jest dołączany (true) na końcu wiadomości dziennika lub włożona (false) na początku wiadomości dziennika. Prawidłowe wartości to `true/false`. *Jeśli nie zdefiniowana, zakłada się `true`, więc dotychczasowe projekty nie są uszkadzane.*

4.29.1.2. Numery wydań przy użyciu wyrażeń regularnych

W podejściu *wyrażeń regularnych*, TortoiseSVN nie pokazuje oddzielnego pola wprowadzania, ale oznacza część opisu zmiany wprowadzonego przez użytkownika, który jest rozpoznawany przez tracker problemów. Jest to wykonywane, gdy użytkownik zapisuje opis zmiany w dzienniku. Oznacza to także, że ID błędu może być w dowolnym miejscu w wiadomości dziennika! Metoda ta jest znacznie bardziej elastyczna i została użyta w samym projekcie TortoiseSVN.

bugtraq:logregex
: Ten atrybut uaktywnia system śledzenia błędów w trybie *Regeks*. Zawiera pojedyncze wyrażenie regularne lub dwa wyrażenia regularne oddzielone znakiem nowej linii.

Jeśli są ustawione dwa wyrażenia, to pierwsze jest używane jako filtr wstępny do znalezienia wyrażeń, które zawierają identyfikatory błędu. Drugie wyrażenie następnie wyciąga odsłonięte identyfikatory błędu z wyniku pierwszego regeksa. To pozwala na użycie listy identyfikatorów błędów i wyrażeń języka naturalnego, jeśli chcesz. Można na przykład wpisać kilka błędów i to zawrzeć ciąg taki jak: „ Ta zmiana rozwiązuje problemy # 23, # 24 i # 25 ”.

Jeśli chcecie wychwycić identyfikatory błędów, użyte w wypowiedzi w powyższej wiadomości dziennika, można użyć następujących ciągów regeks, które są używane przez projekt TortoiseSVN: `[Ip]roblemy?:?(\s*(,|i)?\s*#\d+)+` i `(\d+)`.

Pierwsze wyrażenie wybiera „problemy # 23, # 24 i # 25” z opisu zmiany. Drugi regeks wyciąga zwykłe liczby dziesiętne z wyjścia pierwszego regeksu, więc zwróci „23”, „24” i „25” by ich użyć jako identyfikatorów błędu.

Rozsupłajmy nieco pierwsze wyrażenie, musi zaczynać się od słowa „problem”, możliwe z dużej litery. I może być zakończone opcjonalnym „y” (więcej niż jeden problem) i ewentualnie dwukropkiem. Za nim jest jedna lub więcej grup każda ma zero lub więcej wiodących białych znaków, opcjonalny przecinek lub „i” oraz opcjonalnie spacje. W końcu jest obowiązkowe „#” i obowiązkowa liczba dziesiętna.

Jeśli tylko jedno wyrażenie jest ustawione, to odsłonięte identyfikatory błędu muszą być dopasowany do grup ciągu regex. Przykład: `[Pp]roblem(?:y) #?(\d+)` Ta metoda jest wymagana przez kilka trackerów problemów, np. trac, ale trudniej jest zbudować wyrażenie. Zalecamy używanie tej metody wyłącznie, jeśli w dokumentacja systemu śledzenia błędów to zaleca.

Jeśli nie jesteście zaznajomieni z wyrażeniami regularnymi, prosimy spojrzeć na wprowadzenie na https://en.wikipedia.org/wiki/Regular_expression oraz dokumentację online i samouczek na http://www.regular-expressions.info/.

Nie zawsze łatwo jest uzyskać prawidłowy regeks, zatem aby pomóc w tym wprowadzono okno dialogowe testu dostępne z okna atrybutu bugtraq. Kliknijcie przycisk na prawo od pól edycji w celu przywołania go. Można w nim wpisać testowany tekst, i zmienić każdy regeks, aby zobaczyć wyniki. Jeśli wyrażenie jest nieprawidłowe, tło pola edycji zmienia się na czerwone.

Jeśli zarówno atrybut `bugtraq:message` jak i `bugtraq:logregex` są ustawiane, `logregex` ma pierwszeństwo.

Podpowiedź

Nawet jeśli nie masz trackera problemów z przechwyceniem pre-commit parsującym opis zmiany, nadal można korzystać z niego, aby przekształcić zagadnienia wymienione w opisie zmiany na linki!

I nawet jeśli nie potrzebujecie linków, numery wydań pojawią się jako osobna kolumna w oknie dziennika, co ułatwia znalezienie zmian, które odnoszą się do konkretnego wydania.

Niektóre atrybuty `tsvn:` wymagają wartości `true/false`. TortoiseSVN rozumie także `yes` jako synonim `true` i `no` jako synonim `false`.

Ustawienie atrybutów dla folderów

Te atrybuty muszą być ustawione na folderach by system działał poprawnie. Podczas zatwierdzenia pliku lub folderu atrybuty są odczytywane z tego folderu. Jeżeli atrybuty nie są tam ustawione, TortoiseSVN będzie przeszukiwać w górę drzewo folderów, aby je znaleźć, dopóki nie dotrze do folderu bez informacji o wersji albo korzenia drzewa (np. `C:\`). Jeśli możecie być pewni, że każdy użytkownik pobiera tylko np. z `trunk/` a nie z któregoś folderów podrzędnych, to wystarczy jeśli ustawicie atrybuty na `trunk/`. Jeśli nie możemy być pewni, należy ustawić atrybuty rekurencyjnie dla każdego podkatalogu. Atrybut ustawiony głębiej w hierarchii projektu zastępuje ustawienia na wyższych poziomach (bliżej `trunk/`).

> Od wersji 1.8 TortoiseSVN i Subversion używa tak zwanych `atrybutów dziedziczonych`, co oznacza że atrybut ustawiony na folderze jest automatycznie domyślnie ustawiany w folderach podrzędnych. Nie ma zatem potrzeby ustawiania zmiennej nigdzie poza folderem głównym.
>
> *Tylko* do atrybutów projektu, czyli `tsvn:`, `bugtraq:` i `webviewer:` można użyć pola wyboru Rekursywne, aby ustawić atrybut do wszystkich podkatalogów w hierarchii, bez jednoczesnego ustawienia na wszystkich plikach.

Przy dodawaniu nowych podfolderów do kopii roboczej za pomocą TortoiseSVN, wszelkie atrybuty projektu obecne w folderze nadrzędnym będą również automatycznie dodawane do nowego folderu podrzędnego.

Brak informacji o trackerze problemów z przeglądarki repozytorium

Ponieważ integracja z systemami śledzenia błędów zależy od dostępu do atrybuitów Subversion, zobaczycie wyniki tylko podczas oglądania folderów pobranych do kopii roboczej. Wczytywanie atrybutów zdalnie stanowi powolną operację, więc nie będzie widać wyników działania tej funkcji w przeglądarce repozytorium, chyba że zacznie się przeglądanie na kopii roboczej. Jeśli uruchomi się przeglądarkę repozytorium wprowadzając adres URL repozytorium, nie zobaczy się tej funkcji.

Z tego samego powodu, właściwości projektu nie zostaną przekazane automatycznie, gdy folder podrzędny jest dodawany za pomocą przeglądarki repo.

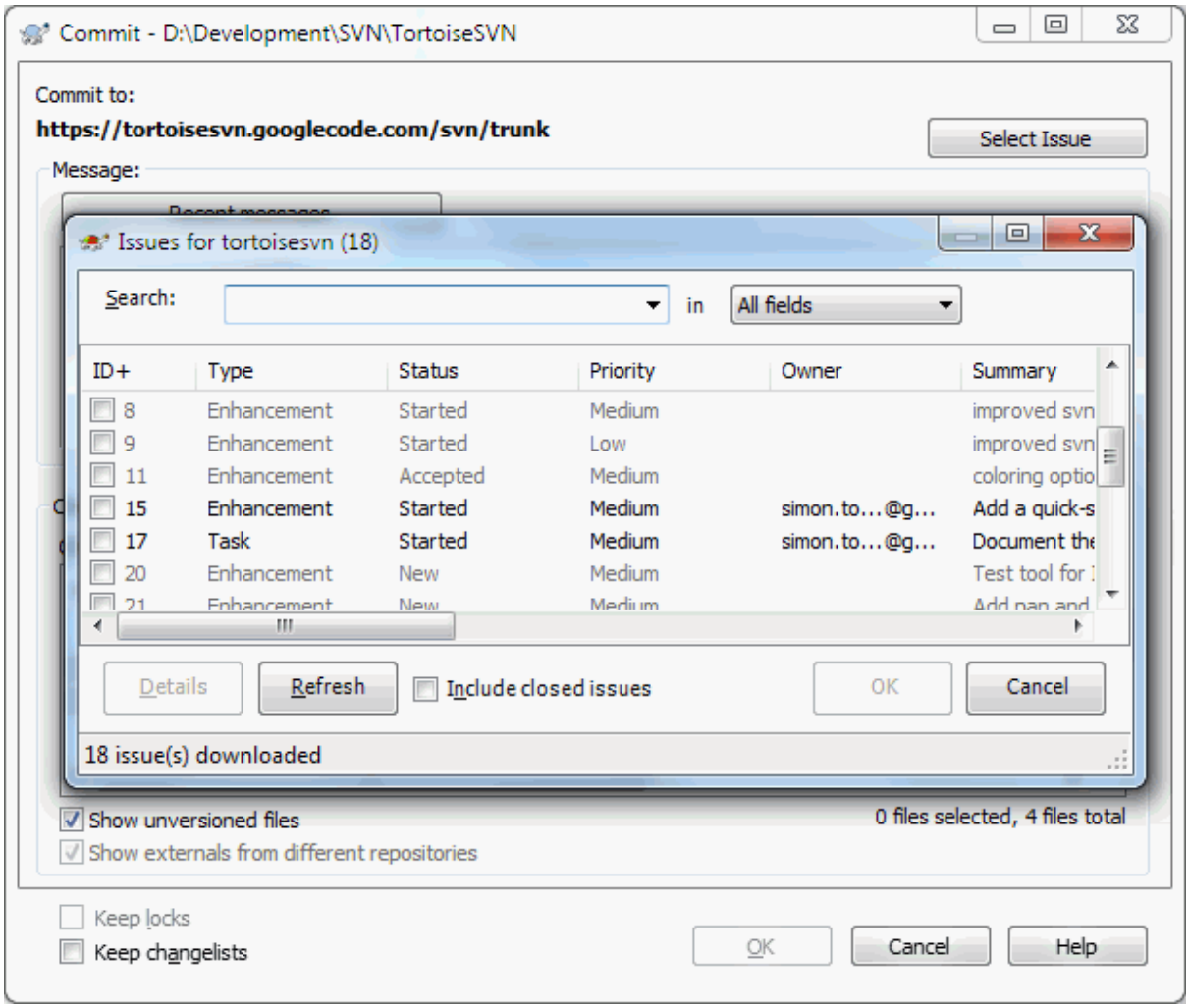
Ta integracja z systemem śledzenia błędów nie jest ograniczona tylko do TortoiseSVN; może być wykonana z dowolnym klientem Subversion. By uzyskać więcej informacji, należy przeczytać *Issue Tracker Integration Specification* [https://svn.code.sf.net/p/tortoisesvn/code/trunk/doc/notes/issuetrackers.txt] z repozytorium źródłowego TortoiseSVN. (Sekcja 3, „Licencja” wyjaśnia, jak uzyskać dostęp do repozytorium.)

4.29.2. Pobieranie informacji z trackera problemów

Poprzednia część poświęcona była dodawaniu informacji o wydaniu do opisów zmian. Ale co, jeśli chcecie uzyskać informacje od trackera problemów? Okno dialogowe zatwierdzenia posiada interfejs COM, który pozwala na integrację zewnętrznych programów mogących łączyć się z trackerem. Zazwyczaj wykonane będzie zapytanie do trackera, aby uzyskać listę przypisanych do Was otwartych zagadnień, tak aby można było wybrać zagadnienia, które są przedmiotem niniejszego zatwierdzenia.

Taki interfejs jest oczywiście bardzo specyficzny dla systemu śledzenia błędów, więc nie możemy dostarczyć tej części, a opis jak stworzyć taki program wykracza poza zakres tego podręcznika. Definicję interfejsu i próbki wtyczek w C# i C++/ATL można uzyskać z folderu `contrib` w *repozytorium TortoiseSVN* [https://svn.code.sf.net/p/tortoisesvn/code/trunk/contrib/issue-tracker-plugins]. (Sekcja 3, „Licencja” wyjaśnia, jak uzyskać dostęp do repozytorium.) Podsumowanie API znajduje się również w Rozdział 7, *Interfejs IBugtraqProvider*. Innym (roboczym) przykładem wtyczki np. w C # jest *Gurtle* [http://code.google.com/p/gurtle/], który implementuje wymagany interfejs COM do współpracy z trackerem wydań *Google Code* [http://code.google.com/hosting/].

Dla celów ilustracji, załóżmy, że administrator systemu pod zapewnił dostęp do wtyczki trackera problemów, które masz zainstalowane oraz zostały już skonfigurowane niektóre z kopii roboczych do korzystania z wtyczki w oknie ustawień TortoiseSVN. Po otwarciu okna dialogowego zatwierdzenia z kopii roboczej, do której wtyczka została przypisana, pojawi się nowy przycisk w górnej części okna.



Rysunek 4.71. Przykładowe okno dialogowe zapytania trackera problemów

W tym przykładzie można wybrać jedno lub więcej otwartych zagadnień. Wtyczka może wtedy wygenerować specjalnie sformatowany tekst, który dodaje do wiadomości dziennika.

4.30. Integracja z internetowymi przeglądarkami repozytoriów

Istnieje kilka dostępnych internetowych przeglądarek repozytorium działających z Subversion, takich jak *ViewVC* [http://www.viewvc.org/] i *WebSVN* [http://websvn.tigris.org/]. TortoiseSVN zapewnia możliwość połączenia z tych przeglądarek.

Możecie zintegrować wybraną przeglądarkę repo w TortoiseSVN. Aby to zrobić, musicie zdefiniować kilka atrybutów, które zdefiniują powiązania. Muszą być ustawione na folderach: (Sekcja 4.18, „Ustawienia projektu”)

webviewer:revision
: Atrybut ten należy ustawić na adres URL przeglądarki repo, aby wyświetlić wszystkie zmiany w konkretnej wersji. Należy prawidłowo enkodować URI i zawrzeć `%REVISION%`. `%REVISION%` jest zastępowany numerem wersji w zapytaniu. Pozwala to TortoiseSVN wyświetlić wejście menu kontekstowego Context Menu → Podgląd wersji w przeglądarce w oknie dziennika.

webviewer:pathrevision
: Atrybut ten należy ustawić na adres URL przeglądarki repo, aby wyświetlić zmiany do konkretnego pliku w określonej wersji. Należy prawidłowo enkodować URI i zawrzeć `%REVISION%` oraz `%PATH%`. % `%PATH%` jest zastępowany ścieżkę względną do katalogu głównego repozytorium. Pozwala to TortoiseSVN wyświetlić

wejście menu kontekstowego Context Menu → Podgląd wersji dla ścieżki w przeglądarce w oknie dziennika. Na przykład, jeśli klikniecie prawym przyciskiem myszy w dolnym panelu okna dziennika na wpisie pliku `/trunk/src/file` to `%PATH%` w adresie URL zostanie zastąpiony przez `/trunk/src/file`.

Można również używać względnych adresów URL zamiast bezwzględnych. Jest to przydatne w przypadku, gdy przeglądarka internetowa jest w tej samej domenie/serwerze co repozytorium źródłowe. W przypadku, gdy nazwa domeny zmienia się kiedykolwiek, nie musisz dostosować atrybutów `webviewer:revision` ani `webviewer:pathrevision`. Format jest taki sam jak dla atrybutu `bugtraq:url`. Zobacz Sekcja 4.29, „Integracja z systemami śledzenia błędów / śledzenia problemów”.

Ustawienie atrybutów dla folderów

Te atrybuty muszą być ustawione na folderach by system działał poprawnie. Podczas zatwierdzenia pliku lub folderu atrybuty są odczytywane z tego folderu. Jeżeli atrybuty nie są tam ustawione, TortoiseSVN będzie przeszukiwać w górę drzewo folderów, aby je znaleźć, dopóki nie dotrze do folderu bez informacji o wersji albo korzenia drzewa (np. `C:\`). Jeśli możecie być pewni, że każdy użytkownik pobiera tylko np. z `trunk/` a nie z któregoś folderów podrzędnych, to wystarczy jeśli ustawicie atrybuty na `trunk/`. Jeśli nie możemy być pewni, należy ustawić atrybuty rekurencyjnie dla każdego podkatalogu. Atrybut ustawiony głębiej w hierarchii projektu zastępuje ustawienia na wyższych poziomach (bliżej `trunk/`).

Tylko do atrybutów projektu, czyli `tsvn:`, `bugtraq:` i `webviewer:` można użyć pola wyboru Rekursywne, aby ustawić atrybut do wszystkich podkatalogów w hierarchii, bez jednoczesnego ustawienia na wszystkich plikach.

Przy dodawaniu nowych podfolderów do kopii roboczej za pomocą TortoiseSVN, wszelkie atrybuty projektu obecne w folderze nadrzędnym będą również automatycznie dodawane do nowego folderu podrzędnego.

Ograniczenia użycia przeglądarki repozytorium

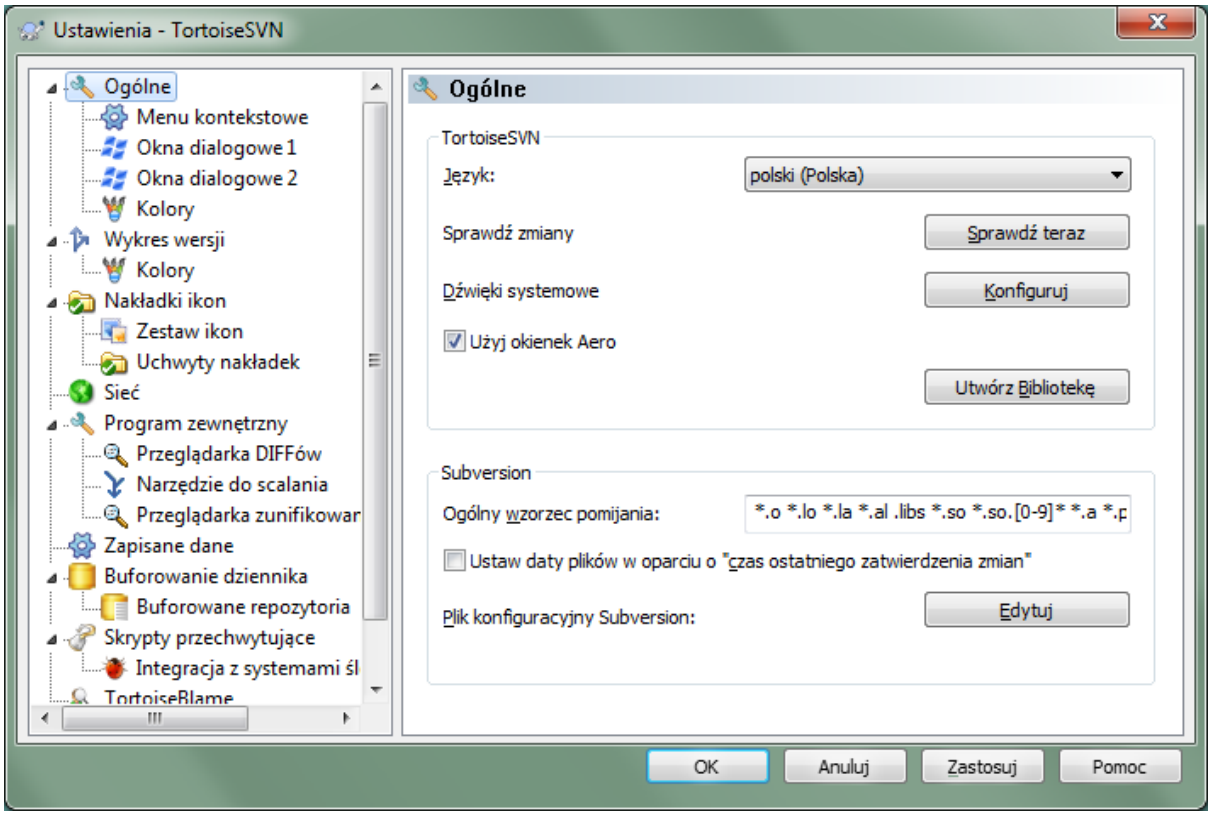
Ponieważ integracja przeglądarki repozytorium zależy od dostępu do atrybutów Subversion, zobaczycie wyniki tylko podczas oglądania folderów pobranych do kopii roboczej. Wczytywanie atrybutów zdalnie stanowi powolną operację, więc nie będzie widać wyników działania tej funkcji w przeglądarce repozytorium, chyba że zacznie się przeglądanie na kopii roboczej. Jeśli uruchomi się przeglądarkę repozytorium wprowadzając adres URL repozytorium, nie zobaczy się tej funkcji.

Z tego samego powodu, właściwości projektu nie zostaną przekazane automatycznie, gdy folder podrzędny jest dodawany za pomocą przeglądarki repo.

4.31. Ustawienia TortoiseSVN

Aby dowiedzieć do czego służą poszczególne ustawienia, po prostu pozostawcie kursor na chwilę na polu edycji/wyboru ... i wyskoczy pomocna podpowiedź.

4.31.1. Ustawienia ogólne



Rysunek 4.72. Okno dialogowe ustawień, strona Ogólne

To okno dialogowe pozwala określić preferowany język oraz konkretne ustawienia Subversion.

Język
: Wybiera wasz język interfejsu użytkownika. Oczywiście należy zainstalować odpowiedni pakiet językowy zanim wybierze się język UI inny niż domyślny angielski.

Sprawdź zmiany
: TortoiseSVN połączy się okresowo ze swoją witryną pobierania by sprawdzić, czy jest dostępna nowsza wersja programu. Jeśli tak, pokaże link powiadomienia w oknie dialogowym zatwierdzenia. Użyjcie Sprawdź teraz jeśli chcecie otrzymać odpowiedź od razu. Nowa wersja nie zostanie pobrana, po prostu wyświetli się okno z informacją, że nowa wersja jest dostępna.

Dźwięki systemowe
: TortoiseSVN posiada trzy własne dźwięki, które są instalowane domyślnie.

  - Błąd
  - Uwaga
  - Ostrzeżenie

  Możesz wybrać inne dźwięki (lub wyłączyć te dźwięki całkowicie) w Panelu sterowania Windows. Konfiguruj jest skrótem do Panelu sterowania.

Użyj okienek Aero
: W systemie Windows Vista i późniejszych decyduje czy dialogi korzystają ze stylu Aero.

Utwórz Bibliotekę
: W Windows 7 można utworzyć bibliotekę, która grupuje kopie robocze,rozproszone w różnych miejscach w systemie.

Ogólny wzorzec pomijania

Ogólne wzorce pomijania są stosowane do zapobiegania pojawienia się niewersjonowanych plików np. oknie dialogowym zatwierdzenia. Pliki pasujące do wzorców są również ignorowane przez import. Ignorowane są pliki lub katalogi z pasującymi nazwami lub rozszerzeniami. Wzorce są oddzielone spacjami np. `bin obj *.bak *.~?? *.jar *.[Tt]mp`. Te wzorce nie powinny zawierać żadnych separatorów ścieżki. Zauważ też, że nie sposób odróżnić plików od katalogów. Czytajcie Sekcja 4.14.1, „Dopasowanie wzorców w listach ignorowania” dla uzyskania dodatkowych informacji na temat składni dopasowania do wzorca.

Zauważcie, że wzorce ignorowania, które są tu określone, będą miały wpływ również inne klienty Subversion zainstalowane na komputerze, w tym klienta linii poleceń.

Ostrzeżenie

Jeśli używacie pliku konfiguracyjnego Subversion do ustawienia wzorca `global-ignores`, zastępuje on ustawienia na tym ekranie. Plik konfiguracyjny Subversion jest dostępny poprzez Edytuj jak opisano poniżej.

Ten wzorzec pomijania będzie dotyczył wszystkich projektów. Nie jest on wersjonowany, zatem nie obejmie innych użytkowników. Z drugiej strony można użyć wersjonowanych atrybutów `svn:ignore` albo `svn:global-ignores` wyłączyć pliki oraz foldery z kontroli wersji. Przejrzyjcie Sekcja 4.14, „Ignorowanie plików i folderów” by uzyskać więcej informacji.

Ustawcie daty plików w oparciu o „czas ostatniego zatwierdzenia zmian”

Ta opcja mówi TortoiseSVN ustawić datę pliku na czas ostatniego zatwierdzenia podczas wykonywania pobrania lub aktualizacji. W przeciwnym razie TortoiseSVN będzie korzystać z bieżącej daty. Przy tworzeniu oprogramowania na ogół najlepiej jest używać bieżącej daty, ponieważ systemy kompilacji zwykle po znacznikachdaty decydują, które pliki wymagają kompilacji. Jeśli używacie „czas ostatniego zatwierdzania zmian” i powrócicie do starszej wersji pliku, projekt może nie skompilować się tak, jak się tego spodziewaliśmy.

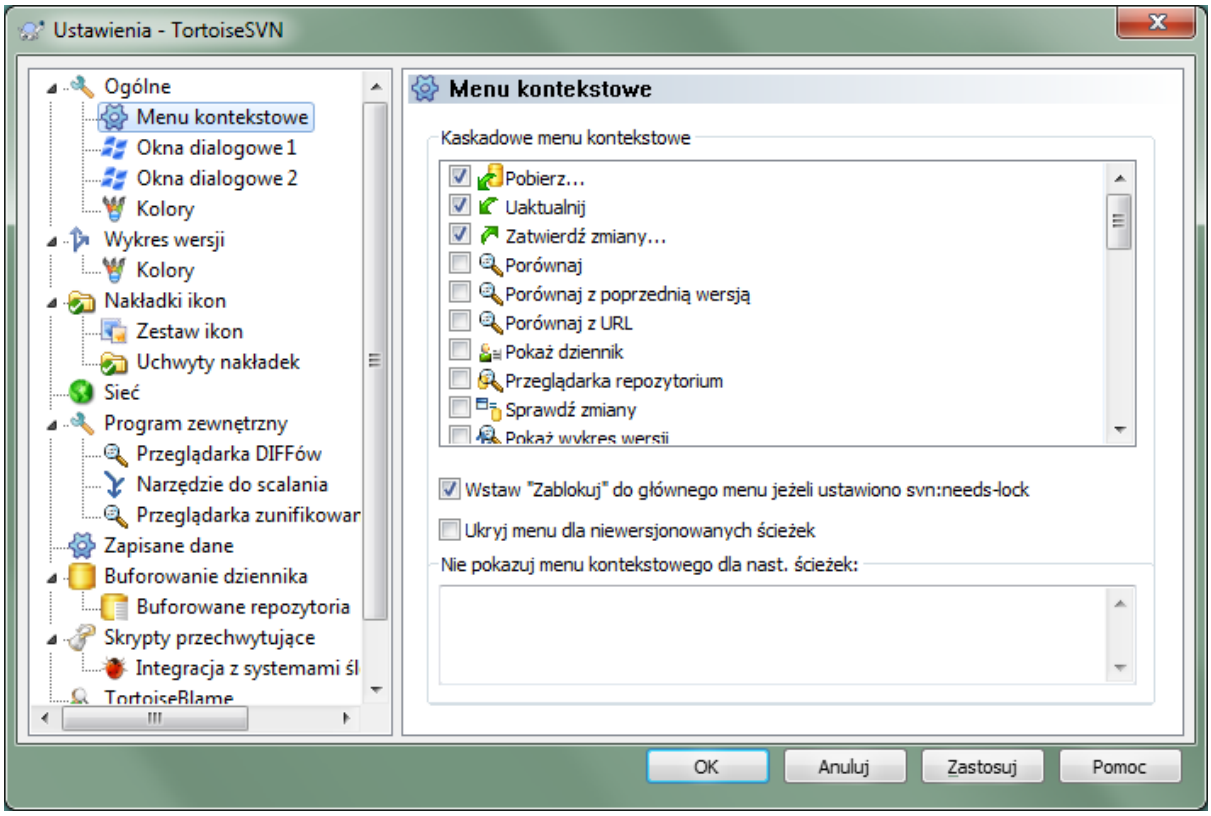
Plik konfiguracyjny Subversion

Użyjcie Edytuj by edytować plik konfiguracyjny Subversion bezpośrednio. Niektóre ustawienia nie mogą być modyfikowane bezpośrednio przez TortoiseSVN i muszą być ustawione tutaj. Aby uzyskać więcej informacji o pliku `config` Subversion, przejrzyjcie *Obszar Konfiguracji środowiska wykonawczego* [http://svnbook.red-bean.com/en/1.8/svn.advanced.confarea.html]. Sekcja *Automatyczne ustawianie atrybutów* [http://svnbook.red-bean.com/en/1.8/svn.advanced.props.html#svn.advanced.props.auto] jest szczególnie interesująca, a to jest konfigurowane w tym miejscu. Należy pamiętać, że Subversion może odczytywać informacje z różnych miejsc, i trzeba wiedzieć, które z nich ma pierwszeństwo. Patrz *Konfiguracja a rejestry systemu Windows* [http://svnbook.red-bean.com/en/1.8/svn.advanced.confarea.html#svn.advanced.confarea.windows-registry], aby dowiedzieć się więcej.

Stosujcie lokalne modyfikacje do svn:externals podczas aktualizacji

Ta opcja mówi TortoiseSVN, aby zawsze stosować lokalne modyfikacje atrybutu `svn:externals` podczas aktualizowania kopii roboczej.

4.31.1.1. Ustawienia menu kontekstowego



Rysunek 4.73. Okno dialogowe ustawień, strona Menu kontekstowe

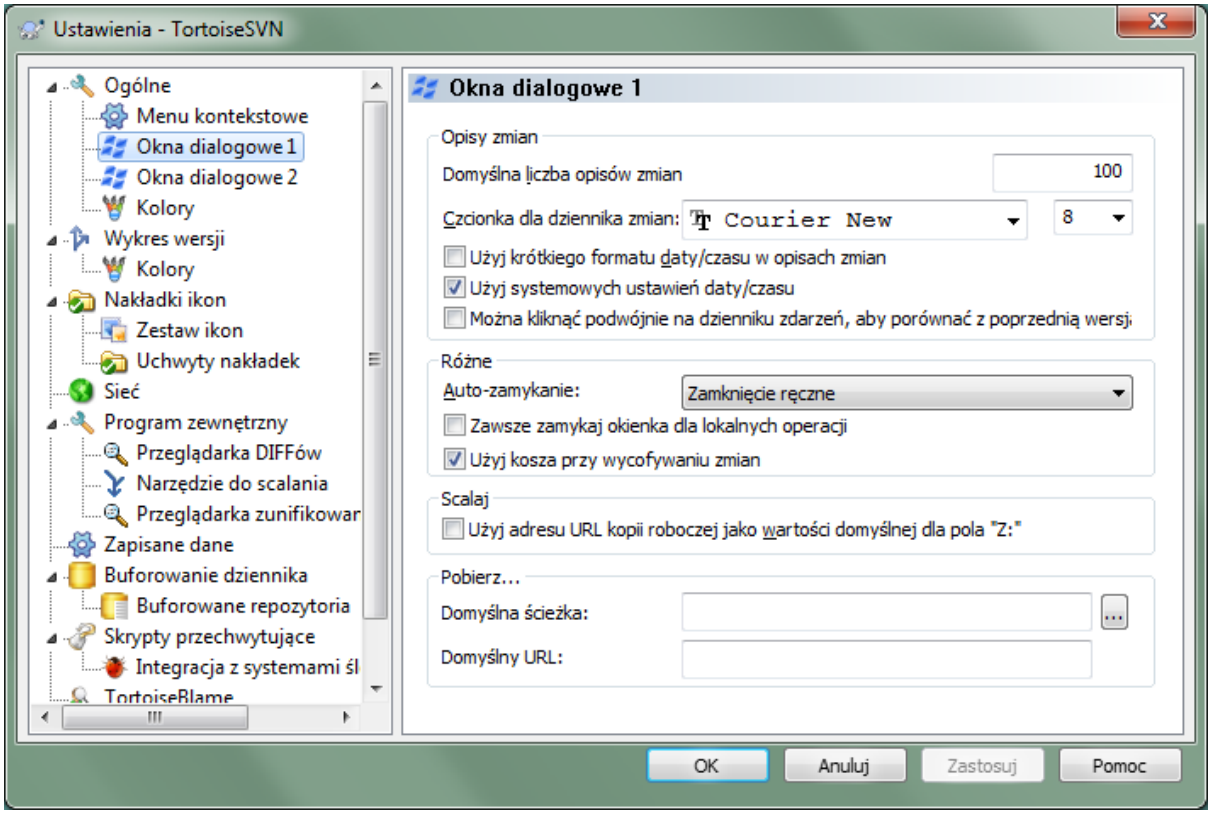
Strona ta umożliwia określenie, które z wpisów menu kontekstowego TortoiseSVN pojawią się w głównym menu kontekstowym, a które pojawią się w podmenu TortoiseSVN. Domyślnie większość rzeczy są odznaczone i pojawiają się w podmenu.

Występuje tu przypadek szczególny dla Zablokuj. Można oczywiście promować go do najwyższego poziomu korzystając z listy powyżej, ale ponieważ większość plików nie ma potrzeby blokowania to tylko powoduje bałagan. Jednak plik z atrybutem `svn:needs-lock` potrzebuje tego działania za każdym razem przy edycji, więc w tym przypadku jest on bardzo przydatny na najwyższym poziomie. Zaznaczenie pola tutaj oznacza, że jeśli jest zaznaczony plik, który ma ustawiony atrybut `svn:needs-lock`, Zablokuj pojawi się zawsze na najwyższym poziomie.

Przez większość czasu nie ma potrzeby używania menu kontekstowego TortoiseSVN poza folderami, które są pod kontrolą wersji przez Subversion. W przypadku folderów niewersjonowanych, tak naprawdę w menu kontekstowym potrzeba polecenia pobrania. Jeśli zaznaczycie opcję `Ukryj menu dla niewersjonowanych ścieżek` , TortoiseSVN nie będzie dodawać swoich wpisów do menu kontekstowego dla folderów bez informacji o wersji. Ale wpisy są dodawane do wszystkich elementów i ścieżek w wersjonowanych folderach. Można przy tym odzyskać pozycje dla katalogów bez informacji o wersji przytrzymując klawisz klawisz **Shift** podczas otwierania menu kontekstowego.

Jeśli istnieją jakieś ścieżki na komputerze, dla których po prostu nie chcecie by menu kontekstowe TortoiseSVN w ogóle pojawiało się, można je wypisać w oknie na dole.

4.31.1.2. Ustawienia TortoiseSVN dla okien dialogowych 1



Rysunek 4.74. Okno dialogowe ustawień, strona Okna dialogowe 1

To okno pozwala skonfigurować niektóre okna dialogowe TortoiseSVN by wyglądały tak, jak lubicie.

Domyślna liczba opisów zmian
: Ogranicza liczbę opisów zmian, które TortoiseSVN ładuje po raz pierwszy wybiera się TortoiseSVN → Pokaż dziennik Przydatne dla wolnych połączeń z serwerem. Zawsze możecie użyć Wszystkie lub Następne 100, aby uzyskać więcej opisów.

Czcionka komunikatów
: Wybiera krój i rozmiar czcionki do wyświetlania samego opisu zmiany w środkowym panelu okna dziennika wersji i podczas tworzenia opisu zmiany w oknie zatwierdzenia.

Użyj krótkiego formatu daty/czasu w opisach zmian
: Jeśli standardowa długa wiadomość zużywa zbyt dużo miejsca na ekranie używa się skróconej formy.

Można kliknąć podwójnie na dzienniku zdarzeń, aby porównać z poprzednią wersją
: Jeśli często zajmujecie się porównaniem zmian w górnym panelu w oknie dziennika, możecie użyć tej opcji by umożliwić to działanie przez podwójne kliknięcie. Nie jest ona domyślnie włączona, ponieważ pobieranie różnicy jest często długim procesem i wiele osób woli uniknąć oczekiwania po przypadkowym dwukliku, dlatego ta opcja nie jest domyślnie włączona.

Automatyczne zamknięcie
: TortoiseSVN może automatycznie zamykać wszystkie okna dialogowe postępu, gdy akcja jest zakończona bez błędów. To ustawienie pozwala na wybranie warunków dla zamykania okien dialogowych. Domyślnym (zalecanym) ustawieniem jest Zamknięcie ręczne, które pozwala na przejrzenie wszystkich wiadomości i sprawdzenie co się stało. Można jednak zdecydować, że chcecie zignorować niektóre rodzaje wiadomości i okno zamknie się automatycznie, jeśli nie ma istotnych zmian.

  Automatyczne zamknięcie, jeśli nie ma scaleń, dodań lub usunięć oznacza, że okno postępu zostanie zamknięte, gdyby wystąpiły tylko proste aktualizacje, ale jeśli jakieś zmiany z repozytorium zostały scalone z

twoimi, lub jeżeli jakieś pliki zostały dodane lub usunięte, okno pozostanie otwarte. Będzie również otwarte, czy powstaną jakieś konflikty lub błędy w trakcie operacji.

Automatycznie zamykaj jeśli nie ma konfliktów rozluźnia kryteria dalej i zamyka okno, nawet jeśli były scalenia, dodania lub usunięcia. Jednak, gdyby zaistniały jakichkolwiek konflikty lub błędy, okno pozostaje otwarte.

Automatyczne zamknięcie, jeśli nie ma błędów zawsze zamyka okno, nawet gdyby były konflikty. Jedynym warunkiem, który utrzymuje okno otwarte jest błąd, który występuje, gdy Subversion nie jest w stanie wykonać zadania. Na przykład, aktualizacja nie powiedzie się, ponieważ serwer jest niedostępny, lub zatwierdzenia nie powiodą się, ponieważ kopia robocza jest nieaktualna.

Zawsze zamykaj okienka dla lokalnych operacji
: Lokalne operacje takie jak dodawanie plików lub wycofywanie zmian nie wymagają łączenia się do repozytorium i są szybko przetwarzane, więc okno dialogowe postępu jest często mało interesujące. Wybierzcie tę opcję, jeśli chcecie, aby automatycznie zamknąć okno dialogowe postępu po tych czynnościach, chyba że pojawią się błędy.

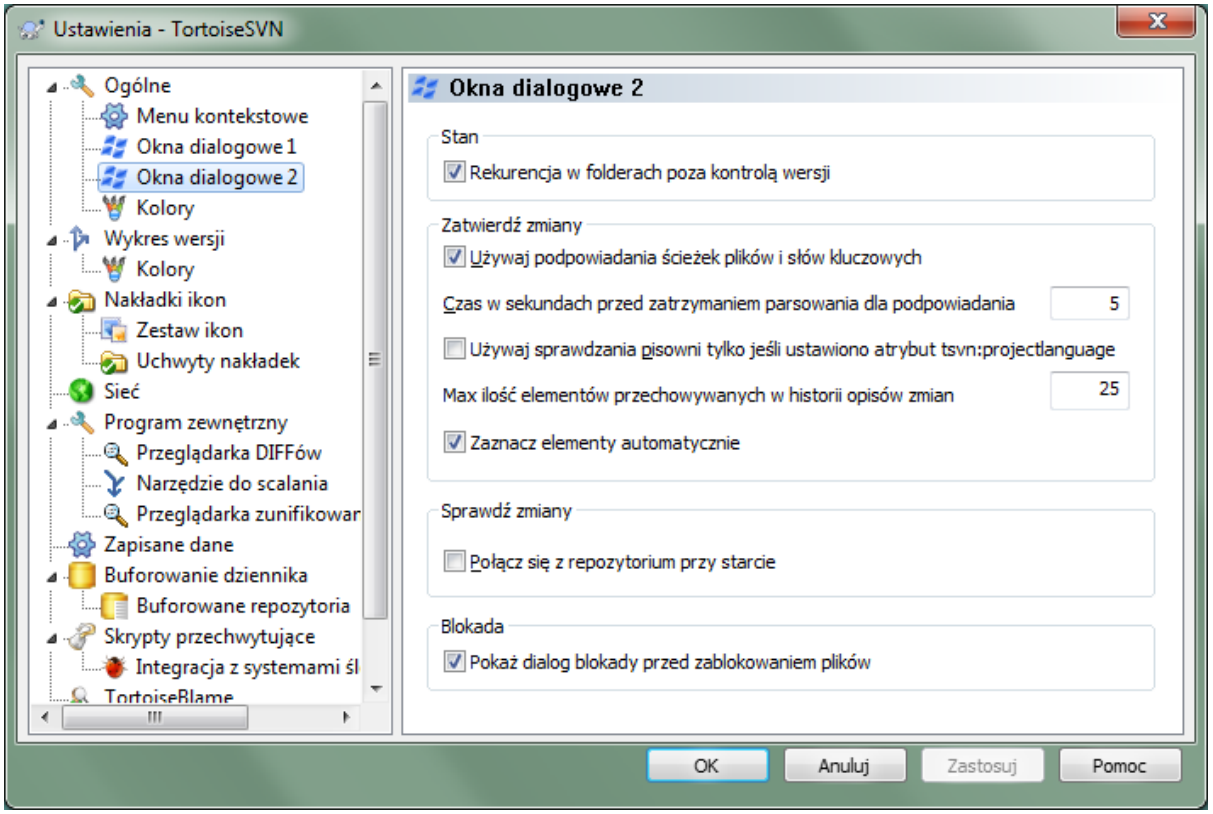
Użyj kosza przy wycofywaniu zmian
: Podczas wycofania lokalnych modyfikacji, zmiany są odrzucane. TortoiseSVN daje dodatkowe zabezpieczenie poprzez wysłanie zmodyfikowanego pliku do kosza przed przywróceniem pierwotnej kopii. Jeśli wolicie, aby przeskoczyć przenoszenie do kosza, usuńcie zaznaczenie tej opcji.

Użyj adresu URL kopii roboczej jako domyZ:
: W oknie dialogowym scalenia, domyślnym zachowaniem jest pamiętanie URL Z: między scaleniami. Jednak niektórzy ludzie lubią wykonywać scalenia z wielu różnych punktów w hierarchii, wtedy łatwiej rozpocząć z adresu URL bieżącej kopii roboczej. Może on być edytowany by odnieść się do równoległej ścieżki na innej gałęzi.

Domyślna ścieżka pobierania
: Można określić domyślną ścieżkę dla pobrań. Jeśli trzymasz wszystkie pobrania w jednym miejscu, dobrze jest mieć dysk i folder wstępnie wypełnione, trzeba tylko dodać nową nazwę folderu na końcu.

Domyślny URL pobierania
: Można także określić domyślny adres URL dla pobrań. Jeśli często pobieracie podprojekty pewnego bardzo dużego projektu, może być przydatne wstępne wypełnienie URL by trzeba było dodać tylko nazwę podprojektu na końcu.

4.31.1.3. Ustawienia TortoiseSVN dla okien dialogowych 2



Rysunek 4.75. Okno dialogowe ustawień, strona Okna dialogowe 2

Rekurencja w folderach poza kontrolą wersji
: Jeśli to pole jest zaznaczone (stan domyślny), a następnie, gdy stan niewersjonowanego katalogu jest pokazany w oknach Dodaj, Zatwierdź lub Sprawdź zmiany, każdy podrzędny plik i folder jest także pokazany. Jeśli usuniecie zaznaczenie tego pola, jest widoczny tylko rodzic bez informacji o wersji. Usunięcie zaznaczenia redukuje bałagan w tych oknach dialogowych. W takim przypadku po wybraniu folderu bez informacji o wersji do Dodania, jest on dodawany rekurencyjnie.

  W oknie dialogowym Sprawdź zmiany można wybrać podgląd ignorowanych pozycji. Jeśli to pole jest zaznaczone, gdy znajduje się ignorowany folder, wszystkie elementy podrzędne również się pojawią.

Używaj podpowiadania ścieżek plików i słów kluczowych
: Okno dialogowe zatwierdzenia zawiera ułatwienie analizowania listy plików do zatwierdzenia. Po wpisaniu pierwszych trzech liter z pozycji na liście, pojawia się pole automatycznego uzupełniania i można nacisnąć klawisz Enter, aby zakończyć nazwę pliku. Zaznaczcie pole, aby włączyć tę funkcję.

Czas w sekundach przed zatrzymaniem parsowania do podpowiedzi
: Analizator składni autouzupełniania może być bardzo powolny jeśli ma wiele dużych plików do sprawdzenia. Ten limit czasu powstrzymuje okno zatwierdzenia przed wyszukiwaniem trwającym zbyt długo. Jeśli brakuje ważnych informacji autouzupełnienia, można wydłużyć limit czasu.

Używaj sprawdzania pisowni tylko jeśli ustawiono atrybut `tsvn:projectlanguage`
: Jeśli nie chcecie korzystać ze sprawdzenia pisowni dla wszystkich zatwierdzeń, zaznaczcie to pole. Moduł sprawdzania pisowni będzie nadal aktywny tam, gdzie atrybuty projektu tego wymagają.

Max ilość elementów przechowywanych w historii opisów zmian
: Po wpisaniu w wiadomości dziennika w oknie zatwierdzenia, TortoiseSVN zapisuje ją do ewentualnego ponownego wykorzystania później. Domyślnie będzie to ostatnie 25 opisów zmian dla każdego repozytorium, ale można dostosować tę liczbę tutaj. Jeśli macie wiele różnych repozytoriów, może zechcecie ograniczyć ją, aby uniknąć wypełnienia rejestru.

Zauważcie, że to ustawienie dotyczy tylko wiadomości, które wpisano na tym komputerze. Nie ma ono nic wspólnego z buforem dziennika.

Zaznacz elementy automatycznie
: Normalne postępowanie w oknie zatwierdzenia dla wszystkich zmodyfikowanych (wersjonowanych) elementów, które zostaną wybrane do zatwierdzenia automatycznie. Jeśli wolisz, aby rozpocząć bez wybranych i zaznaczać pozycje do zatwierdzenia ręcznie, usuń zaznaczenie tego pola.

Ponownie otwórz okno po zatwierdzeniu jeśli pozostaną niezatwierdzone elementy
: Otwiera to okno dialogowe zatwierdzenia automatycznie w tym samym folderze po poprawnym zatwierdzeniu. Okno jest otwierane tylko jeśli pozostały jeszcze elementy do zatwierdzenia.

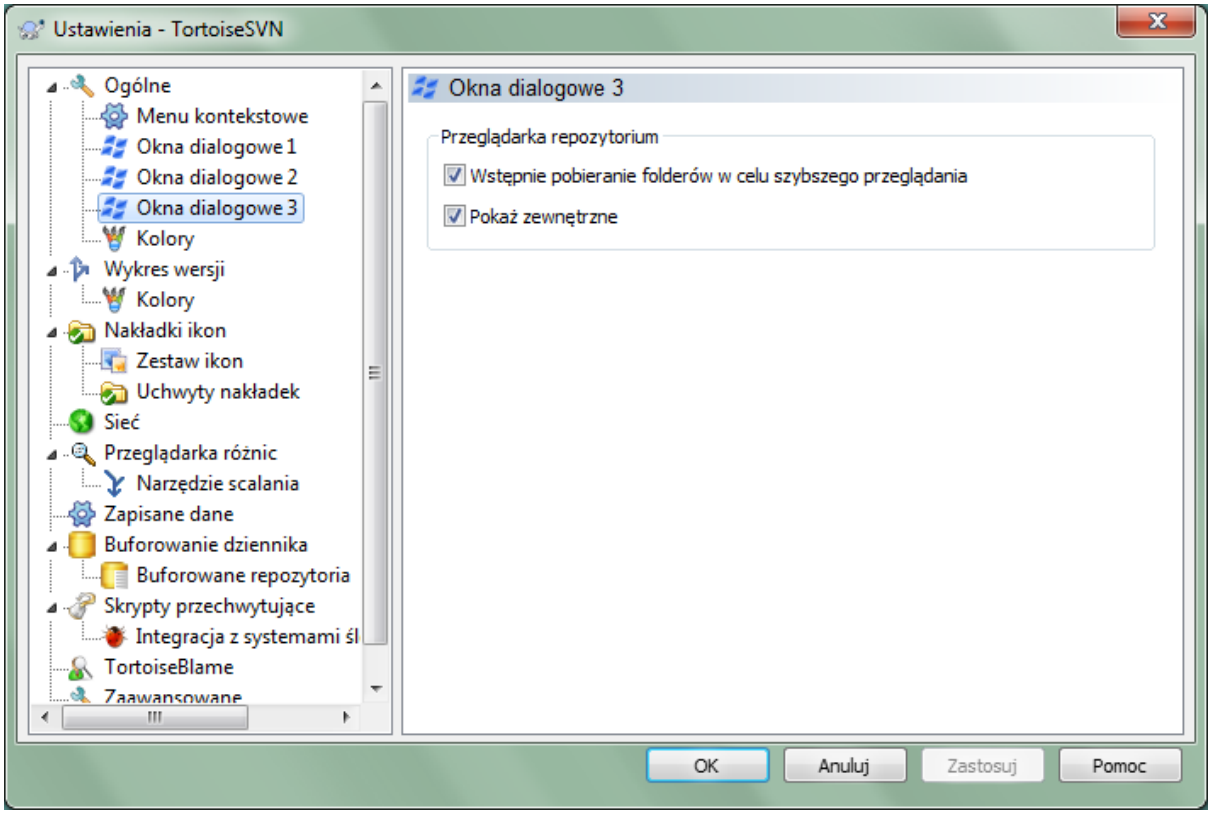
Połącz się z repozytorium przy starcie
: Okno dialogowe Sprawdź zmiany sprawdza domyślnie tylko kopię roboczą a łączy z się repozytorium tylko po kliknięciu Sprawdź repozytorium. Jeśli zawsze chcecie sprawdzać repozytorium, można skorzystać z opcji, aby wymusić automatyczne wykonanie tego działania.

Pokaż dialog blokady przed zablokowaniem plików
: Po wybraniu jednego lub więcej plików i użyciu TortoiseSVN → Blokada na te pliki nakładana jest blokada. W niektórych projektach jest zwyczajem, aby wpisać komunikat blokady wyjaśniający, dlaczego trzeba było zablokować pliki. Jeśli nie używacie wiadomości blokady można usunąć zaznaczenie tego pola, co spowoduje pominięcie okna dialogoweco i zablokuje pliki natychmiast.

  Jeśli użyjecie polecenia blokady na folderze, okno blokady pokazywane jest zawsze, by dać jednocześnie możliwość wyboru plików do zablokowania.

  Jeśli projekt korzysta z atrybutu `tsvn:lockmsgminsize`, okno dialogowe blokady pojawi się niezależnie od tego ustawienia, ponieważ projekt *wymaga* wiadomości blokowania.

4.31.1.4. Ustawienia TortoiseSVN dla okien dialogowych 3



Rysunek 4.76. Okno dialogowe ustawień, strona Okna dialogowe 3

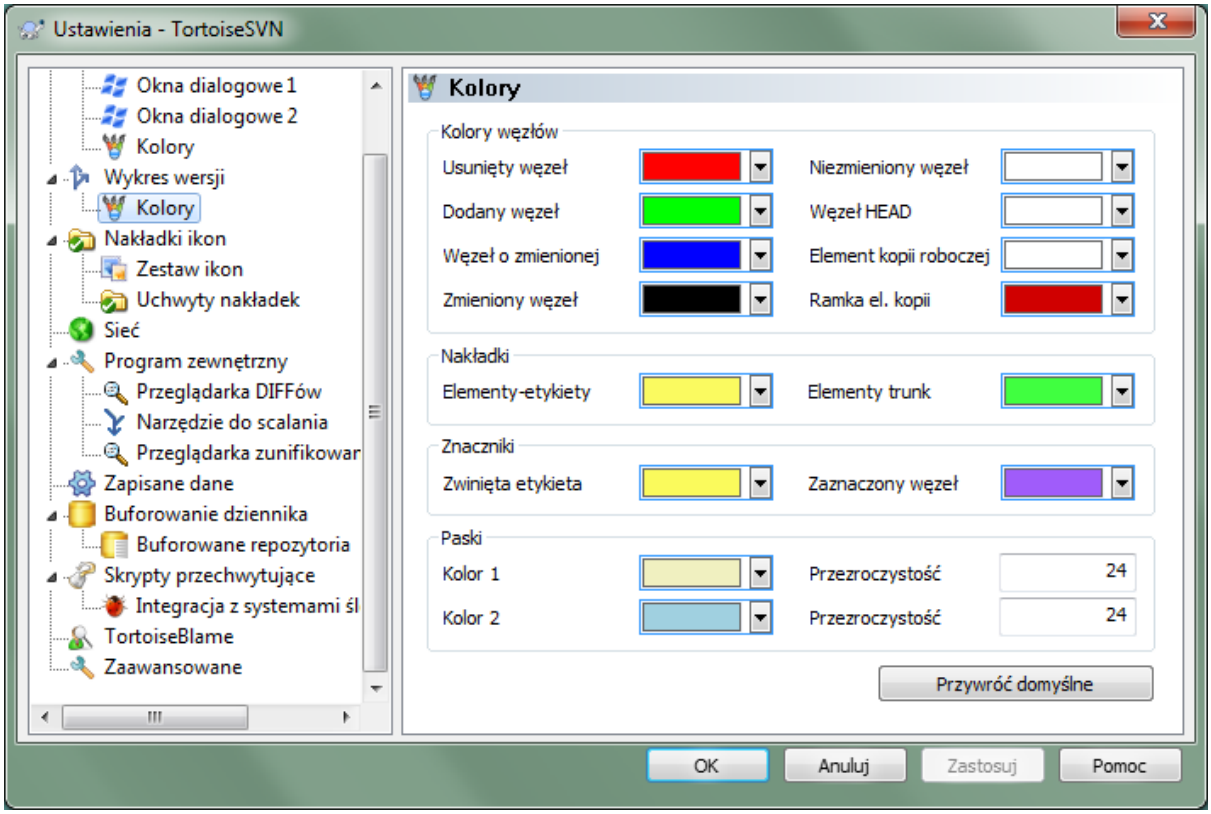
Wstępnie pobieranie folderów w celu szybszego przeglądania
: Jeśli ta opcja jest zaznaczona (stan domyślny), to przeglądarka repozytorium pobiera informacje o pokazywanych folderach w tle. W ten sposób, gdy tylko rozpoczniesz przeglądanie jednego z tych folderów, informacje są już dostępne.

  Niektóre serwery jednak nie potrafią obsłużyć wielu żądań, które to powoduje lub nie są poprawnie skonfigurowane przez co traktują tak wiele żądań jako coś złego i zaczynają je blokować. W tym przypadku można wyłączyć wstępne pobieranie tutaj.

Pokaż zewnętrzne
: Jeśli ta opcja jest zaznaczona (stan domyślny), to w przeglądarka repozytorium pokazuje pliki i foldery, które są dołączone z użyciem atrybutu `svn:externals` jako normalne pliki i foldery, ale z ikoną nakładki oznaczającą je jako pochodzące ze źródła zewnętrznego.

  Podobnie jak w przypadku funkcji pobierania wstępnego wyjaśnionej powyżej, to również może powodować zbyt wielkie obciążenie na słabych serwerach. W takim przypadku można wyłączyć tę funkcję tutaj.

4.31.1.5. Ustawienia kolorów TortoiseSVN



Rysunek 4.77. Okno dialogowe ustawień, strona Kolory

To okno dialogowe pozwala na konfigurację kolorów tekstu używanego w oknach dialogowych TortoiseSVN tak, jak lubisz.

Konflikt / przeszkoda
: Wystąpił konflikt podczas aktualizacji, lub może wystąpić w trakcie scalenia. Uaktualnienie zostało zakłócone przez istniejący niewersjonowany plik/folder o takiej samej nazwie jak element wersjonowany.

  Kolor ten jest stosowany również do komunikatów o błędach w oknach dialogowych postępu.

Dodane pliki
: Elementy dodane do repozytorium.

Brakujący / usunięty / zastąpiony
: Elementy usunięte z repozytorium, brakujące w kopii roboczej lub usunięte z kopii roboczej i zastąpione przez inny plik o tej samej nazwie.

Scalono
: Zmiany z repozytorium skutecznie scalone w KR bez stwarzania konfliktów.

Zmodyfikowany / skopiowany
: Dodane z historią, jak również ścieżki skopiowane w repozytorium. Stosowany również w oknie dialogowym dziennika dla wpisów, które obejmują skopiowane elementy.

Usunięty węzeł
: Element, który został usunięty z repozytorium.

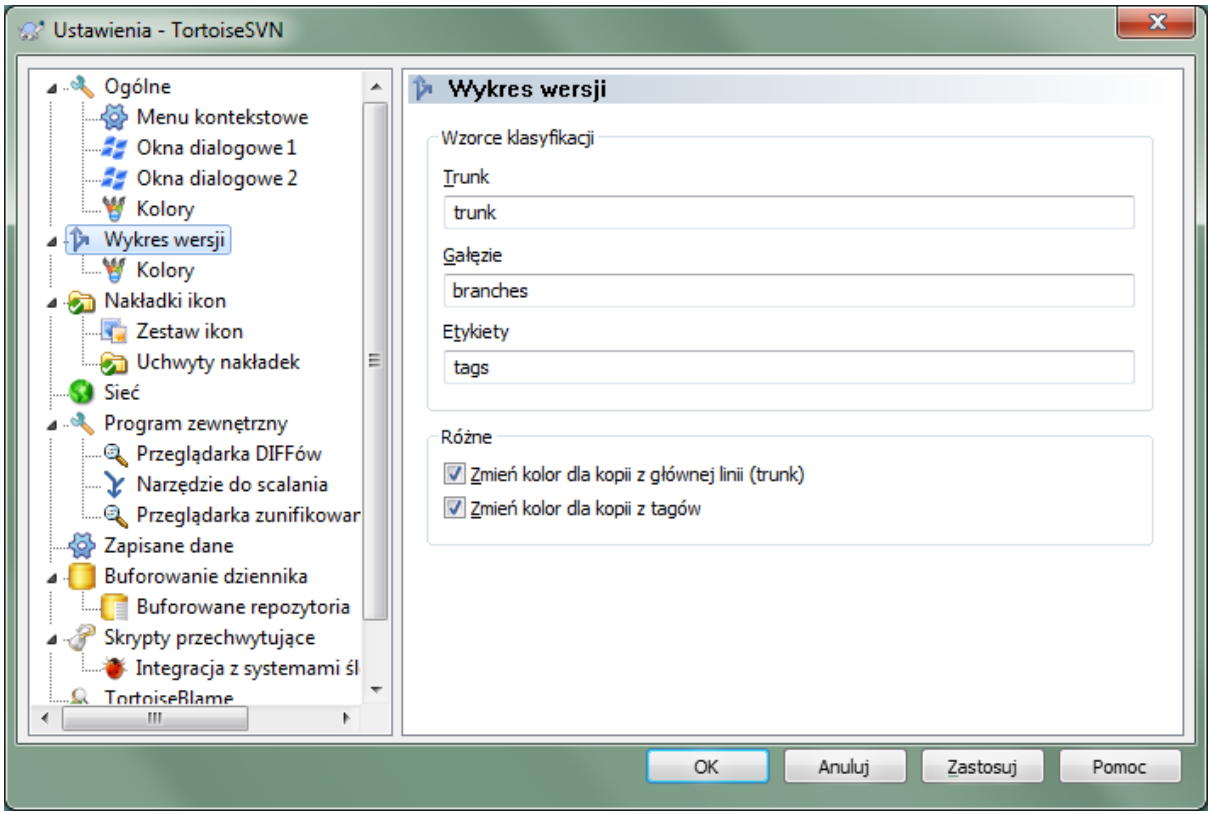
Dodany węzeł
: Element, który został dodany do repozytorium przez dodanie, skopiowanie lub przeniesienie.

Węzeł o zmienionej nazwie
: Element, której nazwa została zmieniona w repozytorium.

Węzeł zastąpiony
: Oryginalny element został usunięty a zastępuje go nowy element o tej samej nazwie.

Filtruj dopasowane
: Podczas korzystania z filtrowania w oknie dziennika, wyszukiwane słowa są podświetlane w wynikach za pomocą tego koloru.

4.31.2. Ustawienia wykresu wersji



Rysunek 4.78. Okno dialogowe ustawień, strona Wykres wersji

Wzorce klasyfikacji
: Wykres wersji próbuje pokazać lepszy wgląd struktury repozytorium rozróżniając linię główną, gałęzie i etykiety. Ponieważ nie ma takiej klasyfikacji wbudowanej w Subversion, informacje te pochodzą z nazw

ścieżek. Ustawienia domyślne zakładają, że używacie tradycyjnych angielskich nazw, jak zaproponowano w dokumentacji Subversion, ale oczywiście wasze zwyczaje nazewnicze mogą się różnić.

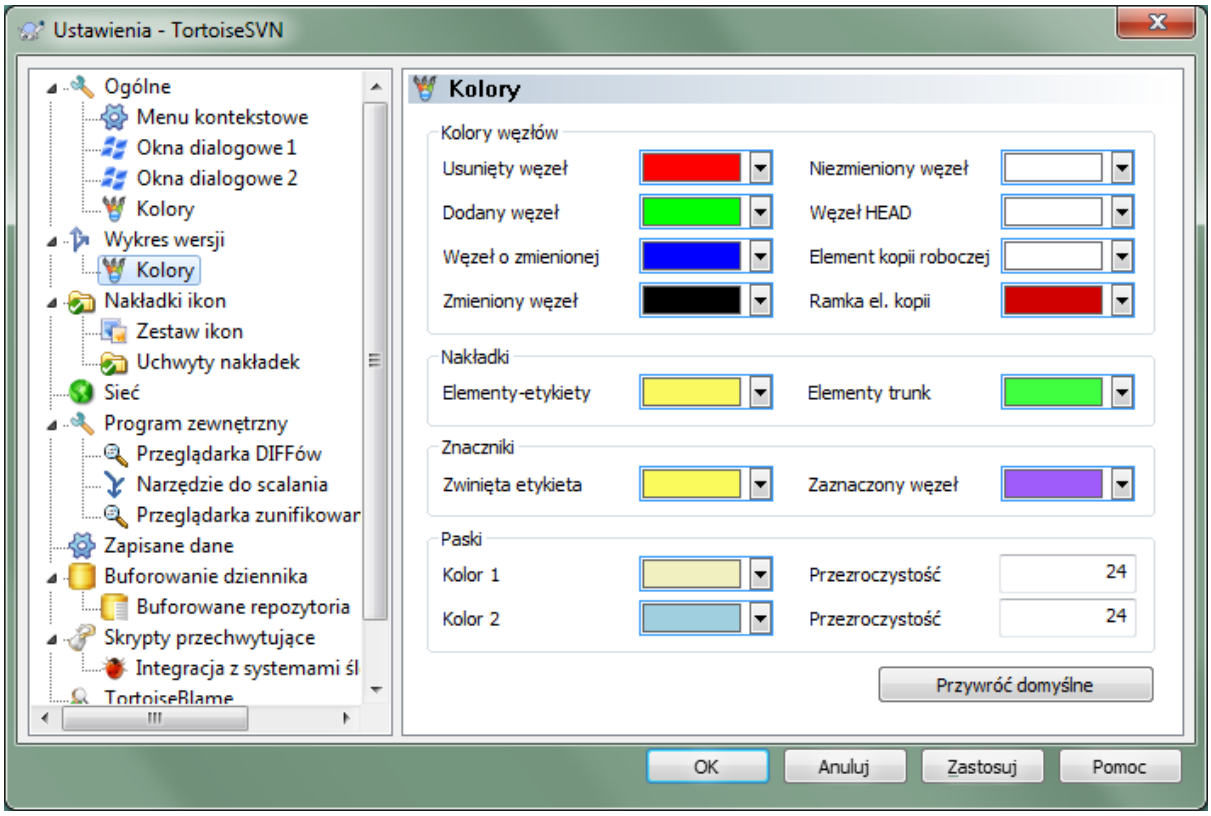
Wzorce stosowane do rozpoznania tych ścieżek należy podać w trzech polach. Wzorce zostaną dopasowane bez rozróżnienia wielkości znaków, ale należy wpisać je małymi literami. Symbole wieloznaczne `*` i `?` będą działać w zwykły sposób, można także użyć `;`, aby oddzielić kolejne wzorce. Nie należy wprowadzać żadnych dodatkowych spacji, jako że zostaną one uwzględnione w specyfikacji dopasowania.

Wykrywanie zatwierdzenia do etykiety

Należy zauważyć, że wzorce te są również używane do wykrywanie zatwierdzeń na etykietach, nie tylko na grafie wersji.

Modyfikacja kolorów
: Kolory są używane na wykresie wersji aby wskazać typ węzła, tzn. czy węzeł jest dodany, usuwany, zmieniony. Aby pomóc uwydatnić klasyfikację węzła, można zezwolić na mieszanie kolorów na wykresie, co pozwoli wskazywać zarówno typ węzła jak i klasyfikację. Jeśli pole jest zaznaczone, jest stosowane mieszanie. Jeśli pole nie jest zaznaczone, kolor jest używany do wskazania typu węzła. Użyjcie okna wyboru koloru bu przydzielić określenia używanym kolorom.

4.31.2.1. Kolory wykresu wersji



Rysunek 4.79. Okno dialogowe ustawień, strona Wykres wersji - Kolory

Ta strona umożliwia konfigurowanie używanych kolorów. Należy pamiętać, że kolor określany tutaj to kolor wypełnienia. Większość węzłów jest w kolorze mieszanki typu węzła, koloru tła i ewentualnie koloru klasyfikacji.

Usunięty węzeł
: Elementy, które zostały usunięte i nie skopiowane nigdzie indziej w tej samej wersji.

Dodany węzeł
: Elementy nowododane lub skopiowane (dodanie z historią).

Węzeł o zmienionej nazwie
: Elementy usunięte z jednej lokalizacji i dodane w innej w tej samej wersji.

Zmieniony węzełWęzeł zmodyfikowany
: Proste modyfikacje bez dodawania ani usunięcia.

Niezmieniony węzeł
: Może być używany, aby pokazać wersję służącą jako źródło kopii, nawet jeśli nie ma zmian (dla rysowanego elementu) wykonanych w tej wersji.

Węzeł HEAD
: Bieżąca wersja HEAD w repozytorium.

Węzeł KR
: Jeśli zdecydujecie się pokazać dodatkowy węzeł dla zmodyfikowanej kopii roboczej, dołączony do jego ostatnio-zatwierdzonej wersji na wykresie, zostanie użyty ten kolor.

Ramka el. kopii roboczej
: Jeśli zdecydujecie się pokazać, czy kopia robocza jest modyfikowana, zostanie użyty ten kolor obrysu na węźle KR gdy znaleziono zmiany.

Elementy-etykiety
: Węzły klasyfikowane jako etykiety mogą być mieszane z tym kolorem.

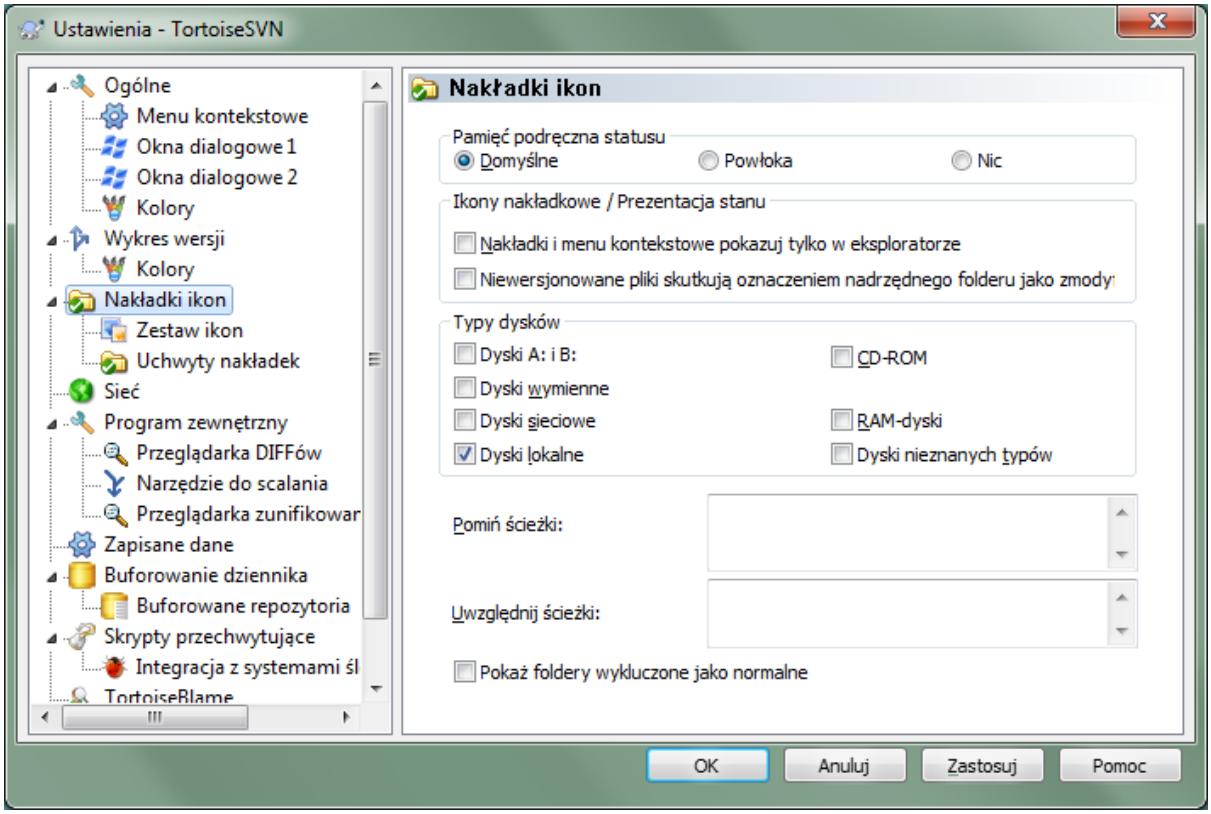
Elementy trunk
: Węzły klasyfikowane jako linie główne mogą być mieszane z tym kolorem.

Znaczniki zwiniętych etykiet
: Jeśli używacie zwiniętych etykiet aby zaoszczędzić miejsca, tagi są zaznaczone na źródle kopii przy bloku w tym kolorze.

Znaczniki zaznaczonego węzła
: Kiedy klikniecie lewym przyciskiem na węźle, aby go wybrać, znacznik używany do wskazania wyboru to blok w tym kolorze.

Paski
: Kolory te są stosowane, gdy wykres jest podzielony na mniejsze drzewa i tło stanowią kolorowe naprzemienne pasy, aby uwydatnić oddzielne drzewa.

4.31.3. Ustawienia nakładek ikon



Rysunek 4.80. Okno dialogowe ustawień, strona Nakładki ikon

Ta strona pozwala na wybór elementów, na których TortoiseSVN wyświetli nakładki ikon.

Ponieważ sporo czasu zajmuje pobranie statusu kopii roboczej, TortoiseSVN używa bufora do przechowywania stanu stąd eksplorator nie muli się podczas wyświetlania nakładek. Możecie wybrać, który typ bufora TortoiseSVN należy używać zgodnie z możliwościami systemu i rozmiarem kopii roboczej:

Domyślny
: Buforuje wszystkie istotne informacje w oddzielnym procesie (`TSVNCache.exe`). Proces przegląda wszystkie dyski w poszukiwaniu zmian i pobiera ponownie status, jeśli pliki znajdujące się w kopii roboczej zostaną zmodyfikowane. Proces wykonuje się z najniższym możliwym priorytetem, by inne programy nie przymulały się z jego powodu. Oznacza to również, że informacja o statusie nie jest w *czasie rzeczywistym*, ale może zająć kilka sekund, zanim nakładki zmienią się.

  Zaleta: nakładki pokazują stan rekursywnie, tzn. jeśli plik w głębi kopii roboczej został modyfikowany, wszystkie foldery do głównego katalogu kopii roboczej dostaną także nakładki zmodyfikowania. A ponieważ proces może wysyłać powiadomienia do powłoki, nakładki na lewym widoku drzewa często również się zmieniają.

  Wada: proces działa stale, nawet jeśli nie pracujesz nad projektami. Oprócz tego, zużywa około 10-50 MB pamięci RAM w zależności od liczby i wielkości kopii roboczych.

Powłoka
: Buforowanie jest wykonywane bezpośrednio wewnątrz rozszerzenia dll powłoki, ale tylko dla aktualnie widocznego folderu. Za każdym razem gdy przechodzisz do innego folderu, informacja o statusie jest wczytywana ponownie.

  Zaleta: potrzebuje tylko niewielkiej ilości pamięci (około 1 MB pamięci RAM) i może pokazywać stan w *czasie rzeczywistym*.

Wada: Ponieważ tylko jeden folder jest buforowany, nakładki nie pokazują stanu rekurencyjnie. Dla dużych kopii roboczych, może zająć więcej czasu pokazanie folderu w Eksploratorze niż z domyślnym buforem. Również kolumna typ MIME nie jest dostępna.

Nic
: Przy tym ustawieniu TortoiseSVN nie pobieramy w ogóle stanu w eksploratorze. W związku z tym, pliki nie dostają nakładek a foldery dostają tylko nakładkę 'zwykły', jeśli są wersjonowane. Żadne inne nakładki nie są wyświetlane, ani nie są dostępne dodatkowe kolumny.

  Zalety: żadnego używa dodatkowej pamięci i brak spowolnienia eksploratora podczas przeglądania.

  Wada: Informacja o stanie plików i folderów nie jest wyświetlana w eksploratorze Windows. Aby sprawdzić, czy kopie robocze są zmodyfikowane, trzeba użyć okna „Sprawdź zmiany”.

Domyślnie, ikony nakładek oraz menu kontekstowe pojawi się we wszystkich oknach dialogowych otwórz/zapisz, jak również w eksploratorze Windows. Jeśli mają być wyświetlane *tylko* w eksploratorze Windows, należy zaznaczyć pole wyboru Nakładki i menu kontekstowe pokazuj tylko w eksploratorze.

Możecie wymusić typ podręcznej pamięci statusu *Brak* dla podwyższonych procesów przez zaznaczenie pola Zablokuj pamięć podręczną statusu dla procesów podwyższonych. Jest to poprawne jeśli chce się zapobiec utworzeniu innego procesu `TSVNCache.exe` z podwyższonymi uprawnieniami.

Możecie także wybrać, aby zaznaczyć foldery jako zmienione jeśli zawierają elementy bez kontroli wersji. Może to być przydatne dla przypomnienia, że utworzono nowe pliki, które nie są jeszcze wprowadzone do kontroli wersji. Ta opcja jest dostępna tylko w przypadku korzystania z *domyślnej* opcji buforowania statusu (patrz poniżej).

Jeśli macie pliki na liście zmian `ignore-on-commit`, możecie sprawić by te pliki nie rozgłaszały swego stanu na folder nadrzędny. W ten sposób gdy zmienione zostały tylko pliki na tej liście zmian, folder nadrzędny pokazuje wciąż nakładkę ikony "niezmodyfikowane".

Kolejna grupa pozwala wybrać, które klasy składowania powinny pokazać nakładki. Domyślnie, tylko twarde dyski są zaznaczone. Można nawet wyłączyć wszystkie nakładki ikon, ale gdzie w tym zabawa?

Dyski sieciowe mogą być bardzo wolne, więc domyślnie ikony nie są wyświetlane na kopiach roboczych znajdujących się na udziałach sieciowych.

Pamięci flash USB wydają się być szczególnym przypadkiem gdzie typ napędu jest identyfikowany przez samo urządzenie. Niektóre pojawiają się jako dyski stałe, a inne jako dyski wymienne.

Pomiń ścieżki jest wykorzystywana do wskazania TortoiseSVN tych ścieżek, dla których *nie* powinien pokazywać nakładek ikon show i kolumny statusu. Jest to przydatne, jeśli macie jakieś bardzo duże kopie robocze zawierające tylko biblioteki, w których nie przewiduje się zupełnie zmian i dlatego nie potrzebują nakładek, lub jeśli chcecie by TortoiseSVN wyszukiwał tylko w określonych folderach.

Każda ścieżka wpisana tutaj ma jest stosowana rekurencyjnie, więc żaden z folderów podrzędnych również nie pokaże nakładek. Jeśli chcecie pominąć *tylko* tak nazwany folder, dołączcie `?` po zapisie ścieżki.

To samo dotyczy Uwzględnij ścieżki. Tyle tylko, że dla tych ścieżek nakładki są wyświetlane, nawet jeśli nakładki są wyłączone dla określonego typu napędu lub przez pominięcia ścieżki określone powyżej.

Użytkownicy często pytają, jak te trzy ustawienia oddziałują wzajemnie na siebie. Dla każdej ścieżki sprawdza się listy uwzględnienia i pominięcia, szukając w górę struktury katalogów do znalezienia pasującego. Kiedy znajdzie się pierwszy dopasowany, wypełnia się regułę uwzględnienia albo pominięcia. Jeśli istnieje konflikt, pojedynczy katalog ma pierwszeństwo przed zastosowaniem rekurencji, następnie uwzględnienie ma pierwszeństwo przed pominięciem.

Powinien tu pomóc przykład:

```
Pomiń: C: C:\develop\?
C:\develop\tsvn\obj
C:\develop\tsvn\bin
Uwzględnij:
C:\develop
```

Te ustawienia wyłączają nakładki ikon z dysku C:, z wyjątkiem `c:\develop\`. Wszystkie projekty wewnątrz tego katalogu pokazują nakładki, z wyjątkiem samego folderu `c:\develop`, który jest specjalnie ignorowany. Szybkozmienne binarne foldery są również wykluczone.

TSVNCache.exe korzysta także z tych ścieżek w celu ograniczenia obszaru skanowania. Jeśli chcecie by szukał tylko w poszczególnych folderach, wyłączyć wszystkie typy dysków i dołączyć tylko foldery dla których użytkownik chce skanowania.

Pomiń napędy `SUBST`

Często wygodne jest użycie dysku `SUBST` dla dostępu do kopii roboczej, np. za pomocą polecenia

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

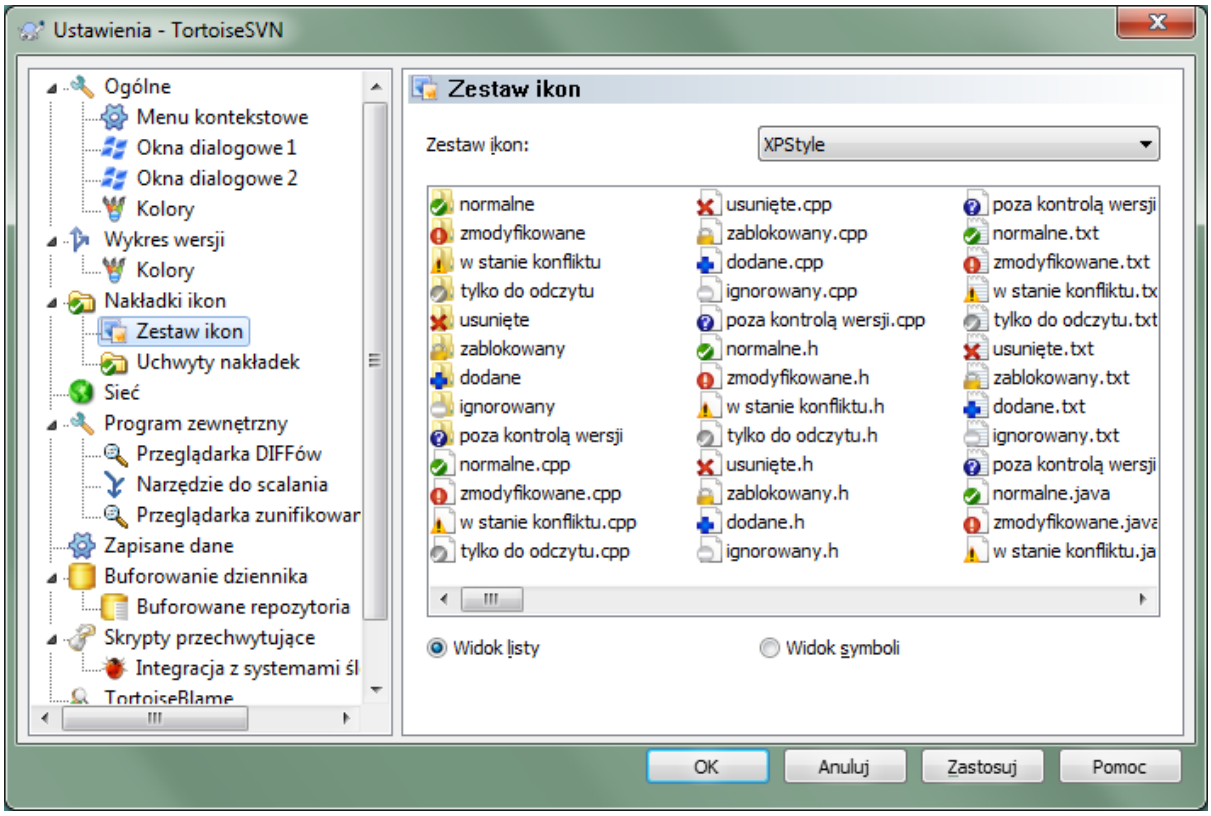
Jednak może to spowodować brak aktualizacji nakładek, gdyż `TSVNCache` otrzymuje tylko jedno powiadomienie zmiany w pliku, a jest ono zazwyczaj wykonywane dla oryginalnej ścieżki. Oznacza to, że nakładki na ścieżce `subst` mogą nie być nigdy uaktualnione.

Prostym sposobem obejścia tego jest pominięcie oryginalnej ścieżki przy pokazywaniu nakładek, aby nakładki pokazały się na ścieżce `subst`.

Czasami chcecie wykluczyć obszary, zawierające kopie robocze, co oszczędza czas TSVNCache przy skanowaniu i monitorowaniu zmian, ale nadal potrzebujecie wizualnej informacji, że folder zawiera kopię roboczą. Pole wyboru Pokaż wykluczone foldery główne jako 'zwykłe' pozwala to zrobić. Dzięki tej opcji foldery główne kopii roboczych z każdego wykluczonego obszaru (typ dysku nie jest zaznaczony, lub wskazanie wyłączenia) pojawi się jako normalny i aktualny, z zielonym znacznikiem wyboru. Jest to przypomnienie, że właśnie oglądacie kopię roboczą, chociaż nakładka foldera mogą nie być prawidłowa. Pliki nie posiadają zupełnie żadnych nakładek. Pamiętajcie, że menu kontekstowe nadal działa, chociaż nakładki nie są pokazywane.

Jako specjalny wyjątek od reguły, napędy `A:` i `B:` nigdy nie są brane pod uwagę przy zaznaczeniu opcji Pokaż foldery wykluczone jako 'normalne'. To dlatego, że Windows jest zmuszony zweryfikować napęd, a to może powodować kilkusekundowe opóźnienie po uruchomieniu eksploratora, nawet jeśli komputer nie posiada napędu dyskietek.

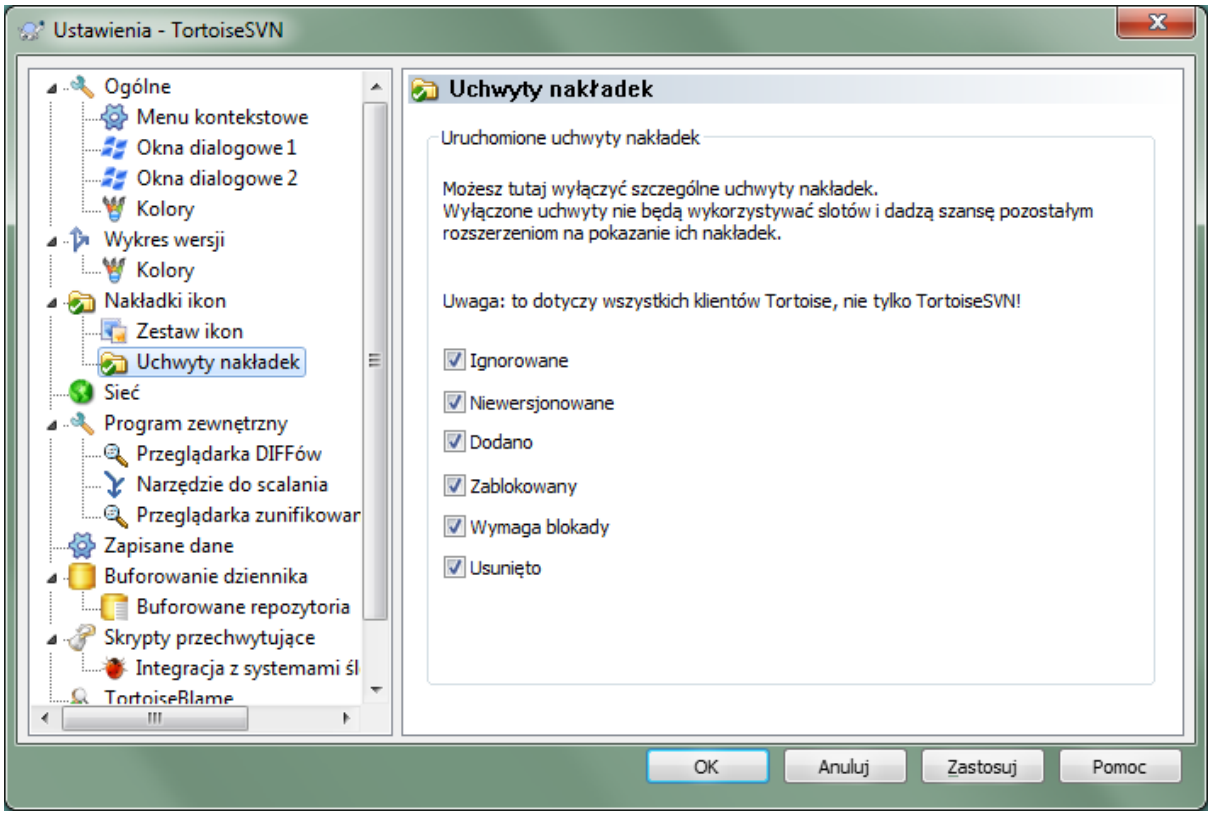
4.31.3.1. Wybór zestawu ikon



Rysunek 4.81. Okno dialogowe ustawień, strona Zestaw ikon

Można zmienić zestaw nakładek ikon na taki, który najbardziej lubicie. Należy pamiętać, że jeśli zmieniacie zestaw nakładek, może być konieczne ponowne uruchomienie komputera, aby zmiany odniosły skutek.

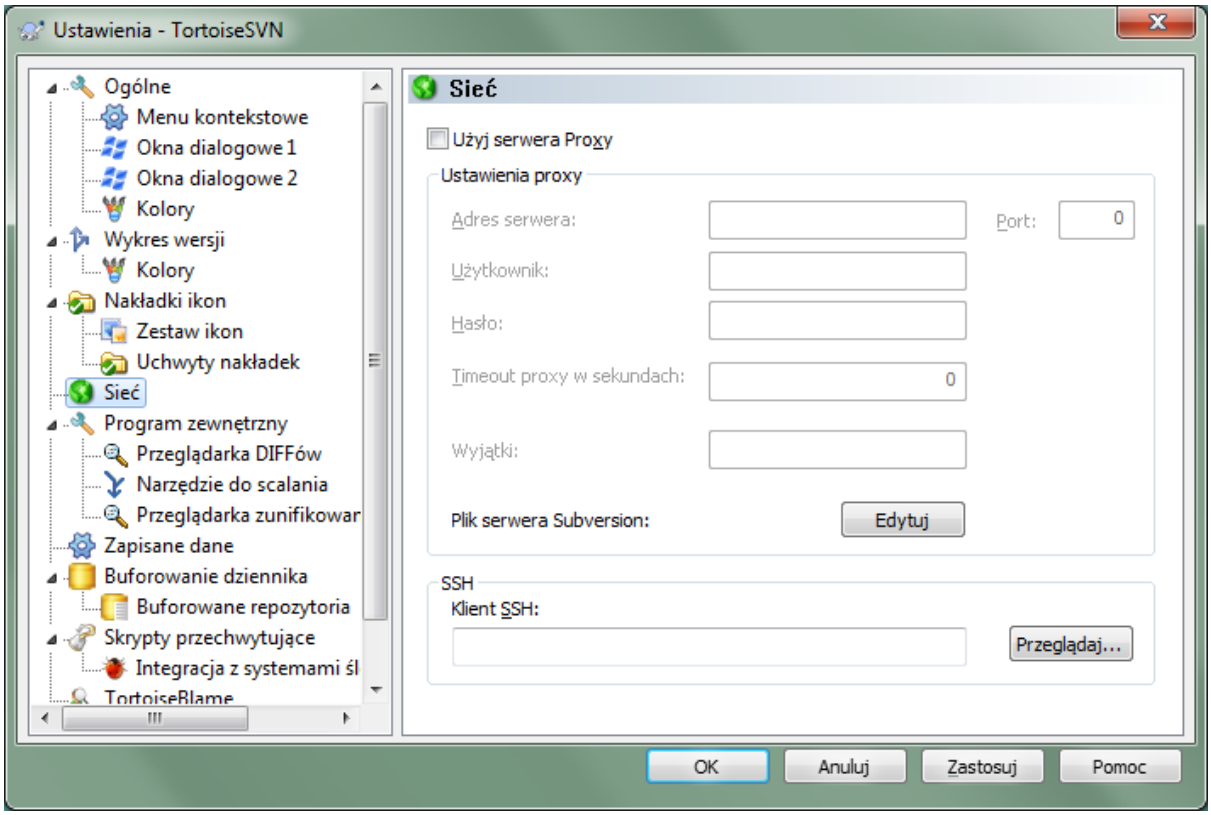
4.31.3.2. Uruchomione uchwyty nakładek



Rysunek 4.82. Okno dialogowe ustawień, strona Uchwyty nakładek

Ponieważ liczba dostępnych nakładek jest poważnie ograniczona, możecie wyłączyć niektóre uchwyty w celu zapewnienia, że te, które chcecie zostaną załadowane. Ponieważ TortoiseSVN używa wspólnych komponentów TortoiseOverlays, która jest dzielona z innymi klientami Tortoise (np. TortoiseCVS, TortoiseHg), to ustawienie dotyczy również tych klientów.

4.31.4. Ustawienia sieciowe



Rysunek 4.83. Okno dialogowe ustawień, strona Sieć

Tutaj można skonfigurować serwer proxy, jeśli zajdzie taka potrzeba, aby przedostać się przez zaporę firmy.

Jeśli potrzebne są inne ustawienia proxy dla poszczególnych repozytoriów, należy użyć pliku Subversion `servers` by to skonfigurować. Skorzystajcie z Edycja by przejść tam bezpośrednio. Sprawdźcie szczegóły na *Obszar konfiguracji środowiska wykonawczego* [http://svnbook.red-bean.com/en/1.8/svn.advanced.confarea.html] by dowiedzieć się, jak korzystać z tego pliku.

Można również określić, którego programu TortoiseSVN ma użyć, aby ustanowić bezpieczne połączenie do repozytorium svn+ssh. Zalecamy stosowanie TortoisePlink.exe. Jest to wersja popularnego programu Plink, i jest dołączona do TortoiseSVN, ale jest kompilowana jako aplikacja pozbawiona okien, więc nie dostaniecie okna DOS wyskakującego przy każdym uwierzytelnieniu.

Musicie podać pełną ścieżkę do pliku wykonywalnego. Dla TortoisePlink.exe jest to standardowy folder bin TortoiseSVN. Użyj Przeglądaj, aby móc go zlokalizować. Zauważcie, że jeśli ścieżka zawiera spacje, należy ująć ją w cudzysłów, np.

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

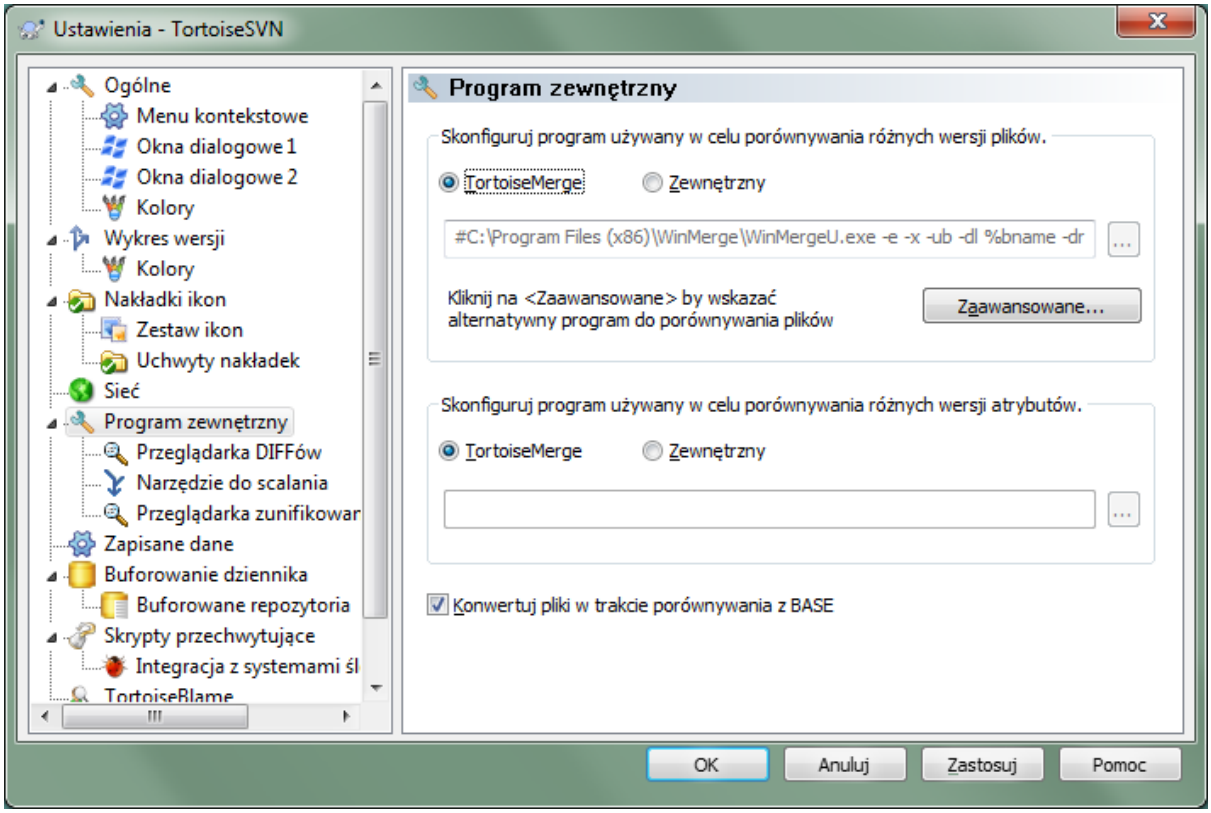
Jednym z efektów ubocznych braku okna jest to, że nie ma miejsca na komunikaty o błędach, więc jeśli uwierzytelnianie się nie powiedzie, pojawi się po prostu komunikat w stylu „Nie można pisać do standardowego wyjścia”. Z tego powodu zalecamy, aby na początku ustawić połączenie przy pomocy standardowego Plink. Kiedy wszystko działa, można użyć TortoisePlink z dokładnie takimi samymi parametrami.

TortoisePlink nie posiada żadnej własnej dokumentacji, ponieważ jest to tylko drobna poprawka Plink. Dowiedzcie się o parametrach linii poleceń ze *strony PuTTY* [https://www.chiark.greenend.org.uk/~sgtatham/putty/].

Aby uniknąć wielokrotnego monitu o hasło, można również rozważyć użycie narzędzia buforowania haseł, takich jak Pageant. Ono także dostępne jest do pobrania ze strony PuTTY.

Na koniec, ustawienie SSH na serwerze i na klientach jest procesem nietrywialnym, który pozostaje poza zakresem tego pliku pomocy. Można jednak zapoznać się z przewodnikiem w FAQ TortoiseSVN znajdującym się w *Subversion/TortoiseSVN SSH How-To* [https://tortoisesvn.net/ssh_howto.html].

4.31.5. Ustawienia programów zewnętrznych



Rysunek 4.84. Okno dialogowe ustawień, strona Przeglądarka DIFFów

Tutaj możecie zdefiniować własne narzędzie porównania/scalenia, którego powinien używać TortoiseSVN. Domyślnym ustawieniem jest korzystanie z TortoiseMerge, który jest instalowany wraz z TortoiseSVN.

Czytajcie Sekcja 4.11.6, „Zewnętrzne narzędzia porównywania/scalania”, gdzie znajduje się lista niektórych zewnętrznych programów do porównania/scalania z jakich lidzie korzystają z TortoiseSVN.

4.31.5.1. Przeglądarka diffów

Zewnętrzny program do porównywania może być używany do porównywania różnych wersji plików. Zewnętrzny program będzie musiał uzyskać nazwy plików z wiersza poleceń, wraz z innymi opcjami linii poleceń. TortoiseSVN używa podstawienia parametrów poprzedzonych %. Gdy napotkany jeden z nich, zastąpi go odpowiednią wartością. Kolejność parametrów zależy od użytego programu porównującego.

%base
: Oryginalny plik bez zmian

%bname
: Tytuł okna dla pliku bazowego

%nqbname
: Tytuł okna dla pliku bazowego, bez cudzysłowu

%mine
: Twój plik, z wprowadzonymi zmianami

%yname
Tytuł okna dla twojego pliku

%nqyname
Tytuł okna dla twojego pliku, bez cudzysłowu

%burl
Adres URL oryginalnego pliku, jeśli dostępny

%nqburl
Adres URL oryginalnego pliku, jeśli dostępny, bez cudzysłowu

%yurl
Adres URL drugiego pliku, jeśli dostępny

%nqyurl
Adres URL drugiego pliku, jeśli dostępny, bez cudzysłowu

%brev
Wersja oryginalnego pliku, jeśli dostępna

%nqbrev
Wersja oryginalnego pliku, jeśli dostępna, bez cudzysłowu

%yrev
Wersja drugiego pliku, jeśli dostępna

%nqyrev
Wersja drugiego pliku, jeśli dostępna, bez cudzysłowu

%peg
Wersja wieszakowa, jeśli dostępna

%nqpeg
Wersja wieszakowa, jeśli dostępna, bez cudzysłowu

%fname
Nazwa pliku. Pusty ciąg znaku w przypadku porównywania dwóch plików zamiast dwóch stanów tego samego pliku.

%nqfname
Nazwa pliku, bez cudzysłowu

Tytuły okien nie są czystymi nazwami plików. TortoiseSVN traktuje je jako nazwy do wyświetlenia i odpowiednio tworzy nazwy. Tak więc np. jeśli robicie porównanie pliku w wersji 123 z plikiem w kopii roboczej, nazwami będą `nazwa_pliku : wersja 123` i `nazwa_pliku : kopia robocza`.

Przykład, dla ExamDiff Pro:

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

lub dla KDiff3:

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

lub dla WinMerge:

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

lub dla Araxis:

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

lub dla UltraCompare:

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

lub dla DiffMerge:

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

Jeśli używacie atrybutu `svn:keywords` by rozszerzyć słowa kluczowe, a w szczególności *wersję* z pliku, może wystąpić różnica między plikami, która wynika wyłącznie z obecności bieżącej wartości słowa kluczowego. Również w przypadku korzystania z `svn:eol-style = native` w pliku BASE będą czyste zakończenia linii `LF` natomiast Wasz plik będzie miał zakończenia `CR-LF`. TortoiseSVN zwykle ukrywa te różnice automatycznie przez początkową analizę pliku BASE z rozszerzeniem słów kluczowych i zakończeń linii przed wykonaniem operacji porównania. Jednakże może to zająć dużo czasu w przypadku dużych plików. Jeśli Konwertuj pliki w trakcie porównywania z BASE nie jest zaznaczone, TortoiseSVN pominie wstępne przetwarzanie plików.

Możecie także określić inne narzędzia porównywania do wykorzystania przy pomocy atrybutów Subversion. O ile zwykle bywają to krótkie, proste ciągi znaków, możecie chcieć użyć prostszej i bardziej kompaktowej przeglądarki.

Jeśli skonfigurowano alternatywne narzędzie porównywania, można uzyskać dostęp do TortoiseMerge *i* narzędzia firm trzecich z menu kontekstowego. Menu kontekstowe → Porównaj wykorzystuje podstawowe narzędzie porównania, a **SHIFT**+Menu kontekstowe → Porównaj używa drugorzędnego narzędzia porównania.

Na dole okna dialogowego programu do przeglądania plików różnicowych unified-diff (pliki poprawek). Nie są wymagane parametry. Ustawienie Domyślny oznacza użycie TortoiseUDiff, który jest instalowany wraz z TortoiseSVN oraz kodów kolorów dodanych i usuniętych linii.

Ponieważ Unified Diff jest tylko formatem tekstowym, można użyć ulubionego edytora tekstu, jeśli wolicie.

4.31.5.2. Narzędzie do scalania

Zewnętrzny program scalania użyty do rozwiązania konfliktu plików. Podstawianie parametrów jest używane w taki sam sposób jak w programie porównującym.

%base
: oryginalny plik bez zmian własnych ani cudzych

%bname
: Tytuł okna dla pliku bazowego

%nqbname
: Tytuł okna dla pliku bazowego, bez cudzysłowu

%mine
: Wasz plik, z wprowadzonymi zmianami

%yname
: Tytuł okna dla twojego pliku

%nqyname
: Tytuł okna dla twojego pliku, bez cudzysłowu

%theirs
plik jaki znajduje się w repozytorium

%tname
Tytuł okna dla pliku z repozytorium

%nqtname
Tytuł okna dla pliku z repozytorium, bez cudzysłowu

%merged
plik w stanie konfliktu, wynik operacji scalenia

%mname
Tytuł okna dla scalonego pliku

%nqmname
Tytuł okna dla scalonego pliku, bez cudzysłowu

%fname
Nazwa pliku z konfliktem

%nqfname
Nazwa pliku z konfliktem, bez cudzysłowu

Przykład, dla Perforce Merge:

```
C:\Sciezka-Do\P4Merge.exe %base %theirs %mine %merged
```

lub dla KDiff3:

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

lub dla Araxis:

```
C:\Sciezka-Do\compare.exe /max /wait /3 /title1:%tname /title2:%bname
/title3:%yname %theirs %base %mine %merged /a2
```

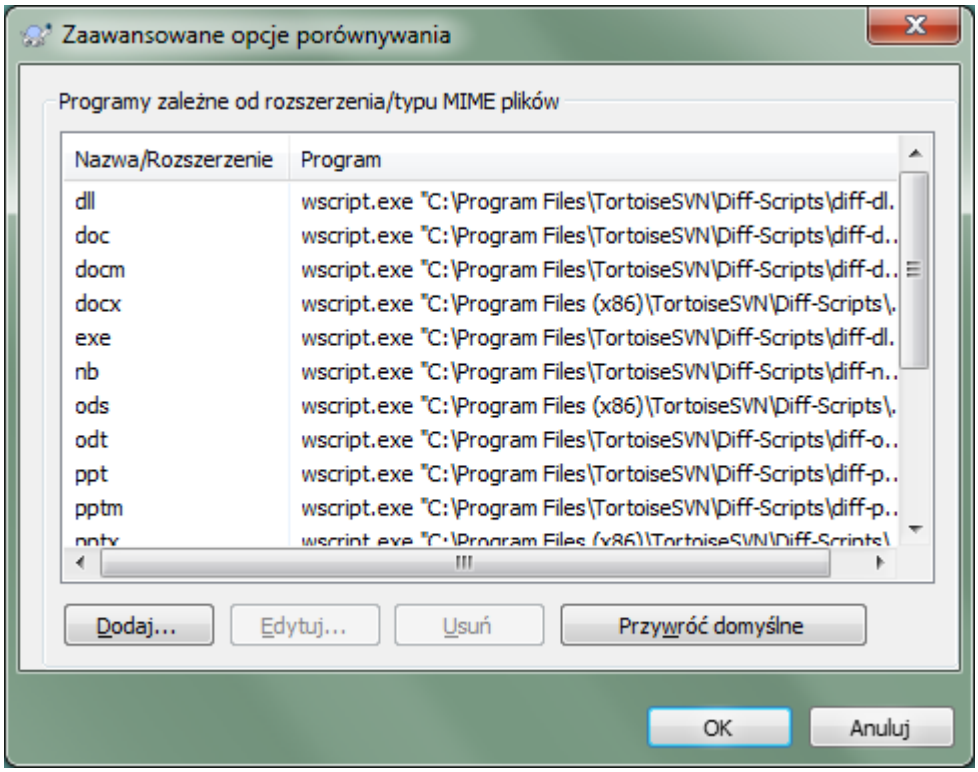
lub dla WinMerge (2.8 or later):

```
C:\Sciezka-Do\WinMerge.exe %merged
```

lub dla DiffMerge:

```
C:\Sciezka-Do\DiffMerge.exe -caption=%mname -result=%merged -merge -nosplash -t1=%yname
```

4.31.5.3. Zaawansowane ustawienia porównywania/scalania

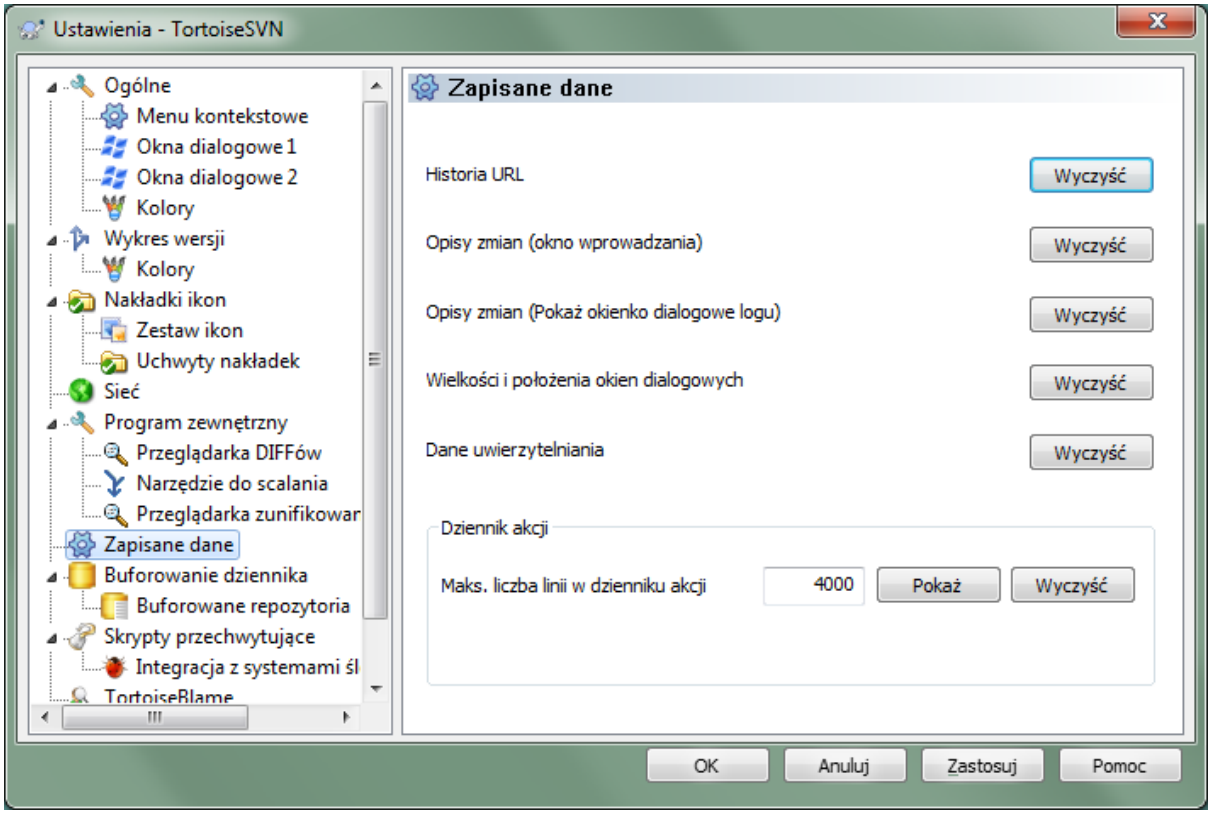


Rysunek 4.85. Okno dialogowe ustawień, okno Zaawansowane opcje porównywania/scalania

W zaawansowanych ustawieniach można zdefiniować różne programy porównywania i scalania dla każdego rozszerzenia pliku. Na przykład można skojarzyć Photoshop jako program „porównania” dla plików `.jpg` :-) Można także powiązać atrybut `svn:mime-type` z programem porównywania i scalania.

Aby skojarzyć za pomocą rozszerzenia pliku, musicie podać rozszerzenie. Użyjcie `.bmp` do opisania pliki map bitowych Windows. Aby skojarzyć za pomocą atrybutu `svn:mime-type`, określcie typ MIME, w tym ukośnik, na przykład `text/xml`.

4.31.6. Ustawienia zapisanych danych



Rysunek 4.86. Okno dialogowe ustawień, strona Zapisane dane

Dla Waszej wygody, TortoiseSVN zapisuje wiele używanych ustawień i pamięta, gdzie byliście ostatnio. Jeśli chcecie wyczyścić ten bufor danych, możecie to zrobić tutaj.

Historia URL
: Gdy pobieracie kopię roboczą, scalacie zmiany lub korzystacie z przeglądarki repozytorium, TortoiseSVN prowadzi rejestr ostatnio używanych adresów URL i podpowiada je w menu rozwijalnym. Czasem lista ta jest zaśmiecona nieaktualnymi adresami więc warto przeczyścić ją okresowo.

  Jeśli chcecie usunąć pojedynczy element z jednej z list rozwijalnych można to zrobić na miejscu. Wystarczy kliknąć na strzałkę by otworzyć menu rozwijalne, przesunąć kursor myszy nad element, który chcecie usunąć i wcisnąć **Shift+Del**.

Opisy zmian (okno wprowadzania)
: TortoiseSVN przechowuje ostatnie opisy zmian z zatwierdzeń, który zostały wprowadzone. Są one przechowywane dla repozytorium, więc jeśli łączycie się do wielu repozytoriów lista ta może się pokaźnie rozwinąć.

Opisy zmian (Pokaż okienko dialogowe logu)
: TortoiseSVN buforuje opisy zmian wczytywane przez okno Pokaż dziennik, aby zaoszczędzić czas przy następnym wyświetleniu dziennika. Jeśli ktoś edytuje opis zmiany a ten opis jest już w buforze, nie będzie widać zmiany do chwili wyczyszczenia bufora. Buforowanie opisów zmian włącza się na karcie Bufor dziennika.

Wielkości i położenia okien dialogowych
: Wiele okien dialogowych zapamiętuje rozmiar i pozycję, w której zostały ostatnio użyte.

Dane uwierzytelniania
: Podczas uwierzytelniania wobec serwera Subversion, nazwa użytkownika i hasło są buforowane lokalnie, więc nie trzeba wpisywać ich za każdym razem. Możecie usunąć je ze względów bezpieczeństwa, lub dlatego,

że chcecie uzyskać dostęp do repozytorium pod inną nazwą użytkownika ... czy Andrzej wie, że używasz jego komputera?

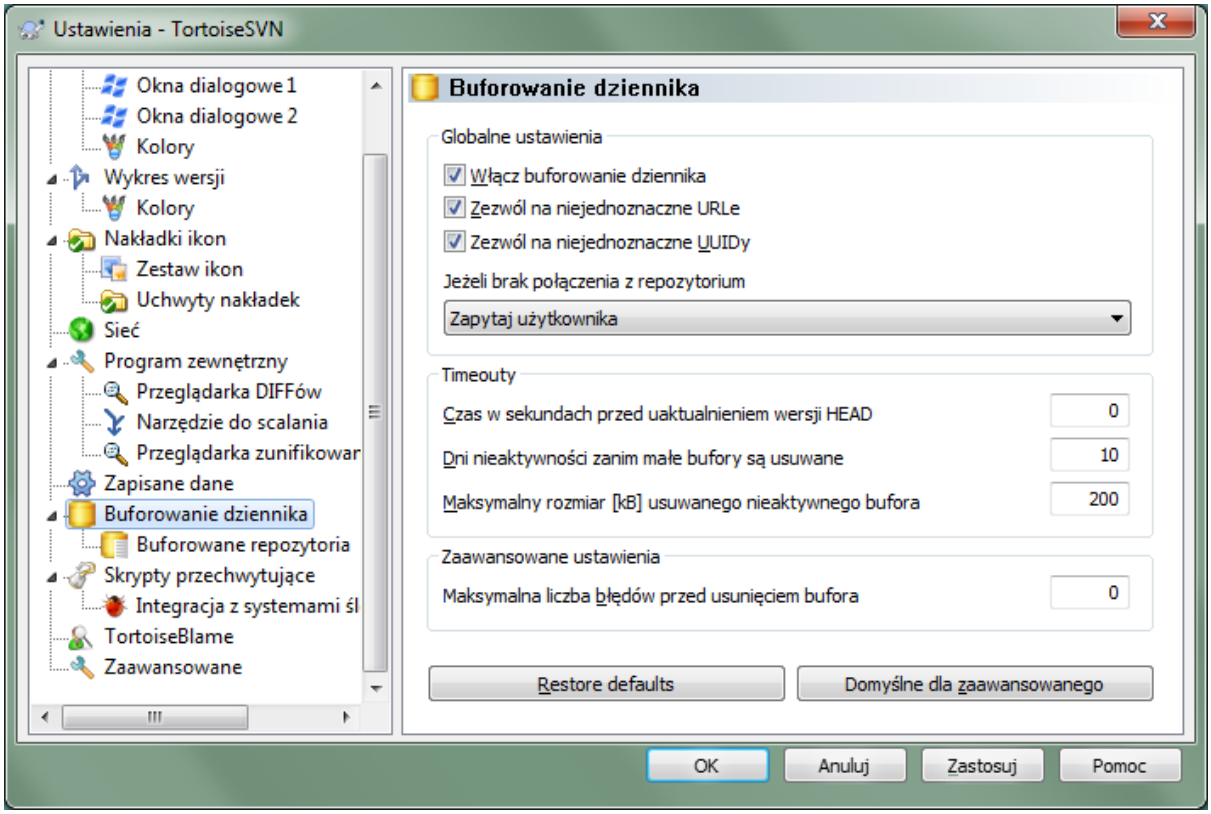
Jeśli chcesz wyczyścić dane autentykacyjne tylko dla jednego serwera, użyj przycisku Wyczyść... zamiast Wyczyść wszystko.

Dziennik akcji
TortoiseSVN prowadzi dziennik wszystkiego, co napisano na jego dialogach postępu. Może to być przydatne, gdy na przykład chcecie sprawdzić, co się stało podczas ostatnich poleceń aktualizacji.

Plik dziennika ma ograniczoną długość, a gdy rozrośnie się zbyt bardzo, najstarsze treści zostaną odrzucone. Domyślnie przechowywane jest 4000 linii, ale można dostosować tę liczbę.

Stąd można przeglądać zawartość pliku dziennika, a także go wyczyścić.

4.31.7. Buforowanie dziennika



Rysunek 4.87. Okno dialogowe ustawień, strona Bufor Dziennika

To okno pozwala na skonfigurowanie funkcję buforowania dziennika z TortoiseSVN, który utrzymuje lokalną kopię opisów zmian i zmienionych ścieżek by uniknąć czasochłonnych wczytań danych z serwera. Korzystanie z bufora dziennika może znacznie przyspieszyć okno dziennika oraz wykres wersji. Kolejną przydatną cechą jest to, że opisy zmian są nadal dostępne w trybie offline.

Włącz buforowanie dziennika
Umożliwia buforowanie danych dziennika, gdy jest wymagane. Jeśli jest zaznaczona, dane będą pobierane z bufora, gdy dostępne, a wszystkie wiadomości w buforze spoza bufora zostaną pobrane z serwera i dodane do pamięci podręcznej.

Jeżeli buforowanie jest wyłączone, dane zawsze będą pobierane bezpośrednio z serwera, a nie przechowywane lokalnie.

Zezwól na niejednoznaczne URLe
: Czasami może być potrzebne połączenie z serwerem, które używa tego samego adresu URL dla wszystkich repozytoriów. Starsze wersje `svnbridge` poradziłyby sobie z tym. Jeśli potrzebny jest dostęp do takich repozytoriów trzeba zaznaczyć tę opcję. Jeśli nie, lepiej zostawić to pole puste w celu zwiększenia wydajności.

Zezwól na niejednoznaczne
: Niektóre usługi hostingowe nadają wszystkim repozytoriom ten sam UUID. Można nawet zrobić to samemu, kopiując folder repozytorium, aby utworzyć nowe. Z różnych powodów jest to zły pomysł - UUID musi być *unikalny*. Jednak bufor dziennika będzie nadal działać w tej sytuacji, jeśli zaznaczycie to pole. Jeśli nie potrzebujecie, zostawcie to pole niezaznaczone dla poprawy wydajności.

Jeśli nie można połączyć się z repozytorium
: Jeśli pracujecie w trybie offline, lub serwer repozytorium jest wyłączony, bufor dziennika może być nadal używany do dostarczania opisów zmian znajdujących się już w buforze. Oczywiście pamięć podręczna może nie być na bieżąco, występują więc opcje sterujące, czy funkcja powinna być używana.

  Kiedy dane dziennika są pobierane z bufora bez kontaktu z serwerem, okno dialogowe używające tych wiadomości pokaże stan offline na pasku tytułu.

Czas przed uaktualnieniem wersji HEAD
: Po wywołaniu okna dziennika należy zwykle połączyć się z serwerem w celu sprawdzenia ewentualnych nowych opisów zmian. Jeśli omawiany limit czasu jest różny od zera, kontakt z serwerem zostanie nawiązany tylko wtedy, gdy ten okres upłynął już od ostatniego kontaktu. Może to zmniejszyć komunikację z serwerem, jeśli często otwiera się okno dziennika a serwer jest wolny, przy czym przedstawione dane mogą nie być całkowicie aktualne. Jeśli chcecie korzystać z tej funkcji zalecamy użycie kompromisowej wartości 300 (5 minut).

Dni nieaktywności zanim małe bufory są usuwane
: Podczas przeglądania wielu repozytoriów można zgromadzić wiele buforów dziennika. Jeśli nie korzysta się z nich aktywnie, bufory nie rozrastają się zbytnio, więc TortoiseSVN czyści je domyślnie po ustalonym czasie. Użyj tej pozycji do kontroli oczyszczania bufora.

Maksymalny rozmiar usuwanych nieaktywnych buforów
: Większe pamięci podręczne są trudniejsze do odzyskania, więc TortoiseSVN czyści tylko małe bufory. Dostrajanie progu wykonuje się tą wartością.

Maksymalna liczba błędów przed usunięciem bufora
: Czasami coś pójdzie nie tak z buforowaniem i spowoduje to awarię. W takim przypadku pamięć podręczna jest zwykle automatycznie usuwana, aby zapobiec powtórzeniu się problemu. Jeśli używacie mniej stabilnego nocnego wydania testowego, możecie zdecydować się zachować bufor mimo wszystko.

4.31.7.1. Buforowane repozytoria

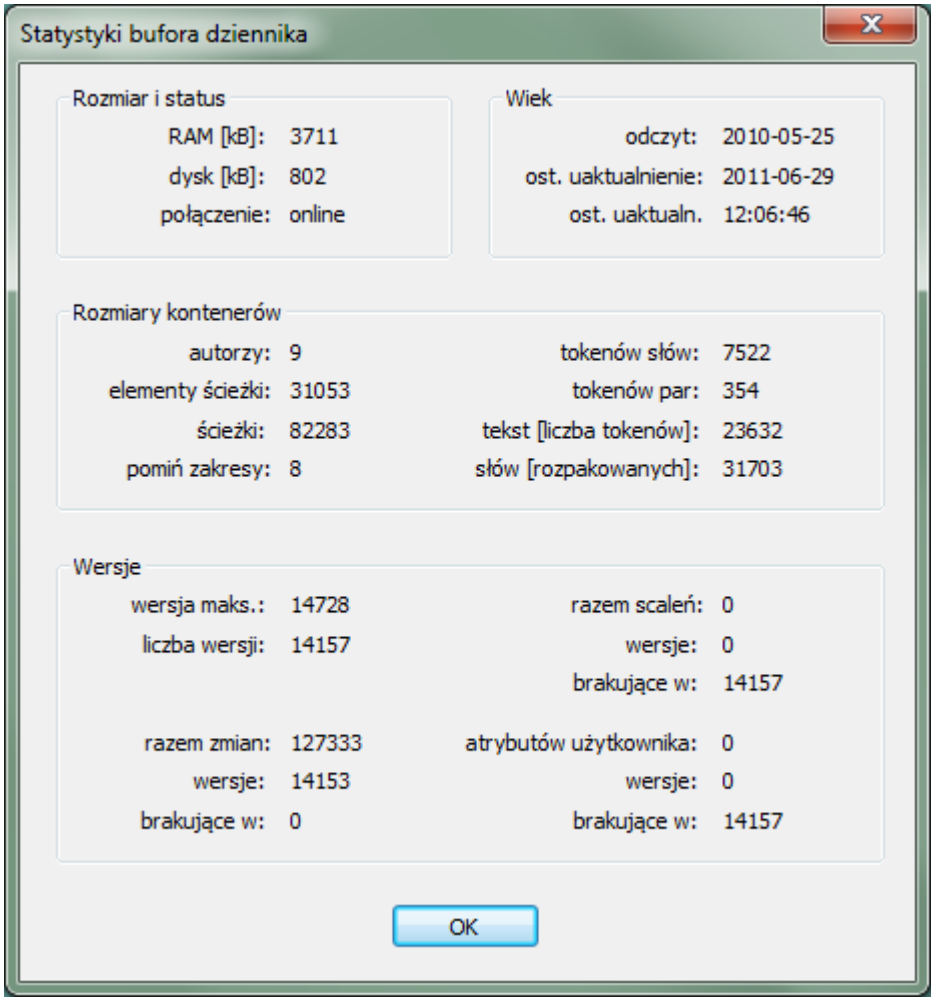
Na tej stronie można zobaczyć listę repozytoriów, które są przechowywane w buforze lokalnym, oraz wielkość używanego bufora. Po wybraniu jednego z tych repozytoriów można użyć przycisków opisanych poniżej.

Kliknijcie na Uaktualnij aby całkowicie odświeżyć bufor i wypełnić wszystkie luki. Dla dużego repozytorium może to być bardzo czasochłonne, ale przydatne, jeśli macie zamiar przejść do trybu offline i chcecie mieć bufor najpełniej odwzorowujący stan serwera.

Kliknijcie na eksport aby wyeksportować cały bufor jako zbiór plików CSV. Może to być przydatne, jeśli chcecie do przetwarzać opisy zmian przy użyciu programu zewnętrznego, mimo że jest przydatne głównie dla programistów.

Kliknijcie na Usuń aby usunąć wszystkie buforowane dane dla wybranych repozytoriów. Nie wyłącza to buforowania dla repozytorium więc wraz z następnym żądaniem danych z dziennika, zostanie utworzony nowy bufor.

4.31.7.2. Statystyki bufora dziennika



Rysunek 4.88. Okno dialogowe ustawień, Statystyki bufora dziennika

Kliknijcie przycisk Szczegóły aby zobaczyć szczegółowe statystyki dla danego bufora. Wiele pól pokazanych tutaj jest w obszarze zainteresowań głównie twórców TortoiseSVN, więc nie wszystkie są szczegółowo opisane.

RAM
: Wielkość pamięci potrzebnej do obsługi tego bufora.

Dysk
: Ilość miejsca na dysku zużywanego na bufor. Dane są kompresowane, więc wykorzystanie dysku jest zwykle w miarę skromne.

Połączenie
: Pokazuje, czy repozytorium było dostępne podczas ostatniego użycia bufora.

Ostatnie uaktualnienie
: Czas ostatniej zmiany zawartości bufora.

Ostatnie uaktualnienie HEAD
: Czas ostatniego żądania wersji HEAD z serwera.

Autorzy
: Liczba różnych autorów z wiadomości zapisanych w buforze.

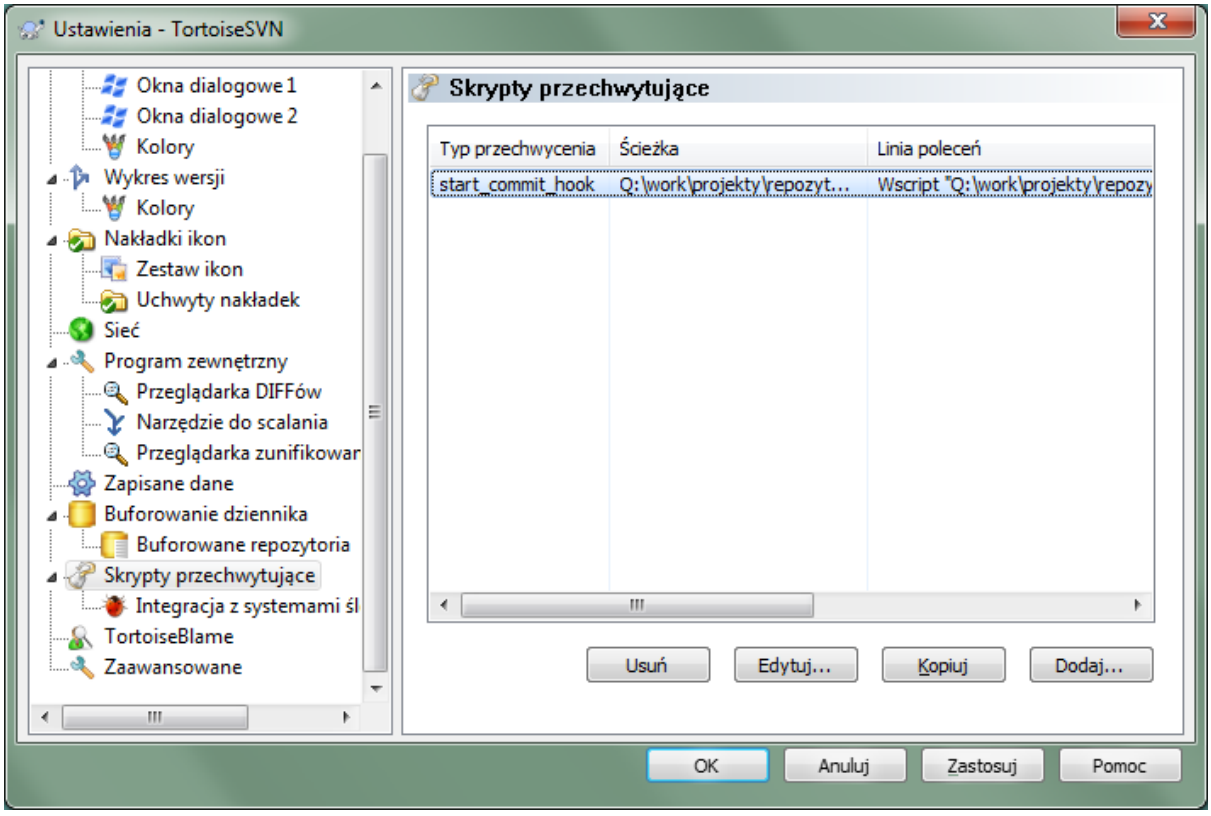
Ścieżki
: Liczba ścieżek na liście, jak to można zobaczyć za pomocą `svn log -v`.

Pomiń zakresy
: Liczba zakresów wersji, które nie zostały wczytane tylko dlatego, że nie były wymagane. Jest to miara liczby luk w buforze.

Wersja maks.
: Najwyższy numer wersji zapisany w buforze.

Liczba wersji
: Liczba wersji w pamięci podręcznej. Jest to kolejny miernik kompletności bufora.

4.31.8. Skrypty przechwytujące po stronie klienta

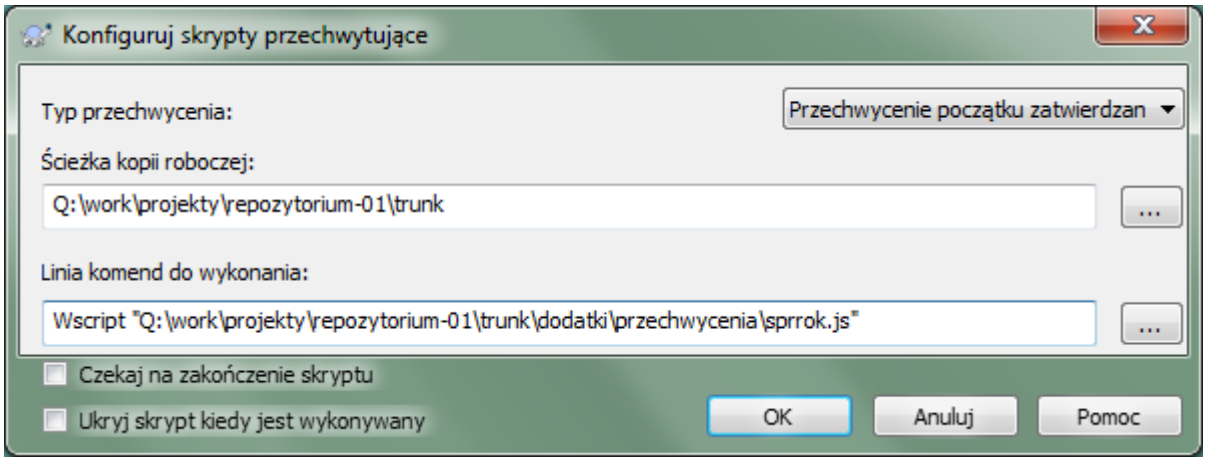


Rysunek 4.89. Okno dialogowe ustawień, strona Skrypty przechwytujące

To okno dialogowe pozwala skonfigurować skrypty przechwytujące, które zostaną wykonane automatycznie, gdy wykonywane są pewne działania Subversion. W przeciwieństwie do skryptów przechwytujących opisanych w Sekcja 3.3, „Skrypty przechwytujące po stronie serwera”, te skrypty są wykonywane lokalnie na komputerze klienta.

Jedna aplikacja dla takich przechwyceń może służyć do wywołania programu jak `SubWCRev.exe`, aby zaktualizować numery wersji po zatwierdzeniu, a może wyzwolić rekompilację.

Należy zauważyć, że można wskazać takie skrypty przechwytujące przy użyciu atrybutów z kopii roboczej. By dowiedzieć się więcej przejrzyjcie sekcję Sekcja 4.18.2, „Atrybuty projektu TortoiseSVN”.



Rysunek 4.90. Okno dialogowe ustawień, Konfiguruj skrypty przechwytujące

Aby dodać nowy skrypt, po prostu kliknij Dodaj i wypełnij szczegóły.

Obecnie dostępne są następujące rodzaje skryptów przechwytujących

Start-commit
: Wykonywany zanim zostanie wyświetlone okno dialogowe zatwierdzenia. Można go użyć jeśli przechwycenie modyfikuje wersjonowany plik i wpływa na listę plików, które muszą być zatwierdzone oraz opis zmiany. Należy jednak pamiętać, że ponieważ przechwycenie jest wywołane na wczesnym etapie, pełna lista elementów wybranych do zatwierdzenia nie jest dostępna.

Ręczne pre-zatwierdzenie
: Jeśli został zaznaczony, okno zatwierdzenia wyświetla przycisk Uruchom Przechwycenie, który po kliknięciu uruchamia wskazany skrypt przechwytujący.Skrypt przechwytujący dostaje listę wszystkich sprawdzanych plików i folderów oraz komunikat zatwierdzenia jeśli został wprowadzony.

Sprawdzenie zatwierdzenia
: Wywoływany gdy użytkownik naciśnie OK na oknie zatwierdzenia a przed zamknięciem okna. To przechwycenie pobiera listę wszystkich zaznaczonych plików. Jeśli przechwycenie zwróci błąd, okno zatwierdzenia pozostanie otwarte.

  Jeśli zwrócony błąd zawiera ścieżki w oddzielone znakiem nowej linii, ścieżki te zostaną wybrane w oknie zatwierdzenia po wyświetleniu komunikatu błędu.

Pre-commit
: Wywoływany po kliknięciu przez użytkownika OK w oknie zatwierdzenia, a przed faktycznym początkiem zatwierdzenia. To przechwycenie dysponuje listą dokładnie wszystkich elementów, które zostaną zatwierdzone.

Post-commit
: Called after the commit finishes successfully.

Start-update
: Wywoływany przed wyświetleniem okna dialogowego uaktualnij-do-wersji.

Pre-update
: Wywoływany zanim rozpocznie się aktualizacja lub przełączenie.

Post-update
: Wywoływany po aktualizacji, przełączeniu lub pobraniu (zarówno udanym jak i nie).

Pre-connect
: Wywoływany przed próbą połączenia się z repozytorium. Wywoływany co najwyżej raz na pięć minut.

Pre-lock
: Wywoływane przed próbą zablokowania pliku.

Post-lock
: Wywoływane po zablokowaniu pliku.

Przechwycenie jest zdefiniowane dla danej ścieżki kopii roboczej. Wystarczy tylko określić ścieżkę najwyższego poziomu; jeśli wykonuje się operację na podkatalogu, TortoiseSVN będzie automatycznie wyszukiwać w górę do pasującej ścieżki.

Następnie należy określić z wiersz polecenia do wykonania, począwszy od ścieżki do skryptu przechwytującego lub pliku wykonywalnego. Może to być plik wsadowy, plik wykonywalny lub dowolny inny plik, który ma ważne skojarzenie w Windows, np. skrypt. Zauważ, że skrypt nie może być określony przy użyciu ścieżki UNC gdyż powłoka systemu Windows nie pozwala na wykonanie takich skryptów ze względu na ograniczenia bezpieczeństwa.

Wiersz polecenia zawiera kilka parametrów, które wypełnia TortoiseSVN. Przekazywane parametry zależą od wywoływanego rodzaju przechwycenia. Każde przechwycenie ma swoje własne parametry, które są przekazywane w następującym porządku:

Start-commit
: `PATHMESSAGEFILECWD`

Ręczne pre-zatwierdzenie
: `PATHMESSAGEFILECWD`

Sprawdzenie zatwierdzenia
: `PATHMESSAGEFILECWD`

Pre-commit
: `PATHDEPTHMESSAGEFILECWD`

Post-commit
: `PATHDEPTHMESSAGEFILEREVISIONERRORCWD`

Start-update
: `PATHCWD`

Pre-update
: `PATHDEPTHREVISIONCWD`

Post-update
: `PATHDEPTHREVISIONERRORCWDRESULTPATH`

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

Pre-lock
: `PATHLOCKFORCEMESSAGEFILEERRORCWD`

Post-lock
: `PATHLOCKFORCEMESSAGEFILEERRORCWD`

Znaczenie każdego z tych parametrów jest opisane tutaj:

PATH
: Ścieżka do pliku tymczasowego, który zawiera wszystkie ścieżki, dla których operacja została uruchomiona. Każda ścieżka jest w osobnej linii w pliku tymczasowym.

  Należy zauważyć, że dla operacji wykonywanych zdalnie, np. w przeglądarce repozytorium, ścieżki te nie są lokalne ale adresami url odnośnych elementów.

DEPTH
: Głębokość, z którą zatwierdzenie/aktualizacja jest wykonywana.

Dozwolonymi wartościami są:

-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
: Ścieżka do pliku zawierającego opis zatwierdzanej zmiany. Plik zawiera tekst w formacie UTF-8. Po pomyślnym wykonaniu przechwycenia start-commit, opis zmiany jest ponownie odczytywany, dając przechwyceniu szansę, aby go zmodyfikować.

REVISION
: Wersja repozytorium, do której powinna doprowadzić aktualizacja lub zatwierdzenie.

LOCK
: Wartość `true` podczas blokowania, albo `false` przy odblokowywaniu.

FORCE
: Wartości `true` albo `false`, zależnie czy operacja była wymuszona, czy nie.

ERROR
: Ścieżka do pliku zawierającego komunikat o błędzie. Jeśli nie było błędu, plik będzie pusty.

CWD
: Bieżący katalog roboczy, w którym skrypt jest uruchamiany. Jest ustawiony na wspólny katalog główny wszystkich dotkniętych ścieżek.

RESULTPATH
: Ścieżka do pliku tymczasowego zawierającego wszystkie ścieżki, których w jakikolwiek sposób dotyczy operacja. Każda ścieżka jest w osobnej linii w pliku tymczasowym.

Zauważcie, że chociaż nadaliśmy te nazwy parametrów dla wygody, nie musicie odwoływać się do tych nazw w ustawieniach przechwycenia. Wszystkie parametry podane dla danego haka są zawsze przekazywane, czy chcesz ich, czy nie ;-)

Jeśli chcecie by operacja Subversion została wstrzymana się do zakończenia wykonania przechwycenia, zaznaczcie Czekaj na zakończenie skryptu.

Zwykle chcecie ukryć brzydkie okienka DOS podczas działania skryptu, więc Ukryj skrypt kiedy jest wykonywany jest domyślnie zaznaczone.

Przykładowe skrypty przechwytujące klienta można znaleźć w folderze `contrib` w *repozytorium TortoiseSVN* [https://svn.code.sf.net/p/tortoisesvn/code/trunk/contrib/hook-scripts]. (Sekcja 3, „Licencja” wyjaśnia, jak uzyskać dostęp do repozytorium.)

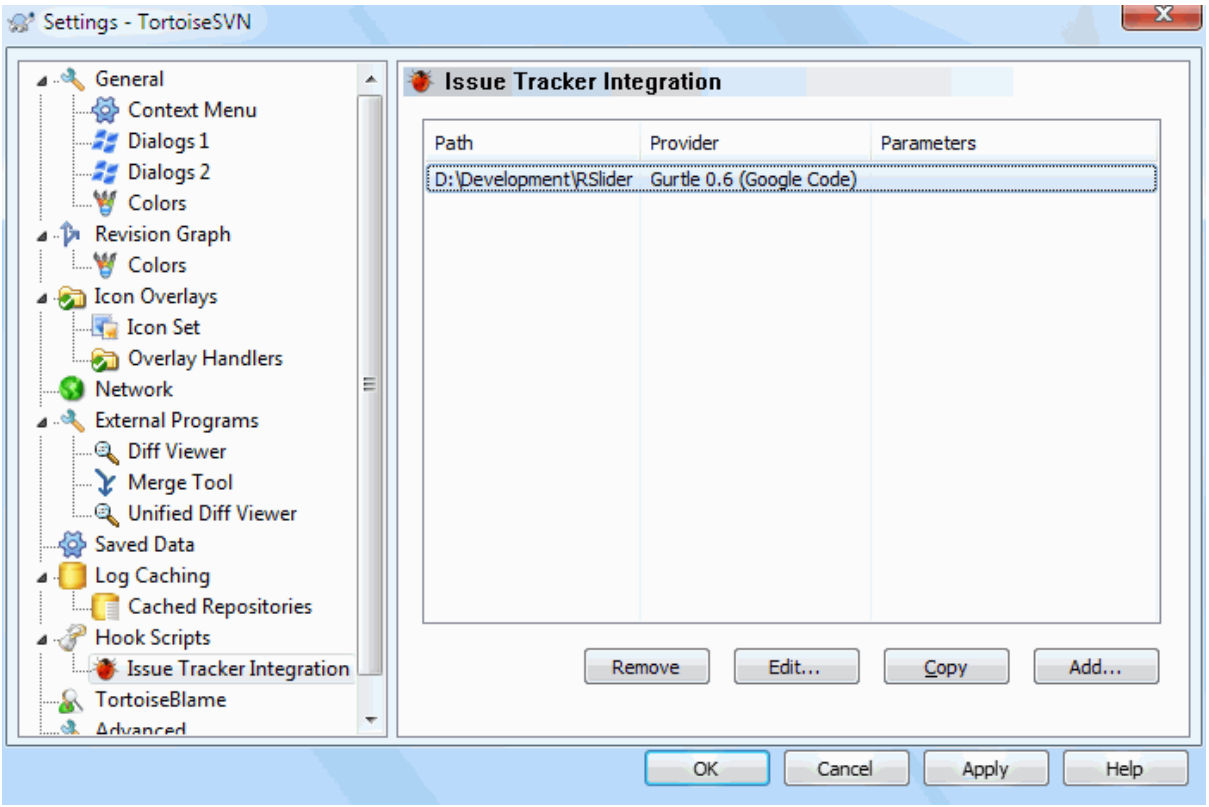
Podczas debugujących skryptów przechwytujących w których chcecie wypisać linie postępu w konsoli DOS, lub wstawić wstrzymanie by wstrzymać zamknięcie okna konsoli gdy skrypt się zakończy. Ponieważ WE/WY jest przekierowane nie będzie to sdziałać poprawnie. Można jednakt przekierować wejście i wyjście na CON by to naprawić, np.

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

W folderze instalacji TortoiseSVN znajduje się małe narzędzie o nazwie `ConnectVPN.exe`. Możecie użyć tego narzędzia skonfigurowanego jako przechwycenie pre-connect do automatycznego połączenia z siecią VPN zanim TortoiseSVN spróbuje połączyć się z repozytorium. Wystarczy podać nazwę połączenia VPN jako pierwszy parametr narzędzia.

4.31.8.1. Integracja ze śledzeniem problemów

TortoiseSVN może użyć wtyczki COM by wykonywać zapytania do trackerów problemów, podczas wejścia okno zatwierdzenia. Korzystanie z takich wtyczek jest opisane w Sekcja 4.29.2, „Pobieranie informacji z trackera problemów”. Jeśli administrator systemu przekazał Wam wtyczki i są już zainstalowane i zarejestrowane, jest to miejsce, aby określić w jaki sposób integrują się one z kopią roboczą.



Rysunek 4.91. Okno dialogowe ustawień, strona Integracja z systemami śledzenia błędów

Kliknijcie na Dodaj... by użyć wtyczki z określonej kopii roboczej. Tutaj możecie określić ścieżkę kopii roboczej, wybrać wykorzystaną wtyczkę z rozwijanej listy wszystkich zarejestrowanych wtyczek trackera problemów, oraz wszelkie parametry do przekazania. Parametry będą specyficzne dla wtyczki, ale powinien zawierać nazwę użytkownika trackera, aby plugin mógł odnaleźć problemy, które są przypisane do Ciebie.

Jeśli chcecie by wszyscy użytkownicy używali tej samej wtyczki COM dla projektu, można określić wtyczkę także poprzez atrybuty `bugtraq:provideruuid`, `bugtraq:provideruuid64` i `bugtraq:providerparams`.

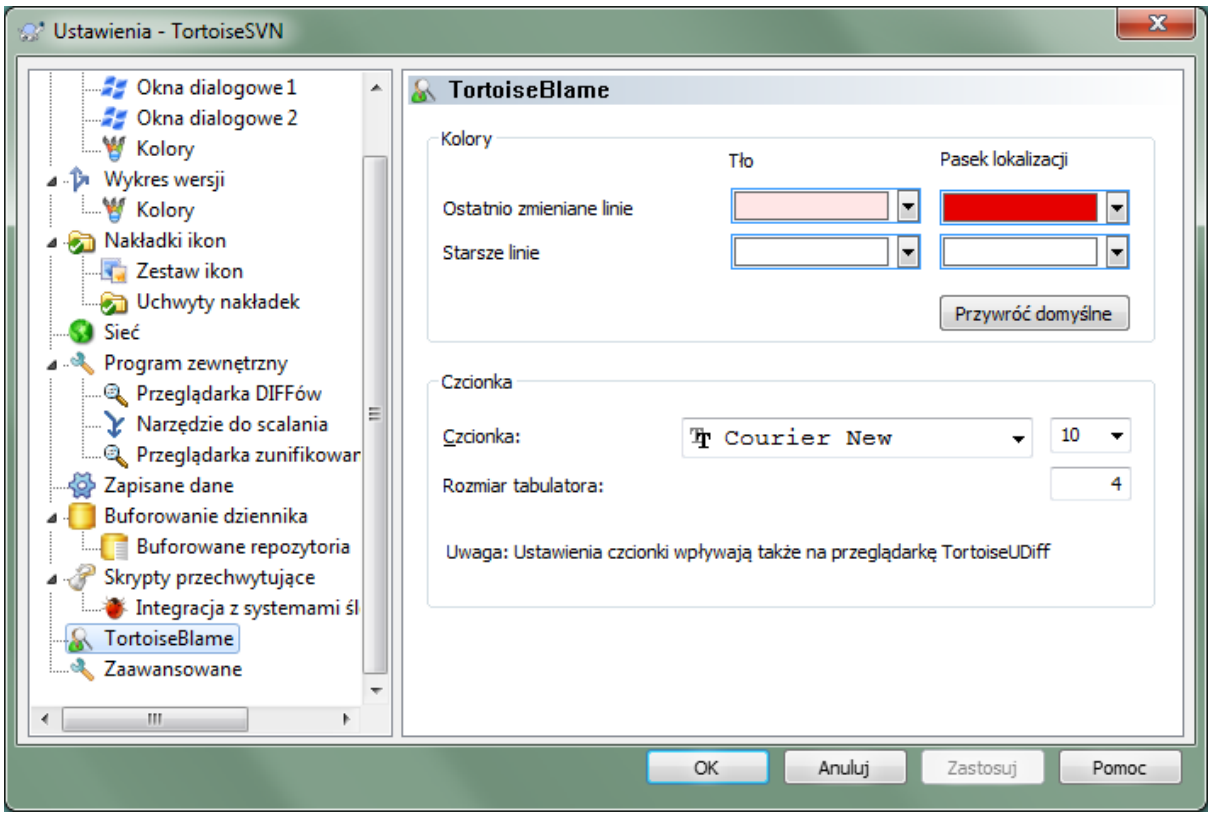
bugtraq:provideruuid
: Ta właściwość określa COM UUID dla IBugtraqProvider, na przykład `{91974081-2DC7-4FB1-B3BE-0DE1C8D6CE4E}`. (Ten przykład to UUID *dostawcy Gurtle bugtraq* [http://code.google.com/p/gurtle/], który jest dostawcą dla systemu śledzenia błędów *Google Code* [http://code.google.com/hosting/].)

bugtraq:provideruuid64
: To jest to samo co `bugtraq:provideruuid`, ale dla 64-bitowej wersji IBugtraqProvider.

bugtraq:providerparams
: Ta właściwość określa parametry przekazane do IBugtraqProvider.

Sprawdźcie w dokumentacji swojej wtyczki IBugtraqProvider aby dowiedzieć się, jak wypełnić te dwa atrybuty.

4.31.9. Ustawienia TortoiseBlame



Rysunek 4.92. Okno dialogowe ustawień, strona TortoiseBlame

Ustawienia używane przez TortoiseBlame są sterowane z głównego menu kontekstowego, a nie bezpośrednio z samego TortoiseBlame.

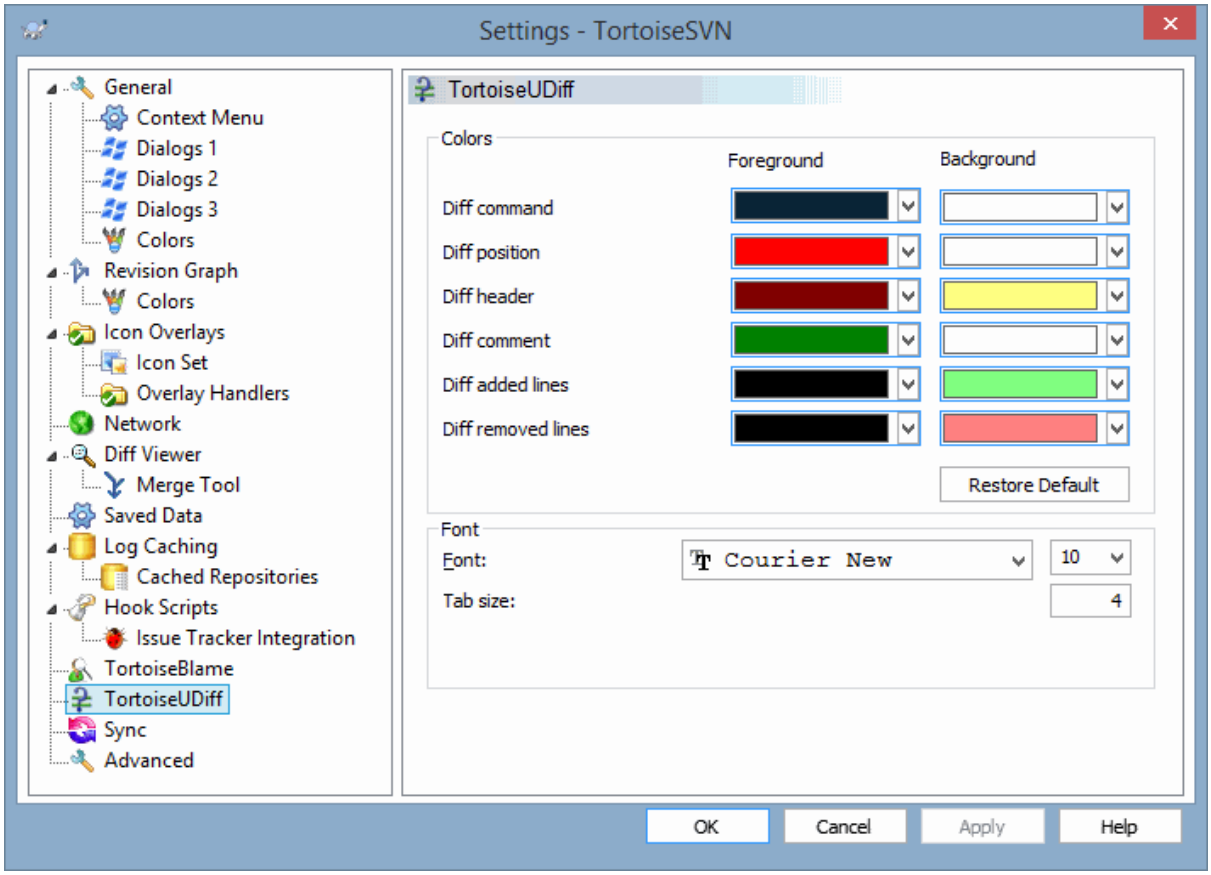
Kolory
: TortoiseBlame może używać koloru tła by wskazać wiek linii w pliku. Ustawia się punkty końcowe poprzez określenie kolorów dla najnowszych i najstarszych wersji, a TortoiseBlame użyje interpolacji liniowej pomiędzy tymi kolorami w zależności od wersji repozytorium wskazanej dla każdej linii.

  Można określić różne kolory do wykorzystania na pasku lokalizacji. Domyślnie używany jest silny kontrast na pasku lokalizacji przy zachowaniu jasnego tła okna głównego, dzięki czemu można wciąż przeczytać tekst.

Czcionka
: Możecie wybrać czcionkę używaną do wyświetlania tekstu i używaną wielkość czcionki. Dotyczy to zarówno zawartości pliku, jak i informacji o autorze i wersji pokazanych w lewym panelu.

Tabulatory
: Określa ile spacji użyć do ekspansji, gdy znak tabulatora znajduje się w pliku.

4.31.10. Ustawienia TortoiseUDiff



Rysunek 4.93. Okno Ustawień, Strona TortoiseUDiff

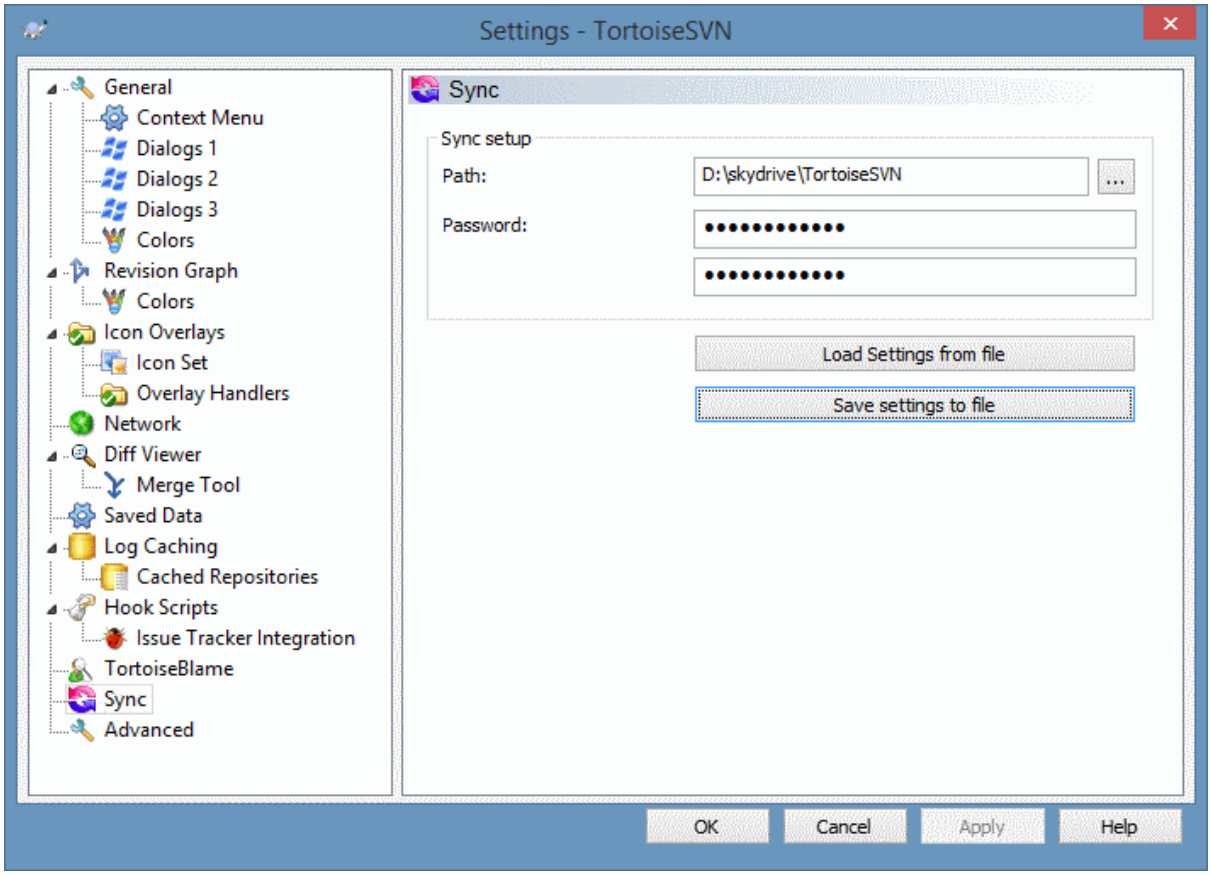
Ustawienia używane przez TortoiseUDiff są ustawiane w podstawowym menu kontekstowym, nie zaś w samum TortoiseUDiff.

Kolory
: Domyślne kolory używane przez TortoiseUDiff są z reguły w porządku, ale tutaj można je poprawić.

Czcionka
: Możecie wskazać czcionkę używaną do wyświetlania treści oraz wielkość liter.

Tabulatory
: Definiuje ile spacji należy użyć do rozsunięcia tekstu, gdy napotkany zostanie znak tabulacji w pliku różnic.

4.31.11. Eksportowanie ustawień TSVN



Rysunek 4.94. Okno Ustawień, Strona Synchro

Możecie synchronizować wszystkie ustawienia TortoiseSVN do i z szyfrowanego pliku. Plik jest szyfrowany wpisanym przez was hasłem zatem nie musicie się niczego obawiać umieszczając go w folderze w chmurze jak OneDrive, GDrive, DropBox, ...

Gdy wskazano już ścieżkę i hasło, TortoiseSVN będzie automatycznie synchronizował wszystkie ustawienia i zachowa je zsynchronizowane.

Możecie importować/eksportować zaszyfrowane pliki ze wszystkimi ustawieniami ręcznie. Robiąc to będziecie proszeni o ścieżkę do pliku i hasło by zaszyfrować/odszyfrować plik.

Podczas ręcznego eksportu ustawień możecie również opcjonalnie dodać wszystkie lokalne ustawienia nie ujęte w zwykłym eksporcie oraz synchronizacji. Ustawienia lokalne to te zawierające lokalne ścieżki, które często różnią się na poszczególnych komputerach. Te lokalne ustawienia dotyczą narzędzi porównywania i scalania oraz skryptów przechwytujących.

4.31.12. Ustawienia zaawansowane

Kilka rzadko używanych ustawień jest dostępnych tylko na stronie zaawansowanych ustawień. Ustawienia te modyfikują bezpośrednio rejestr zatem trzeba wiedzieć, do czego każde z tych ustawień jest używane i jak działa. Nie należy modyfikować tych ustawień, chyba że jesteście pewni, że ich zmiana jest konieczna.

AllowAuthSave
: Czasami wielu użytkowników korzysta z tego samego konta na tym samym komputerze. W takich sytuacjach naprawdę niepożądane jest zapisywanie danych uwierzytelniania. Ustawienie tej wartości na `false` wyłącza `Zapisz parametry uwierzytelnienia` w oknie dialogowym uwierzytelniania.

AllowUnversionedObstruction
: Jeśli aktualizacja dodaje nowy plik z repozytorium, który już istnieje w lokalnej kopii roboczej jako niewersjonowany, domyślną akcją jest utrzymanie lokalnego pliku, pokazując go jako (prawdopodobnie) zmodyfikowaną wersję nowego pliku z repozytorium. Jeśli wolicie by TortoiseSVN tworzył konflikt w takich sytuacjach, ustawcie tę wartość na `false`.

AlwaysExtendedMenu
: Tak jak w Eksploratorze, TortoiseSVN pokazuje dodatkowe polecenia, jeśli klawisz **Shift** był naciśnięty podczas otwierania menu kontekstowego. Aby zmusić TortoiseSVN by zawsze pokazał rozszerzone polecenia, ustawcie tę wartość na `true`.

AutoCompleteMinChars
: Minimalna liczba znaków, od której edytor pokazuje okienko autouzupełniania. Domyślna wartość to `3`.

AutocompleteRemovesExtensions
: Lista automatycznego uzupełniania wyświetlana w edytorze opisu zmiany zatwierdzenia wyświetla nazwy plików wymienionych w zatwierdzeniu. Aby pokazać te nazwy również z usuniętym rozszerzeniem, ustawcie tę wartość na `true`.

BlockPeggedExternals
: Zewnętrzne pliki, które są sztywno związane z konkretną rewizją są domyślnie blokowane przed wybraniem do przesłania do repozytorium. Jest tak dlatego, że kolejna aktualizacja może odwrócić te zmiany chyba, że powiązana rewizja jest dopasowana

  Ustaw tę wartość na `fałsz` w przypadku gdy wciąż chcesz wysłać do repozytorium zmiany w zewnętrznych plikach.

BlockStatus
: Jeśli nie chcecie by eksplorator aktualizował stan nakładek podczas działania innego polecenia TortoiseSVN (np. Uaktualnienie, Zatwierdzenie, ...), należy ustawić tę wartość na `true`.

CacheTrayIcon
: Aby dodać ikonę bufora w zasobniku dla programu TSVNCache ustawcie tę wartość na `true`. Jest to naprawdę przydatne tylko dla programistów, ponieważ pozwala na eleganckie zakończenie programu.

ColumnsEveryWhere
: Dodatkowe kolumny dodawane przez TortoiseSVN do widoku Szczegóły w eksploratorze Windows są zazwyczaj aktywne tylko w kopii roboczej. Jeśli chcecie, by były dostępne wszędzie, nie tylko w kopii roboczej, ustawcie tę wartość na `true`. Należy pamiętać, że dodatkowe kolumny są dostępne tylko w XP. Vista i nowsze nie obsługują już tej funkcji. Jednak niektóre programy firm trzecich zastępujące eksplorator nie obsługują tego nawet w wersjach Windows późniejszych niż XP.

ConfigDir
: Tutaj można określić inną lokalizację dla pliku konfiguracyjnego Subversion. Wpłynie to na wszystkie operacje TortoiseSVN.

CtrlEnter
: W większości okien dialogowych w TortoiseSVN można użyć **Ctrl+Enter**, aby zamknąć okno dialogowe jak po kliknięciu na przycisk OK. Jeśli nie chcecie takiego działania, ustawcie tę wartość na `false`.

Debug
: Ustawcie właściwość na `true`, jeśli chcecie dla każdego polecenia wyświetlić okno pokazujące wiersz polecenia używanego do uruchomienia TortoiseProc.exe.

DebugOutputString
: Ustawcie to na `true`, jeśli chceszcie by TortoiseSVN wydrukował komunikaty debugowania w trakcie wykonania. Wiadomości mogą być przechwytywane tylko przy użyciu specjalistycznych narzędzi debugowania.

DialogTitles
: Domyślny format (wartość 0) tytułów okna dialogowego jest `url/ścieżki - nazwa dialogu - TortoiseSVN`. Jeśli ustawicie tę wartość na 1, format zmieni się na `nazwa dialogu - url/ścieżka - TortoiseSVN`.

DiffBlamesWithTortoiseMerge
: TortoiseSVN pozwala na przypisanie zewnętrznej porównywarki. Większość takich przeglądarek jednak nie nadaje się do adnotowania zmian (Sekcja 4.24.2, „Różnice adnotacji”), więc może chcecie wrócić w tym przypadku do TortoiseMerge. Aby to zrobić, ustawcie tę wartość na `true`.

DlgStickySize
: Ta wartość określa liczbę pikseli poniżej której okno dialogowe przykleja się do krawędzi. Domyślna wartość to 3. By wyłączyć takie zachowanie należy ustawić wartość na zero.

FixCaseRenames
: Niektóre aplikacje zmieniają wielkość liter w nazwach plików bez ostrzeżenia, jednak te zmiany nie są tak naprawdę potrzebne ani pożądane. Na przykład zmiana z `file.txt` na `FILE.TXT` nie przeszkadza normalnym aplikacjom Windows, ale Subversion jest wrażliwe nawielkość liter w takich sytuacjach. Dlatego TortoiseSVN naprawia takie zmiany wielkości liter.

  Jeśli nie chcecie, by TortoiseSVN automatycznie naprawiało takie zmiany wielkości liter, możecie ustalić tę wartość na `false`.

FullRowSelect
: Kontrolka listy statusów, która jest stosowana w różnych oknach dialogowych (np. zatwierdzania, sprawdzenia zmian, dodawania, przywrócenia, ...) wykorzystuje pełen wybór wierszy (np. jeśli wybierzecie pozycję, cały wiersz jest zaznaczony, nie tylko pierwsza kolumna). To działa, ale w wybranym wierszu zaznaczenie obejmuje także obraz tła w dolnym prawym rogu, co może wyglądać brzydko. Aby wyłączyć pełny wybór wiersza, ustawcie tę wartość na `false`.

GroupTaskbarIconsPerRepo
: Opcja ta określa, jak mają być pogrupowane ikony paska zadań Win7 z różnych dialogów i okien TortoiseSVN. Opcja ta nie działa w Windows XP ani Vista!

1. Wartością domyślną jest 0. Z tym ustawieniem ikony grupowane są razem względem typu aplikacji. Wszystkie okna TortoiseSVN są w jednej grupie, a wszystkie okna z TortoiseMerge są w drugiej grupie, ...

Rysunek 4.95. Pasek zadań z grupowaniem domyślnym

2. Jeśli jest ustawiona na 1, to zamiast wszystkich okien w jednej grupie według aplikacji, są one grupowane po repozytoriach. Na przykład, są otwarte okna dziennika i zatwierdzenia dla repozytorium `A` oraz okna sprawdź-modyfikacje i dziennika dla repozytorium `B`, to będą wyświetlane dwie grupy ikon aplikacji na pasku zadań Windows 7, po jednej grupie dla każdego repozytorium. Ale okna TortoiseMerge nie będą zgrupowane razem z dialogami TortoiseSVN.

Rysunek 4.96. Pasek zadań z grupowaniem według repozytoriów

3. Jweśli ustawiona na 2, grupowanie działa jak dla wartości 1, z tym że okna TortoiseSVN, TortoiseMerge, TortoiseBlame, TortoiseIDiff i TortoiseUDiff są grupowane razem. Na przykład mając otwarty dialog zatwierdzenia, jeśli dwa razy klikniecie na zmienionym pliku, otwarte okno porównania TortoiseMerge zostanie wstawione do tej samej grupy na pasku zadań, co okno zatwierdzenia.

Rysunek 4.97. Pasek zadań z grupowaniem według repozytoriów

4. Jeśli jest ustawiona na 3,grupowanie działa jak dla ustawienia opcji na 1, ale grupy nie są tworzone według repozytorium, ale kopii roboczej. Jest to przydatne jeśli trzymacie wszystkie projekty w tym samym repozytorium, zaś różne kopie robocze przeznaczone są dla poszczególnych projektów.

5. Jeśli jest ustawiona na 4, grupowanie działa jak przy ustawieniu opcji na 2, ale grupy nie są tworzone według repozytorium, ale względem kopii roboczej.

HideExternalInfo
: Jeśli ustawiono na `false`, wtedy każda z `svn:externals` jest pokazywana sosbno podczas aktualizacji.

  Jeśli ustawiono na `true` (domyślnie), infirmacja aktualizacji dla zewnętrznych jest pokazywana tylko jeśli zewnętrzne podlegają aktualizacji,tj zostały w jwkiś sposób zmienione. W innymprzypadku nie pokazuje się nic jak dla zwykłych plików i folderów.

GroupTaskbarIconsPerRepoOverlay
: Nie ma to zastosowania, jeśli opcja `GroupTaskbarIconsPerRepo` jest ustawiona na 0 (patrz wyżej).

  Jeśli ta opcja jest ustawiona na`true`, każda ikona na pasku zadań Win7 pokazuje się z małą kolorową prostokątną nakładką, wskazując repozytorium, do którego odnoszą się dialogi lub okna.

Rysunek 4.98. Grupowanie na pasku zadań z nakładkami koloru repozytorium

IncludeExternals
: Domyślnie TortoiseSVN zawsze wykonuje aktualizację wraz z zewnętrznymi. Pozwala to uniknąć problemów z niespójnością kopii roboczej. Jeśli jednak wskazano dużo zewnętrznych, aktualizacja może zająć trochę czasu. Ustawcie tę wartość na `false`, aby uruchamiać aktualizację domyślnie bez zewnętrznych. Aby zaktualizować z włączonymi zewnętrznymi, uruchomcie okno `Uaktualnij do wersji...` lub ustawcie wartość `true` ponownie.

LogFindCopyFrom
: Kiedy okno dziennika jest uruchamiane z kreatora scalenia, już scalone wersje są wyświetlane w kolorze szarym, jednocześnie pokazane są też wersje sprzed punktu utworzenia gałęzi. Te wersje są wyświetlane w kolorze czarnym, ponieważ te nie mogą zostać scalone.

  Jeśli ta opcja jest ustawiona na `true` to TortoiseSVN próbuje znaleźć wersję gdy gałąź została utworzona i ukryć wszystkie wersje, które są przed tą wersją. Ponieważ może to zająć trochę czasu, opcja jest domyślnie wyłączona. Ponadto opcja ta nie działa z niektórymi serwerami SVN (np. Google Code Hosting, zobacz *zgłoszenie #5471* [http://code.google.com/p/support/issues/detail?id=5471]).

LogMultiRevFormat
: Ciąg znaków formatujący kolunikaty dziennika gdy wiele wersji zostało wybranych w oknie dziennika.

Można użyć następujących symboli zastępczych w formatującym ciągu znaków:

%1!ld!
zostanie zastąpiony zawartością numeru wersji

%2!s!
zostanie zastąpiony przez krótki komunikat dziennika dla wersji

LogStatusCheck
Okno dialogowe dziennika pokazuje aktualną wersję ścieżki kopii roboczej pogrubioną czcionką. Wymaga to jednak, by okno dziennika wczytało stan tej ścieżki. Ponieważ dla bardzo dużych kopii roboczych może to trochę potrwać, możecie ustawić jej wartość na `false` by wyłączyć tę funkcję.

MergeLogSeparator
Podczas scalania zmian z innej gałęzi oraz scalania informacji śledzenia istnieje możliwość, by opisy zmian ze scalanych wersji były gromadzone w celu uzupełnienia opisu zmiany zatwierdzenia. Predefiniowany ciąg jest używany do oddzielania poszczególnych opisów scalonych zmian. Możecie ustawić opcję na wartość zawierającą pasujący ciąg znaków rozdzielających.

NumDiffWarning
Jeśli chcecie pokazać różnice jednocześnie dla większej liczby elementów niż określono w tym ustawieniu, zostanie najpierw wyświetlone okno dialogowe z ostrzeżeniem. Wartością domyślną jest 10.

OldVersionCheck
TortoiseSVN sprawdza raz w tygodniu, czy została już wydana nowa wersja. Jeśli zaktualizowana wersja zostanie znaleziona, okno dialogowe zatwierdzenia wyświetla kontrolkę z linkiem oraz tą informacją. Jeśli wolicie wrócić do poprzedniego zachowania, w którym pojawia się okno dialogowe z powiadomieniem o aktualizacji, ustawcie tę wartość na `true`.

RepoBrowserTrySVNParentPath
Przeglądarka repozytorium próbuje pobrać stronę wygenerowaną przez serwer SVN skonfigurowaną w dyrektywie SVNParentPath pobierającej listę wszystkich repozytoriów. By zablokować to działanie, należy ustawić tą wartość na `false`.

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

ScintillaDirect2D
Ta opcja włącza użycie przyspieszenia rysowania Direct2D w kontrolkach Scintilla, używanych jako obszary edycji w np. oknie zatwierdzenia a także w przeglądarce plików różnicowych. Z niektórymi kartami graficznymi nie działa to jednak poprawnie przez co kursor przy wprowadzaniu tekstu nie zawsze jest widoczny. Jeśli tak się zdarza, można wyłączyć funkcję ustawiając tą wartość na `false`.

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

0
The user is asked whether to update the working copy or not, and the commit dialog is not reopened after the update.

1
This is the default. The user is asked whether to update the working copy or not, and the commit dialog is reopened after the update so the user can proceed with the commit right away.

2
Similar to `1`, but instead of updating only the paths selected for a commit, the update is done on the working copy root. This helps to avoid inconsistent working copies.

3
: The user is not asked to update the working copy. The commit simply fails with the out-of-date error message.

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

ShellMenuAccelerators
: TortoiseSVN wykorzystuje skróty klawiaturowe dla swoich wpisów w menu kontekstowym eksploratora. Ponieważ może to prowadzić do podwójnych skrótów (np. `SVN Uaktualnij` ma skrót **Alt-U**, ale ma go też wpis `Utwórz skrót` eksploratora). Jeśli nie chcecie ani nie potrzebujecie akceleratorów dla wpisów TortoiseSVN , ustawcie tę wartość na `false`.

ShowContextMenuIcons
: Może to być przydatne, jeśli używacie czegoś innego niż eksplorator Windows lub macie problemy prawidłowym wyświetlaniem z menu kontekstowego. Ustawcie tę wartość na `false`, jeśli nie chcecie by TortoiseSVN wyświetlał ikony dla elementów powłoki menu kontekstowego. Ustawcie tę wartość na `true`, aby pokazać ikony ponownie.

ShowAppContextMenuIcons
: Jeśli nie chcecie by TortoiseSVN pokazywał ikony menu kontekstowego w swoich oknach, ustawcie tę wartość na `false`.

StyleCommitMessages
: Okna zatwierdzenia i dziennika używają stylów (np. pogrubienie, kursywa) w opisach zmian (patrzcie Sekcja 4.4.5, „Wiadomości dziennika zatwierdzeń” dla szczegółów). Jeśli nie chcecie, aby to zrobić, należy ustawić wartość `false`.

UpdateCheckURL
: Wartość ta zawiera adres URL, z którego TortoiseSVN próbuje pobrać plik tekstowy, aby dowiedzieć się czy są dostępne aktualizacje programu. Może to być przydatne dla administratorów firmy, którzy nie chcą by ich użytkownicy aktualizowali TortoiseSVN, dopóki ci tego nie zatwierdzą.

VersionCheck
: TortoiseSVN sprawdza, czy jest dostępna nowa wersja raz w tygodniu. Jeśli nie chcecie by TortoiseSVN wykonywał takie sprawdzenie, ustawcie tę wartość na `false`.

4.32. Ostatni krok

Obdarujcie nas!

Chociaż TortoiseSVN i TortoiseMerge są bezpłatne, można wspierać deweloperów wysyłając poprawki i odgrywając aktywną rolę w ich rozwoju. Możecie również pomóc dopingując nam podczas niekończących godzin, które spędzamy przed komputerami.

Podczas pracy nad TortoiseSVN lubimy słuchać muzyki. A ponieważ spędzajmy wiele godzin na projektem, potrzebujemy *wiele* muzyki. Dlatego stworzyliśmy kilka list życzeń z naszymi ulubionymi płytami CD i DVD: https://tortoisesvn.net/donate.html Prosimy również spojrzeć na listę osób, które przyczyniły się do projektu, wysyłając poprawki lub tłumaczenia.

Rozdział 5. Monitor Projektu

Monitor projektu jest pomocnym narzędziem sprawdzającym regularnie repozytoria i informującym że wystąpiły tam nowe zatwierdzenia.

Projekty mogą być monitorowane przez ścieżkę kopii roboczej lub bezpośrednio przez adresy URL ich repozytoriów.

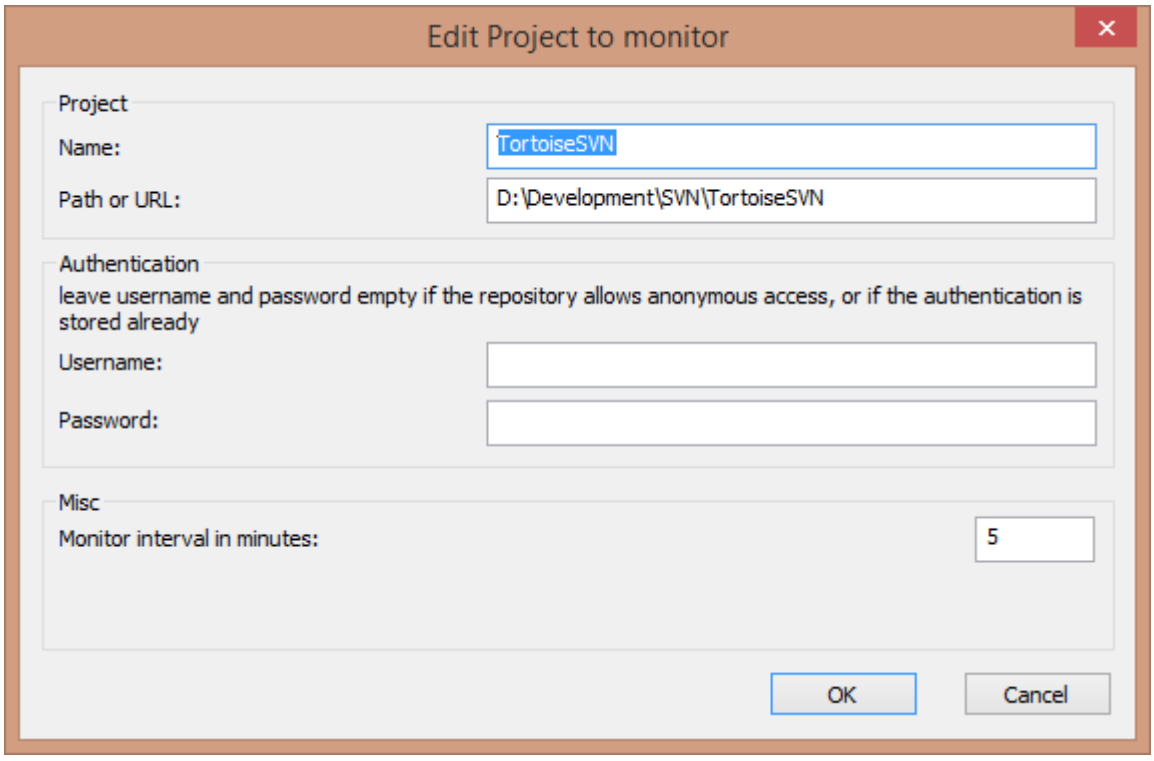
Monitor projektu skanuje każdy projekt co konfigurowanly okres czasu i za każdym razem, gdy zostaną znalezione nowe zatwierdzenia, pokazywane jest wyskakujące okno powiadomienia. Również ikona, która dodawana jest w zasobniku systemowym, zmienia się by wskazać, że są nowe zatwierdzenia.

Snarl

Gdy zainstalowano i aktywowano *Snarl* [https://snarl.fullphat.net/], monitor projektu automatycznie wykorzystuje Snarl do wyświetlania powiadomień o świeżo wyszukanych zatwierdzeniach.

5.1. Dodawanie projektów do monitora

Gdy pierwszy raz uruchamiacie monitor projektu, widok drzewa po lewej stronie jest pusty. By dodać projekt, kliknijcie przycisk Dodaj Projekt u góry okienka.



Rysunek 5.1. Okno edycji projektu monitora projektu

By dodać projekty do śledzenia, wypełnijcie wymagane informacje. Nazwa projektu nie jest opcjonalna i musi być wpisane, wszystkie inne informacje są opcjonalne.

Jeśli pole `Ścieżka lub URL` pozostawiono puste, zostanie dodany folder. Jest to użyteczne do grupowania śledzonych projektów.

Pola `Nazwa użytkownika` i `Hasło` powinny być odfiltrowane tylko jeśli repozytorium nie przewiduje anonimowego dostępu do odczytu , i tylko jeśli dane autoryzacji nie są przechowywane w samym Subversion.

Jeśli macie dostęp do monitorowanego repozytorium przy użyciu TortoiseSVN lub innych klientów svn i już zapisaliście dane autoryzacji, powinniście zostawić je puste: you nie będzie potrzeby ręcznej edycji tych projektów gdy hasło się zmieni.

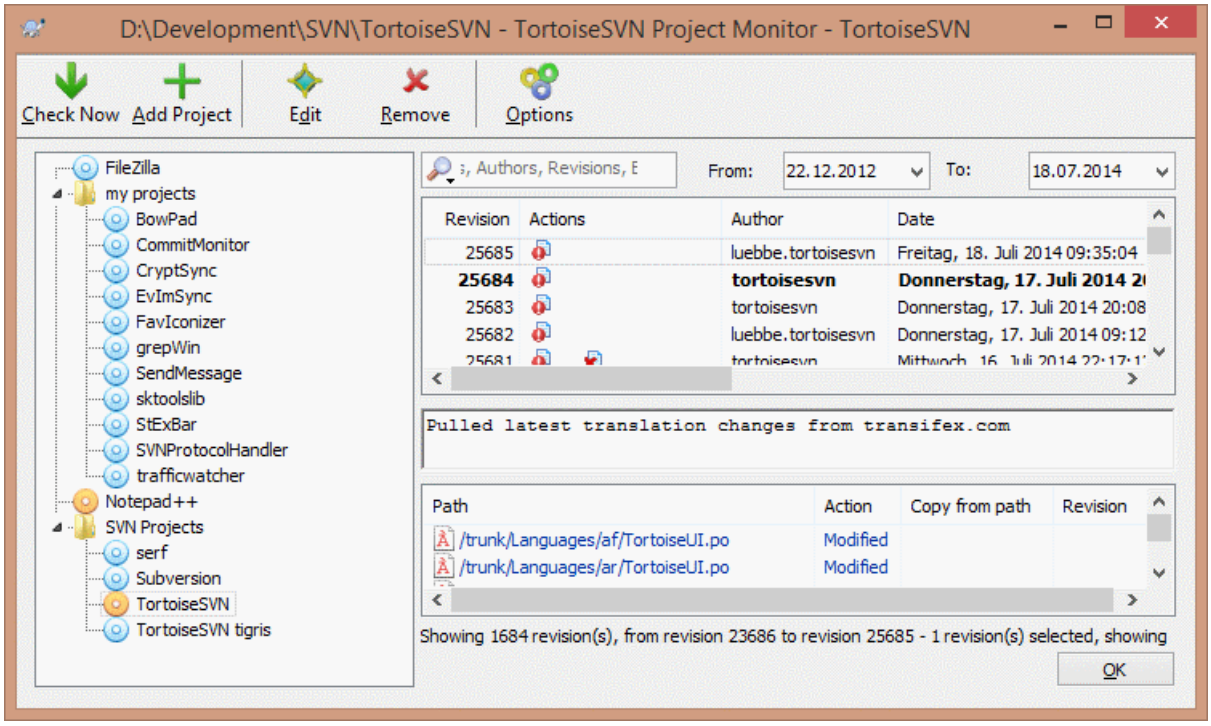
`Interwał monitorowania w minutach` określa liczbę minut oczekiwania na kolejne sprawdzenie. Najmniejszy przedział czasowy to jedna minuta.

interwał sprawdzenia

IJesli wiele osób monitoruje to samo repozytorium a przepustowość procesora jest ograniczona, administrator repozytorium może ustawić minimalny interwał sprawdzenia używając pliku `svnrobots.txt`. Szczegółowe wyjaśnienie jak to działa można znaleźć na stronie monitora projektu:

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

5.2. Okno monitora



Rysunek 5.2. Okno podstawowe monitora projektu

Monitor projektu wyświetla wszystkie monitorowane projekty z lewej strony w widoku drzewa. Projekty mogą być przenoszone względem siebie, na przykład jeden projekt może zostać przeniesiony poniżej innego projektu, stając się jego potomkiem/podprojektem.

Kliknięcie na projekt pokazuje po prawej stronie komunikaty dziennika.

Projekty mające aktualizacje wyświetlane są pogrubioną czcionką, z liczbą zatwierdzeń w nawiasach z prawej. kliknięcie na projekcie oznacza go automatycznie jako przeczytany.

5.2.1. Operacje podstawowe

Oasek zadań u góry okna pozwala na skonfigurowanie i obsługę monitora projektu.

Sprawdź Teraz
: Podczas każdy monitorowany projekt jest sprawdzany zgodnie z ustalonym interwałem, kliknięcie w ten przycisk wymusi natychmiastowe sprawdzenie wszystkich projektów. Zauważcie, że jeśli znajdują się tam aktualizacje, powiadomienia nie ukażą się dopóki wszystkie projekty nie zostaną sprawdzone.

Dodaj Projekt
: Otwiera nowe okno by skonfigurować monitorowanie nowego projektu.

Edytuj
: Otwiera okno konfiguracji wybranego projektu.

Usuń
: Osowa wybrany projekt po wyświetleniu dialogu z potwierdzeniem decyzji.

Zaznacz wszystko jako przeczytane
: Zaznacza wszystkie wersje we wszystkich projektach jako przeczytane. Zauważcie, że jeśli wybierze się projekt z nieprzeczytanymi wersjami, wersje te zostaną automatycznie zaznaczone jako przeczytane przy wybieraniu nowego projektu.

  Jeśli przytrzyma się klawisz **Shift** klikając przycisk, wszystkie stany błędów zostaną również wyczyszczone, o ile są.

Aktualizuj wszystko
: Uruchamia polecenie **Aktualizuj** na wszystkich monitorowanych kopiach roboczych. Nie są aktualizowane projekty monitorowane przez adres url , a tylko odnotowane przez ścieżkę do kopii roboczej.

Opcje
: Pokazuje okno konfiguracji zachowania monitora projektów.

Rozdział 6. Program SubWCRev

SubWCRev to program konsoli Windows, który może być wykorzystany do odczytu stanu kopii roboczej Subversion i ewentualnie podstawiania słów kluczowych w pliku szablonu. Jest to często wykorzystywane jako część procesu budowania jako środek wstrzyknięcia informacji o kopii roboczej do kompilowanego obiektu. Zazwyczaj można to wykorzystać do włączenia numeru wersji w polu „O programie”.

6.1. Linia poleceń SubWCRev

SubWCRev odczytuje status Subversion wszystkich plików w kopii roboczej, z domyślnym wyłączeniem zewnętrznych. Zapisuje najwyższy numer zatwierdzonej wersji i czas zatwierdzenia tej wersji, zapamiętuje również, czy są lokalne modyfikacje w kopii roboczej lub mieszane wersje aktualizacji. Numer wersji, zakres wersji aktualizacji i status modyfikacji są wysyłane na wyjście stdout.

SubWCRev.exe jest wywoływany z wiersza poleceń lub skryptu i jest sterowany za pomocą parametrów wiersza poleceń.

```
SubWCRev SciezkaKopiiRoboczej [PlikWerZrodl PlikWerDocel] [-nmdfe]
```

`WorkingCopyPath` jest ścieżką do sprawdzanej kopii roboczej. Można używać SubWCRev tylko na kopii roboczej, a nie bezpośrednio na repozytorium. Ścieżka może być bezwzględna lub względna do bieżącego katalogu roboczego.

Jeśli chcecie by SubWCRev podstawiał słowa kluczowe, aby takie pola jak wersje repozytorium i URL są zapisywane w pliku tekstowym, należy dostarczyć pliku szablonu `SrcVersionFile` i plik wyjściowy `DstVersionFile`, który zawiera wersja szablonu po podstawieniach.

Możecie wskazać wzorce ignorowania dla SubWCRev by zapobiec braniu pod uwagę określonych plików i ścieżek. Wzorce te są odczytywane z pliku o nazwie `.subwcrevignore`. Plik jest odczytywany ze wskazanego folderu a także z folderu głównego kopii roboczej. Jeżeli plik nie istnieje, żaden plik nie jest ignorowany. Plik `.subwcrevignore` może zawierać wiele wzorców oddzielonych znakami nowej linii. wzorce są dopasowywane do ścieżek względem folderu nadrzędnego repozytorium i sieżek względnych do ścieżki pliku `.subwcrevignore`. Na przykład, by zignorować folder `doc` w kopii roboczej TortoiseSVN, `.subwcrevignore` powinien zawierać następujące linie:

```
/trunk/doc
/trunk/doc/*
```

Albo, brzyjmując, że plik `.subwcrevignore` znajduje się w folderze głównym pobranym z linii głównej, użycie wzorców

```
doc
doc/*
```

działa tak samo jak przykład powyżej.

by ignorować wszystkie obrazki, wzorce ignorowania muszą zostać ustawione tak:

```
*.png
*.jpg
*.ico
*.bmp
```

Ważne

Wzorce ignorowania są wrażliwe na wielkość liter, tak jak sam Subversion.

Podpowiedź

Aby utworzyć plik z kropką na początku w eksploratorze Windows, należy wpisać `.subwcrevignore.`. Odnotujcie kropkę wiodącą.

Istnieje kilka opcjonalnych przełączników, które wpływają na sposób działania SubWCRev. Jeśli używasz więcej niż jednego, muszą być określone jako jedna grupa, np. `-nm`, a nie `-n -m`.

Przełącznik	Opis
-n	Jeśli zostanie podana ta opcja SubWCRev wyjdzie z `ERRORLEVEL 7` jeśli kopia robocza zawiera lokalne modyfikacje. Może to być wykorzystane do zapobiegania kompilacji z obecnymi niezatwierdzonymi zmianami.
-N	Jeśli ta opcja zostanie podana, SubWCRev zakończy się z `ERRORLEVEL 11` jeśli kopia robocza zawiera niewersjonowane obiekty, które nie sa ignorowane.
-m	Jeśli opcja ta zostanie podana SubWCRev wyjdzie z `ERRORLEVEL 8` jeśli kopia robocza zawiera mieszane wersje. Może to być wykorzystane do zapobiegania kompilacji z częściowo aktualizowanej kopii roboczej.
-d	Jeśli opcja ta zostanie podana SubWCRev wyjdzie z `ERRORLEVEL 9`, jeżeli plik docelowy już istnieje.
-f	Jeśli opcja ta zostanie podana SubWCRev zawrze wersję ostatniego zatwierdzenia folderów. Domyślnym zachowaniem jest korzystanie tylko z plików podczas sprawdzania numerów wersji.
-e	Jeśli opcja ta zostanie podana SubWCRev przeszuka katalogi, które są dołączone do `svn:externals`, ale tylko jeśli są z tego samego repozytorium. Domyślnym zachowaniem jest ignorowanie zewnętrznych.
-E	Jeśli podano ten przełącznik, podobnie jak dla `-e`, jednak nie bierze on pod uwagę zewnętrznych ze wskazanymi bezpośrednio wersjami, gdy zakres wersji wewnątrz nich jest jedyną wersją wskazaną przez atrybuty. W ten sposób nie dochodzi do wersji mieszanych.
-x	Jeśli opcja ta zostanie podana SubWCRev wypisze numery wersji w formacie HEX.
-X	Jeśli opcja ta zostanie podana SubWCRev wypisze numery wersji w formacie HEX z przedrostkiem '0X'.
-F	Jeśli opcja ta zostanie podana SubWCRev pominie wszystkie pliki `.subwcrevignore` i doda wszystkie pliki.
-q	Jeśli ta opcja jest podana, SubWCRev wykona zamianę słów kluczowych bez pokazywania statusu kopii roboczej w stdout.

Tabela 6.1. Lista dostępnych przełączników wiersza poleceń

Jeśli nie wystąpił błąd, SubWCRev zwraca zero. Jednak gdy błąd się pojawi, komunikat błędu jest wypisywany do stderr i wyświetlany na konsoli. Zaś zwracane kody błędów to:

Kod Błędu	Opis
1	Błąd składni. Jeden lub więcej argumentów linii poleceń jest nieprawidłowy.
2	Plik lub folder wskazany w linii poleceń nie został znaleziony.
3	Nie dało się otworzyć pliku wejścia lub nie dało się utworzyć pliku docelowego.
4	Nie dało się przydzielić pamięci. Mogło się to zdarzyć jeśli na przykład plik źródłowy jest zbyt duży,
5	Plik źródłowy nie może być poprawnie zeskanowany.
6	Błąd SVN: Subversion zwróciło błąd podczas próby odnalezienia przez SubWCRev informacji z kopii roboczej.

Kod Błędu	Opis
7	Kopia robocza zawiera lokalne zmiany. Wymaga to przełącznika `-n`.
8	Kopia robocza zawiera różne wersje. Wymaga to przełącznika `-m`.
9	Plik docelowy już istnieje. Wymaga to przełącznika `-d`.
10	Wskazana ścieżka nie prowadzi do kopii roboczej ani nie stanowi jej części.
11	Kopia robocza zawiera niewersjonowane pliki lub foldery. Wymaga to ustawienia opcji `-N`.

Tabela 6.2. Lista kodów błędów SubWCRev

6.2. Zastępowanie słów kluczowych

Jeśli pliki źródłowy i docelowy są dostarczane, SubWCRev kopiuje źródło do celu, wykonując substytucji słów kluczowych w następujący sposób:

Slowo kluczowe	Opis
$WCREV$	Zastępowane przez najwyższą wersję zatwierdzenia w kopii roboczej.
$WCREV&$	Zastąpiona przez najwyższą wersję zatwierdzenia w kopii roboczej, połączona koniunkcją logiczną AND z wartością po znaku &. Na przykład: `$WCREV&0xFFFF$`
$WCREV-$, $WCREV+$	Zastąpiona przez najwyższą wersję zatwierdzenia w kopii roboczej, z wartością po + lub - odpowiednio dodaną lub odjętą. Na przykład: `$WCREV-1000$`
$WCDATE$, $WCDATEUTC$	Zastępowane przez datę/czas wykonania najwyższej wersji zatwierdzenia. Domyślnie używany jest format międzynarodowy: `yyyy-mm-dd hh:mm:ss`. Alternatywnie, można określić niestandardowy format, który będzie używany z `strftime()`, na przykład: `$WCDATE=%a %b %d %I:%M:%S %p$`. Aby uzyskać listę dostępnych znaków formatowania, spójrz na *odnośnik internetowy* [http://msdn.microsoft.com/en-us/library/fe06s4ak.aspx].
$WCNOW$, $WCNOWUTC$	Zastępowane przez datę/czas systemowy. Może zostać wykorzystane do wskazywania czasu kompilacji. Można zastosować formatowanie jak opisano dla `$WCDATE$`.
$WCRANGE$	Zastępowane przez zakres wersji aktualizacji z kopii roboczej. Jeśli kopia robocza jest w stanie spójnym, będzie to jedna wersja. Jeśli kopia robocza zawiera różne zmiany, albo ze względu na nieaktualność, albo z powodu celowej aktualizacji do wersji, to zakres zostanie pokazany w formacie 100:200.
$WCMIXED$	`$WCMIXED?TText:FText$` otrzymuje wartość `TText` jeśli występują mieszane wersje aktualizacji albo `FText`, jeśli nie.
$WCMODS$	`$WCMODS?TText:FText$` otrzymuje wartość `TText` jeśli występują lokalne modyfikacje albo `FText`, jeśli nie.
$WCUNVER$	`$WCUNVER?TText:FText$` jest zastąpione przez `TText` jeśli w kopii roboczej znajdują się niewersjonowane obiekty, lub `FText` jeśli nie.
$WCEXTALLFIXED$	`$WCEXTALLFIXED?TText:FText$` jest zastępowane przez `TText` jeśli wszystkie zewnętrzne są osadzone w określonej wersji lub przez `FText` w przeciwnym przypadku.
$WCISTAGGED$	`$WCISTAGGED?TText:FText$` jest zastępowane przez `TText` jeśli adres URL repozytorium zawiera znaczniki wzorca klasyfikacji lub przez `FText` w przeciwnym przypadku.
$WCURL$	Zastąpione przez URL repozytorium dla ścieżki kopii roboczej przekazanej do SubWCRev.

Słowo kluczowe	Opis
$WCINSVN$	`$WCINSVN?TText:FText$` otrzymuje wartość `TText` jeśli wejście jest wersjonowane albo `FText`, jeśli nie.
$WCNEEDSLOCK$	`$WCNEEDSLOCK?TText:FText$` otrzymuje wartość `TText` jeśli wejście ma ustawiony atrybut `svn:needs-lock` albo `svn:needs-lock`, jeśli nie.
$WCISLOCKED$	`$WCISLOCKED?TText:FText$` otrzymuje wartość `TText` jeśli wejście jest zablokowane albo `FText`, jeśli nie.
$WCLOCKDATE$, $WCLOCKDATEUTC$	Zastępowane datą blokady. Formatowanie może być dostosowane jak opisano dla `$WCDATE$`.
$WCLOCKOWNER$	Zastępowane przez nazwę właściciela blokady.
$WCLOCKCOMMENT$	Zastępowane komentarzem blokady.
$WCUNVER$	`$WCUNVER?TText:FText$` został zastąpiony przez `TText` jeśli w kopii roboczej znajdują się niewersjonowane pliki lub foldery, lub `FText` jeśli nie.

Tabela 6.3. Lista dostępnych słów kluczowych

SubWCRev nie obsługuje bezpośrednio zagnieżdżania wyrażeń, więc nie można na przykład używać wyrażeń takich jak:

```
#define SVN_REVISION "$WCMIXED?$WCRANGE$:$WCREV$$"
```

Ale zazwyczaj można to obejść w inny sposób, na przykład:

```
#define SVN_RANGE $WCRANGE$
#define SVN_REV $WCREV$
#define SVN_REVISION "$WCMIXED?SVN_RANGE:SVN_REV$"
```

Podpowiedź

Niektóre z tych słów kluczowych stosuje się do pojedynczych plików, a nie do całej kopii roboczej, więc ma to sens tylko do korzystania z tych, kiedy SubWCRev jest wywołany do skanowania pojedynczego pliku. Dotyczy to `$WCINSVN$`, `$WCNEEDSLOCK$`, `$WCISLOCKED$`, `$WCLOCKDATE$`, `$WCLOCKOWNER$` i `$WCLOCKCOMMENT$`.

6.3. Przykład słowa kluczowego

Poniższy przykład pokazuje, jak słowa kluczowe z pliku szablonu są podstawiane w pliku wynikowym.

```
// Plik testowy dla SubWCRev

char *Revision      = "$WCREV$";
char *Revision16    = "$WCREV&0xFF$";
char *Revisionp100  = "$WCREV+100$";
char *Revisionm100  = "$WCREV-100$";
char *Modified      = "$WCMODS?Modyfikowany:Nie modyfikowany$";
char *Unversioned   = "$WCUNVER?Znaleziono niewersjonowane obiekty:brak niewersjonowany
char *Date          = "$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         = "$WCMIXED?Wersja mieszana WC:Nie mieszana$";
char *ExtAllFixed   = "$WCEXTALLFIXED?Wszystkie zewnętrzne naprawione:Nie wszystkie zew
char *IsTagged      = "$WCISTAGGED?Tagowane:Nie tagowane$";
char *URL           = "$WCURL$";
char *isInSVN       = "$WCINSVN?wersjonowane:nie wersjonowane$";
char *needslck      = "$WCNEEDSLOCK?TRUE:FALSE$";
char *islocked      = "$WCISLOCKED?zablokowane:nie zablokowane$";
char *lockdateutc   = "$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
```

Po uruchomieniu `SubWCRev.exe path\to\workingcopy testfile.tmpl testfile.txt`, plik wyjściowy `testfile.txt` będzie wyglądał tak:

```
// Plik testowy dla SubWCRev

char *Revision  = "22837";
char *Revision16 = "53";
char *Revisionp100 = "22937";
char *Revisionm100 = "22737";
char *Modified  = "Modyfikowany";
char *Unversioned = "brak niewersjonowanych danych";
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  = "Wersja mieszana WC";
char *ExtAllFixed = "Wszystkie zewnętrzne naprawione";
char *IsTagged  = "Nie tagowane";
char *URL = "https://tortoisesvn.googlecode.com/svn/trunk";
char *isInSVN = "wersjonowane";
char *needslck = "FALSE";
char *islocked = "nie zablokowany";
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
```

Podpowiedź

Pliki takie jak ten zostaną włączone do kompilacji więc można oczekiwać, iż będą wersjonowane. Upewnijcie się, by wersjonować plik szablonu, a nie wygenerowany plik, w przeciwnym razie za każdym razem po regeneracji pliku wersji musicie zatwierdzić zmianę, co z kolei oznacza, że plik wersji musi zostać zaktualizowany.

6.4. Interfejs COM

Jeśli potrzebujecie dostępu do informacji wersji Subversion z innych programów, możecie skorzystać z interfejsu COM SubWCRev. Należy utworzyć obiekt `SubWCRev.object`, obsługujący następujące metody:

Metoda	Opis
.GetWCInfo	Ta metoda przeszukuje kopię roboczą gromadząc informacje o wersji. Oczywiście należy ją wykonać, zanim będzie można uzyskać dostęp do informacji przy użyciu pozostałych metod. Pierwszy parametr to ścieżka. Drugi parametr powinien być true, jeśli chcecie dołączyć wersje folderów. Odpowiednik przełącznika `-f` dla linii poleceń. Trzeci parametr powinien być true, jeśli chcecie dołączyć svn:externals. Odpowiednik przełącznika `-e` dla linii poleceń.
.GetWCInfo2	To samo co `GetWCInfo()` ale z czwartym parametrem ustawiającym równoważnik przełącznika `-E` linii poleceń.
.Revision	Najwyższa zatwierdzona wersja w kopii roboczej. Odpowiednik `$WCREV$`.
.Date	Data/czas zatwierdzenia dla najwyższej wersji zatwierdzenia. Odpowiednik `$WCDATE$`.
.Author	Autor najwyższej wersji zatwierdzenia, to jest ostatnia osoba, która dokonała zmian w kopii roboczej.
.MinRev	Minimalna wersja aktualizacji, jak pokazano w `$WCRANGE$`
.MaxRev	Maksymalna wersja aktualizacji, jak pokazano w `$WCRANGE$`
.HasModifications	Prawdziwe jeśli występują lokalne zmiany
.HasUnversioned	Prawda jeśli znajdują się tu niewersjonowane obiekty
.Url	Zastępowane przez URL repozytorium dla ścieżki kopii roboczej użytej w `GetWCInfo`. Odpowiednik `$WCURL$`.
.IsSvnItem	Prawdziwe jeśli element jest wersjonowany.
.NeedsLocking	Prawdziwe jeśli element ma ustawiony atrybut `svn:needs-lock`.
.IsLocked	Prawdziwe jeśli element jest zablokowany.
.LockCreationDate	Ciąg znaków reprezentujący datę kiedy utworzono blokadę lub pusty ciąg znaków jeśli element nie jest zablokowany.
.LockOwner	Ciąg znaków reprezentujący właściciela blokady lub pusty ciąg znaków jeśli element nie jest zablokowany.
.LockComment	Wiadomość wprowadzana podczas nakładania blokady.

Tabela 6.4. COM/wspierane metody automatyzacji

Poniższy przykład pokazuje, jak interfejs może być używany.

```
// testCOM.js - plik javascript
// skrypt testowy obiektu SubWCRev COM/Automation

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);
```

Poniższy listing jest przykładem, jak korzystać z obiektu COM SubWCRev w 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");
}
```

Rozdział 7. Interfejs IBugtraqProvider

Aby uzyskać ściślejszą integrację z trackerami problemów niż po prostu za pomocą atrybutów `bugtraq:`, TortoiseSVN daje możliwość skorzystania z wtyczek COM. Za pomocą takich wtyczek jest możliwe pobieranie informacji bezpośrednio z systemu śledzenia błędów, interakcja z użytkownikiem i przekazywanie informacji o otwartych wydaniach z powrotem do TortoiseSVN, sprawdzanie opisów zmian wprowadzonych przez użytkownika, a uruchamianie działań po udanym zatwierdzeniu np., w zamknięcie zagadnienia.

Nie możemy dostarczyć informacji i samouczków, w jaki sposób zaimplementować obiekt COM w wybranym języku programowania, jednak dołączyliśmy wtyczki przykładowe w C++/ATL i C# w folderze repozytorium `contrib/issue-tracker-plugins`. W tym folderze można znaleźć również wymagane pliki potrzebne do kompilacji wtyczki. (Sekcja 3, „Licencja” wyjaśnia, jak uzyskać dostęp do repozytorium.)

Ważne

Należy dostarczyć zarówno 32-bitowe i 64-bitowej wersje wtyczki. Ponieważ wersja x64 TortoiseSVN nie może używać 32-bitowej wtyczki i vice-versa.

7.1. Konwencje nazewnictwa

Jeśli wydajecie wtyczkę trackera problemów dla TortoiseSVN, prosimy *nie* nazywać jej *Tortoise<Coś tam>*. Chcielibyśmy zarezerwować przedrostek *Tortoise* dla klienta kontroli wersji zintegrowanego z powłoką Windows. Na przykład: TortoiseCVS, TortoiseSVN, TortoiseHg, TortoiseGIT i TortoiseBzr to wszystko klienty kontroli wersji.

Prosimy nazywać wtyczki dla klienta Tortoise *Turtle<Coś tam>*, gdzie *<Coś tam>* odnosi się do systemu śledzenia błędów, z którym się łączycie. Można wybrać nazwę, która brzmi jak *Turtle*, ale ma inną pierwszą literę. Dobrymi przykładami są:

- Gurtle - Plugin trackera problemów dla Google code
- TurtleMine - Plugin trackera problemów dla Redmine
- VurtleOne - Plugin trackera problemów dla VersionOne

7.2. Interfejs IBugtraqProvider

TortoiseSVN 1.5 i następne mogą korzystać z wtyczek, które implementują interfejs IBugtraqProvider. Interfejs zapewnia kilka metod, które wtyczki mogą używać do interakcji z trackerem wydań.

```
HRESULT ValidateParameters (
  // Okno nadrzędne dla dowolnego UI niezbędnego do
  // wyświetlenia podczas walidacji.
  [in] HWND hParentWnd,

  // Ciąg znaków parametru który ma być sprawdzony.
  [in] BSTR parameters,

  // Czy ciąg znaków jest poprawny?
  [out, retval] VARIANT_BOOL *valid
);
```

Metoda ta wywoływana jest w oknie ustawień, gdzie użytkownik może dodać i skonfigurować wtyczkę. Ciąg znaków `parameters` może być używany przez wtyczkę, aby uzyskać dodatkowe wymagane informacje, np. adres URL do śledzenia zgłoszeń, danych logowania itp. Wtyczka powinna sprawdzić ciąg znaków `parameters` i wyświetlić okno błędu, jeśli ciąg nie jest poprawny. Parametr `hParentWnd` powinien być używany dla każdego

okna dialogowego który wtyczka pokazuje jako okno rodzica. Wtyczka musi zwrócić wartość TRUE, jeśli sprawdzenie ciągu znaków `parameters` powiedzie się. Jeśli plugin zwraca FALSE, okno ustawień nie pozwoli użytkownikowi dodać wtyczki do ścieżki na kopii roboczej.

```
HRESULT GetLinkText (
  // Okno nadrzędne dla dowolnego (error) UI niezbędnego do wyświetlenia.
  [in] HWND hParentWnd,

  // Ciąg znaków parametru, na wypadek konieczności porozumienia z
  // serwisem (np.) sprawdzić jaki jest poprawny tekst.
  [in] BSTR parameters,

  // Jaki tekst trzeba wyświetlić?
  // Użyj bieżącej lokalizacji wątku.
  [out, retval] BSTR *linkText
);
```

Wtyczka może tutaj dostarczyć ciągu znaków, który jest używany w oknie zatwierdzenia TortoiseSVN jako etykieta przycisku, który wywołuje wtyczkę, np. "Wskaż problem" lub "Wybierz bilet". Upewnijcie się, że napis nie jest zbyt długi, w przeciwnym razie może nie pasować do przycisku. Jeśli metoda zwraca błąd (np. `E_NOTIMPL`), na przycisku zostanie wyświetlony tekst domyślny.

```
HRESULT GetCommitMessage (
  // Okno nadrzędne dla UI dostawcy.
  [in] HWND hParentWnd,

  // Parametry dla dostawcy.
  [in] BSTR parameters,
  [in] BSTR commonRoot,
  [in] SAFEARRAY(BSTR) pathList,

  // Tekst już obecny w opisie zmiany.
  // Dostawca powinien zawrzeć ten tekst w nowym opisie,
  // gdzie należy.
  [in] BSTR originalMessage,

  // Nowy tekst dla opisu zmiany.
  // Zastępuje oryginalny opis.
  [out, retval] BSTR *newMessage
);
```

Jest to główna metoda wtyczki. Metoda ta wywoływana jest w oknie dialogowym zatwierdzenia TortoiseSVN, gdy użytkownik kliknie na przycisk wtyczki.

Ciąg znaków `parameters` użytkownik musi wpisać w oknie ustawień, gdy konfiguruje plugin. Zazwyczaj wtyczki może wykorzystać go, aby znaleźć adres URL systemu śledzenia błędów i/lub informacji autoryzacyjnych albo więcej.

Ciąg znaków `commonRoot` zawiera ścieżkę nadrzędną wszystkich wybranych elementów, aby otworzyć okno dialogowe zatwierdzenia. Należy pamiętać, że *nie* jest to ścieżka do katalogu głównego wszystkich elementów, które użytkownik wybrał w oknie dialogowym zatwierdzenia. Dla okna gałęzi/etykiety, to jest ścieżka, która ma być skopiowana.

Parametr `pathList` zawiera tablicę ścieżek (jako ciągi znaków) które użytkownik wybrał do zatwierdzenia.

Parametr `OriginalMessage` zawiera tekst wprowadzony w oknie opisu zmiany w oknie zatwierdzenia. Jeśli użytkownik nie wprowadził jeszcze żadnego tekstu, napis ten będzie pusty.

Zwracany ciąg znaków `NewMessage` jest kopiowany do pola edycji opisu zmiany w oknie zatwierdzenia, zastępując to co już wprowadzono. Jeśli wtyczka nie zmienia ciągu znaków `originalMessage`, musi ona zwrócić ten sam ciąg tutaj, w przeciwnym razie tekst wpisany przez użytkownika zostanie utracony.

7.3. Interfejs IBugtraqProvider2

W TortoiseSVN 1.6 został dodany nowy interfejs, który zapewnia większą funkcjonalność wtyczek. Ten interfejs IBugtraqProvider2 dziedziczy z IBugtraqProvider.

```
HRESULT GetCommitMessage2 (
  // Okno nadrzędne dla UI dostawcy.
  [in] HWND hParentWnd,

  // Parametry dla dostawcy.
  [in] BSTR parameters,
  // Wspólny URL zatwierdzenia
  [in] BSTR commonURL,
  [in] BSTR commonRoot,
  [in] SAFEARRAY(BSTR) pathList,

  // Tekst już obecny w opisie zmiany.
  // Dostawca powinien zawrzeć ten tekst w nowym opisie,
  // gdzie należy.
  [in] BSTR originalMessage,

  // Można przypisać niestandardowe właściwości wersji do zatwierdzenia
  // przez ustawienie następnych dwóch parametrów.
  // uwaga: Obie tabele safearray muszą mieć tą samą długość.
  //        Dla każdej nazwy atrybutu musi tam być wartość atrybutu!

  // Zawartość pola bugID (jeśli nadano)
  [in] BSTR bugID,

  // Zmieniona zawartość pola bugID
  [out] BSTR * bugIDOut,

  // Lista nazw atrybutów wersji.
  [out] SAFEARRAY(BSTR) * revPropNames,

  // Lista wartości atrybutów wersji.
  [out] SAFEARRAY(BSTR) * revPropValues,

  // Nowy tekstdla opisu zmiany.
  // Zastępuje oryginalny opis
  [out, retval] BSTR * newMessage
);
```

Metoda ta wywoływana jest w oknie dialogowym zatwierdzenia TortoiseSVN, kiedy użytkownik kliknie na przycisk wtyczki. Jest ona wywoływana zamiast `GetCommitMessage()`. Proszę zapoznać się z dokumentacją `GetCommitMessage` dotyczącą parametrów, które są tam również używane.

Parametr `commonURL` jest nadrzędnym URL wszystkich wybranych elementów, aby otworzyć okno dialogowe zatwierdzenia. Jest to po prostu adres URL dla ścieżki `commonRoot`.

Parametr `bugID` zawiera treści pola ID błędu (jeżeli jest pokazywana, skonfigurowana w atrybucie `bugtraq:message`).

Parametr zwrotny `bugIDOut` jest używany do wypełnienia pola ID błędu, gdy zwracany przez metodę.

Parametry zwrotne `RevPropNames` i `revPropValues` mogą zawierać pary nazwa/wartość dla atrybutów wersji, które zatwierdzenie powinno ustawić. Wtyczka musi upewnić się, że obie tablice mają ten sam rozmiar podczas zwracania! Każda nazwa własności `revPropNames` musi mieć odpowiednią wartość w `revPropValues`. Jeśli nie ma atrybutów wersji do ustawienia, wtyczka musi zwrócić puste tablice.

```
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
);
```

Ta metoda jest wywoływana tuż przed zamknięciem okna dialogowego zatwierdzenia i rozpoczęciem samego zatwierdzenia. Wtyczka może użyć tej metody do sprawdzania wybranych plików/folderów do zatwierdzenia i/lub opisu zmiany wprowadzonego przez użytkownika. Parametry są takie same jak dla `GetCommitMessage2()`, z tą różnicą, że `commonURL` jest wspólnym URL wszystkich *zaznaczonych* elementów, a `commonRoot` ścieżką do katalogu głównego wszystkich zaznaczonych elementów.

Dla okna gałęzi/etykiety `commonURL` jest adres URL źródła kopiowania, a `commonRoot` jest ustawiony na adres docelowy kopii.

Parametr zwrotny `ErrorMessage` musi albo zawierać komunikat o błędzie, który TortoiseSVN pokazuje użytkownikowi albo być pusty by rozpoczęło się zatwierdzenie. Jeśli został zwrócony komunikat o błędzie, TortoiseSVN pokazuje okno błędu i jednocześnie utrzymuje okno zatwierdzenia otwarte więc użytkownik może poprawić to co jest niepoprawne. Wtyczka powinna zwracać zapis błędu, który informuje użytkownika, *co* jest nie tak i jak to poprawić.

```
HRESULT  OnCommitFinished (
  // Okno nadrzędne dla dowolnego (error) UI niezbędnego do wyświetlenia.
  [in] HWND hParentWnd,

  // Wspólny korzeń dla wszystkich ścieżek jakie zostaną zatwierdzone.
  [in] BSTR commonRoot,

  // Wszystkie ścieżki do zatwierdzenia.
  [in] SAFEARRAY(BSTR) pathList,


  // Tekst już wprowadzony do opisu zmiany.
  [in] BSTR logMessage,

  // wersja zatwierdzenia.
  [in] ULONG revision,


  // Błąd do pokazania użytkownikowi jeśli funkcja
  // zwróci coś innego niż S_OK
  [out, retval] BSTR * error
);
```

Metoda ta wywoływana jest po udanym zatwierdzeniu. Wtyczka może użyć tej metody do np. zamknięcia wybranego zagadnienia lub dodania informacji o zatwierdzeniu się do wydania. Parametry są takie same jak dla `GetCommitMessage2`.

```
HRESULT HasOptions(
  // Czy dostawca przewiduje opcje
  [out, retval] VARIANT_BOOL *ret
);
```

Metoda ta wywoływana jest w oknie ustawień, gdzie użytkownik może skonfigurować wtyczki. Jeśli wtyczka udostępnia własne okno konfiguracyjne z `ShowOptionsDialog`, musi ona zwrócić tutaj TRUE, w przeciwnym wypadku musi zwrócić FALSE.

```
HRESULT ShowOptionsDialog(
  // Okno nadrzędne dla okna opcji
  [in] HWND hParentWnd,

  // Parametry dla dostawcy.
  [in] BSTR parameters,

  // Ciąg znaków parametrów
  [out, retval] BSTR * newparameters
);
```

Metoda ta wywoływana jest w oknie ustawień, gdy użytkownik kliknie na przycisk "Opcje", który jest wyświetlany, jeśli `HasOptions` zwraca TRUE. Wtyczka może wyświetlić okno opcji, aby ułatwić użytkownikowi konfigurację wtyczki.

Ciąg `parameters` zawiera ciąg parametrów wtyczki, który jest już ustawiony/wprowadzony.

Parametr zwrotny `newparameters` musi zawierać ciąg parametrów, które wtyczka utworzyła z informacji zgromadzonych w oknie dialogowym Opcje. To ciąg znaków `paramameters` przekazywany jest do wszystkich innych IBugtraqProvider i metod IBugtraqProvider2.

Dodatek A. Często zadawane pytania (FAQ)

Ponieważ TortoiseSVN jest cały czas rozwijany czasami trudno jest prowadzić w pełni aktualną dokumentację. Utrzymujemy *internetowe FAQ* [https://tortoisesvn.net/faq.html], które zawiera wybór pytań najczęściej zadawanych na grupie TortoiseSVN list mailowych <dev@tortoisesvn.tigris.org> i <users@tortoisesvn.tigris.org>.

Prowadzimy także projekt *Tracker Problemów* [https://sourceforge.net/p/tortoisesvn/tickets/], który zawiera lwią część naszej listy rzeczy do zrobienia, i błędy, które zostały już poprawione. Jeżeli uważasz, że znalazłeś błąd lub chcesz poprosić o nową funkcję, sprawdź najpierw tutaj, czy ktoś inny był tu już przed tobą.

Jeśli macie pytanie, na które nie ma odpowiedzi gdzie indziej, najlepiej poprosić na jednej z list mailowych:

- Użyjcie adresu <users@tortoisesvn.tigris.org>, jeśli macie pytania dotyczące korzystania z TortoiseSVN.
- Jeśli chcecie pomóc w rozwoju TortoiseSVN, to należy wziąć udział w dyskusjach na <dev@tortoisesvn.tigris.org>.
- Jeśli chcecie pomóc w tłumaczeniu interfejsu użytkownika TortoiseSVN lub w dokumentacji, wyślijcie e-mail do <translators@tortoisesvn.tigris.org>.

Dodatek B. Jak to zrobić...

Ten dodatek zawiera rozwiązania problemów/pytań możecie mieć podczas używania TortoiseSVN.

B.1. Przenieś/kopiuj wiele plików na raz

Przenoszenie/kopiowanie pojedynczych plików można zrobić za pomocą TortoiseSVN → Zmień nazwę.... Ale jeśli chcecie przenieść/skopiować wiele plików, w ten sposób jest po prostu zbyt wolny i zbyt pracochłonny.

Zalecanym sposobem jest przeciągnięcie prawym przyciskiem myszy plików do nowej lokalizacji. Wystarczy kliknąć prawym przyciskiem myszy na pliki, które chcecie przenieść/skopiować, i nie zwalniać przycisku myszy. Następnie przeciągnąć pliki do nowej lokalizacji i zwolnić przycisk myszy. Pojawi się menu kontekstowe, gdzie możecie wybrać Menu kontekstowe → SVN Kopiuj wersjonowany(-e) plik(i) tutaj. lub Menu kontekstowe → SVN Przenieś wersjonowany(-e) plik(i) tutaj.

B.2. Zmuszenie użytkowników do wprowadzenia opisu zmiany

Istnieją dwa sposoby, aby uniemożliwić użytkownikom zatwierdzenie z pustym opisem zmiany. Jeden z nich jest specyficzny dla TortoiseSVN, inny działa dla wszystkich klientów Subversion, ale wymaga dostępu bezpośrednio do serwera.

B.2.1. Skrypt przechwytujący na serwerze

Jeśli macie bezpośredni dostęp do serwera repozytorium, można zainstalować skrypt pre-commit, który odrzuca wszystkie zatwierdzenia z pustym lub zbyt krótkim opisem zmiany.

W folderze repozytorium na serwerze znajduje się podfolder `hooks`, który zawiera niektóre przykładowe skrypty przechwytujące, do wypróbowania. Plik `pre-commit.tmpl` zawiera przykładowy skrypt, który będzie odrzucać zatwierdzenia, jeśli opis zmian nie został dostarczony lub jest zbyt krótki. Plik zawiera również uwagi na temat jak zainstalować/użyć tego skryptu. Postępujcie zgodnie z instrukcjami w pliku.

Metoda ta jest zalecanym sposobem, jeśli użytkownicy korzystają również z innych klientów Subversion niż TortoiseSVN. Wadą jest to, że zatwierdzenie jest odrzucane przez serwer i dlatego użytkownicy otrzymują komunikat o błędzie. Klient nie może wiedzieć przed zatwierdzeniem, że będzie ono odrzucone. Jeśli chcecie, aby TortoiseSVN miał przycisk OK wyłączony, dopóki opis zmiany nie jest wystarczająco długi, prosimy skorzystać z metody opisanej poniżej.

B.2.2. Atrybuty projektu

TortoiseSVN wykorzystuje atrybuty do niektórych swoich funkcji. Jednym z tych atrybutów jest `tsvn:logminsize`.

Po ustawieniu tego atrybutu na folder, TortoiseSVN wyłączy przycisk OK we wszystkich oknach dialogowych zatwierdzenia, dopóki użytkownik nie wprowadził jeszcze opisu zmiany co najmniej długości określonej przez atrybut.

Szczegółowe informacje na temat tych atrybutów projektu, znajdują się w Sekcja 4.18, „Ustawienia projektu”.

B.3. Aktualizacja wybranych plików z repozytorium

Zazwyczaj uaktualnia się kopię roboczą za pomocą TortoiseSVN → Uaktualnij. Ale jeśli chcecie tylko wybrać kilka nowych plików, które kolega dodał nie wykonując scalania żadnych zmian w innych plikach w tym samym czasie, trzeba innego podejścia.

Użyjcie TortoiseSVN → Sprawdź zmiany. i kliknijcie Sprawdź repozytorium, aby zobaczyć co się zmieniło w repozytorium. Wybierzcie pliki, które chcecie zaktualizować lokalnie, a następnie użyjcie menu kontekstowego, aby zaktualizować tylko te pliki.

B.4. Wycofywanie (Cofnij) zmiany w repozytorium

B.4.1. Okno dialogowe dziennika wersji

Jak dotąd najprostszą metodą wycofania zmian z jednej lub wielu wersji jest użycie okna dziennika wersji.

1. Wybierzcie plik lub folder, w którym trzeba cofnąć zmiany. Jeśli chcecie wycofać wszystkie zmiany, należy być w katalogu najwyższego poziomu.
2. Wybierzcie TortoiseSVN → Pokaż dziennik, aby wyświetlić listę wersji. Być może trzeba użyć Wszystkie lub Następne 100, aby pokazać wersję(-e), która Was interesuje.
3. Należy wybrać wersję, do której należy wycofać. Jeśli chcecie wycofać zakres wersji, wybierzcie pierwszą i przytrzymać klawisz **Shift** podczas wybierania ostatniej. Jeśli macie zamiar wybrać pojedyncze wersji i zakresy, używajcie klawisza **Ctrl** przy wyborze wersji. Kliknięcie prawymprzyciskiem myszy na wybranej wersji(ach), po czym wybór Menu Kontekstowe → Wycofaj zmiany z tej wersji.
4. Jeżeli chcecie, aby wcześniej wersja stała się nową wersją HEAD, kliknijcie prawym przyciskiem myszy na wybraną wersję, a następnie wybierzcie Menu kontekstowe → Wycofaj zmiany do tej wersji. Zostaną odrzucone *wszystkie* zmiany po wybranej wersji.

Masz wycofane zmiany w kopii roboczej. Sprawdźcie wyniki, a następnie zatwierdźcie zmiany.

B.4.2. Użycie okna scalenia

Jeśli potrzeba wprowadzić numery wersji w postaci listy, można użyć okna Scalenie. Poprzednia metoda wykorzystuje sclaenie w tle; ta metoda korzysta z niego bezpośrednio.

1. W swojej kopii roboczej wybierzcie TortoiseSVN → Scalaj.
2. W oknie Typ Scalenia należy wybrać Scalenie zakresu wersji.
3. W polu Z: należy wprowadzić pełny adres URL folderu kopii roboczej. Powinien on pojawić się jako domyślny URL.
4. W polu Zakres wersji doscalenia wprowadźcie listęwersji do wycofania (lub użycie okna dziennika by wybrać je jak toopisanopowyżej).
5. Upewnijcie się, że pole wyboru Odwrócone scalenie jest zaznaczone.
6. W oknie Opcje scalenia potwierdźcie wartości domyślne.
7. Kliknijcie Scal by zakończyć scalenie.

Przywróciliście zmiany w kopii roboczej. Sprawdźcie, czy wyniki są takie jak się spodziewaliście,po czym zatwierdźcie zmiany.

B.4.3. Użycie `svndumpfilter`

Ponieważ TortoiseSVN nigdy nie traci danych, wasze „wycofane” wersje nadal istnieją jako pośrednie wersje w repozytorium. Tylko wersja HEAD została zmieniona do poprzedniego stanu. Jeśli chcecie, aby zmiany zniknęły całkowicie z repozytorium wraz z wymazaniem śladów, że kiedykolwiek istniały, trzeba użyć bardziej skrajnych środków. O ile nie ma naprawdę dobrego powodu, aby to zrobić, jest to *nie zalecane*. Jednym z możliwych powodów byłoby to, że ktoś zatwierdził poufny dokument do publicznego repozytorium.

Jedynym sposobem usunięcia danych z repozytorium jest użycie narzędzia linii poleceń Subversion `svnadmin`. Pełny opis jego działania można znaleźć przechodząc na *Utrzymanie Repozytorium* [http://svnbook.red-bean.com/en/1.8/svn.reposadmin.maint.html].

B.5. Porównanie dwóch wersji pliku lub folderu

Jeśli chcecie porównać dwie historyczne wersje elementu, na przykład wersje 100 i 200 tego samego pliku, wystarczy użyć TortoiseSVN → Pokaż dziennik by wyświetlić historię zmian dla tego pliku. Wybrać dwie wersje do porównania a następnie wykorzystać Menu kontekstowe → Porównaj wersje.

Jeśli chcecie porównać ten sam element z dwóch różnych drzew, na przykład linii głównej i gałęzi, można skorzystać z przeglądarki repozytorium, aby otworzyć oba drzewa, wybrać plik w obu miejscach, a następnie użyć Menu kontekstowe → Porównaj Wersje.

Jeśli chcecie porównać dwa drzewa, aby zobaczyć co się zmieniło, na przykład linię główną i otagowaną wersję, można użyć TortoiseSVN → Pokaż wykres wersji Wybrać dwa węzły do porównania, a następnie wybrać Menu kontekstowe → Porównaj wersje główne (HEAD). Zostanie wyświetlona lista zmienionych plików, a następnie można wybrać pojedyncze pliki, aby wyświetlić szczegóły zmian. Można także eksportować strukturę drzewa zawierającą wszystkie zmienione pliki lub po prostu listę wszystkich zmienionych plików. Czytajcie Sekcja 4.11.3, „Porównanie folderów” by uzyskać więcej informacji. Można używać Menu kontekstowe → Plik różnicowy głównych (HEAD) wersji, aby zobaczyć podsumowanie wszystkich różnic, z minimalnym kontekstem.

B.6. Dołączanie wspólnego podprojektu

Niekiedy potrzeba zawrzeć inny projekt w kopii roboczej, może tu chodzi o kod biblioteki. Mamy co najmniej 4 sposoby poradzenia sobie z tym.

B.6.1. Użycie svn:externals

Ustawcie atrybut `svn:externals` dla folderu w projekcie. Ten atrybut składa się z jednej lub więcej linii, każda linia zawiera nazwę podfolderu, którego chcecie użyć jako folderu pobrania dla wspólnego kodu oraz adres URL repozytorium, z którego mają być robione pobrania. Aby uzyskać szczegółowe informacje sprawdźcie Sekcja 4.19, „Elementy zewnętrzne”.

Zatwierdźcie nowy folder. Teraz po aktualizacji, Subversion będzie zaczytywać kopię tego projektu z jego repozytorium do kopii roboczej. Podfoldery będą w razie potrzeby tworzone automatycznie. Podczas każdej aktualizacji podstawowej kopii roboczej, otrzymacie również najnowsze wersje wszystkich zewnętrznych projektów.

Jeśli zewnętrzny projekt leży w tym samym repozytorium, wszelkie zmiany w nim wprowadzone będą włączone do listy zatwierdzenia, jeśli użytkownik zatwierdza główny projekt.

Jeśli zewnętrzny projekt znajduje się w innym repozytorium, wszelkie zmiany wprowadzone do zewnętrznego projektu zostaną pokazane lub oznaczone podczas zatwierdzenia głównego projektu, jednak wymagane jest osobne zatwierdzenie tych zewnętrznych zmian.

Z trzech opisanych metod, ta jest jedyną, która nie wymaga instalacji po stronie klienta. Po określeniu zewnętrznych we właściwości folderu, wszystkie klienty dostaną zapełnione foldery podczas aktualizacji.

B.6.2. Użycie zagnieżdżonej kopii roboczej

Utwórzcie nowy folder w ramach projektu aby zawierał wspólny kod, ale nie dodawajcie go do Subversion.

Wybierzcie TortoiseSVN → Pobierz dla nowego folderu i pobierzcie do niego kopię wspólnego kodu. Teraz macie oddzielną kopię roboczą zagnieżdżoną w głównej kopii roboczej.

Dwie kopie robocze są niezależne. Kiedy dokonujecie zmian w rodzicu, zmiany zagnieżdżonej KR są ignorowane. Podobnie po aktualizacji rodzica, zagnieżdżona KR nie jest aktualizowana.

B.6.3. Użycie względnego położenia

Jeśli używacie tego samego wspólnego jądra kodu w kilku projektach, a nie chcecie przechowywać wielu kopii roboczych to dla każdego projektu, który go używa, wystarczy je pobrać w innym miejscu, które jest powiązane z wszystkimi innymi projektami, które z niego korzystają. Na przykład:

```
C:\Projects\Proj1
C:\Projects\Proj2
C:\Projects\Proj3
C:\Projects\Common
```

i odnosić się do wspólnego kodu przy użyciu ścieżki względnej, np. `..\..\Common\DSPcore`.

Jeśli projekty są rozproszone w niezależnych miejscach można użyć wariantu, polegającego na umieszczeniu wspólnego kodu w jednym miejscu, wykonaniu substytucji napędu dla tej lokalizacji, by otrzymać literę dysku do wpisania na sztywno w kodzie projektów, np. Pobierzcie wspólny kod do `D:\Documents\Framework` lub `C:\Documents and Settings\{login}\My Documents\framework` następnie wykonajcie

```
SUBST X: "D:\Documents\framework"
```

aby utworzyć mapowanie dysku używanego w kodzie źródłowym. Wasz kod może wtedy użyć absolutnej lokalizacji.

```
#include "X:\superio\superio.h"
```

Ta metoda działa tylko w środowisku wszystkich PC, i trzeba będzie udokumentować wymagane mapowania dysków tak, by wasz zespół wiedział, gdzie znajdują się te tajemnicze pliki. Metoda ta jest do zastosowania wyłącznie w zamkniętych środowiskach programistycznych, i nie zalecana do ogólnego użytku.

B.6.4. Dodanie projektu do repozytorium

Prawdopodobnie najprostszym sposobem jest dodanie projektu w podfolderze własnego projektu w kopii roboczej. Jednak ma to taką wadę, że trzeba aktualizować i uzupełniać ten projekt zewnętrzny ręcznie.

Aby pomuc w aktualizacji, TortoiseSVN posiada polecenie w menu kontekstowym przesunięcia prawym klawiszem. PO prostu przesuń prawym przyciskiem myszy folder z rozpakowaną nową wersją biblioteki zewnętrznej do katalogu w kopii roboczej, a następnie wybierz Menu Kontekstowe → Tu GałąźDostawcy SVN. Spowoduje to skopiowanie nowych plików do folderu docelowego automatycznie dodając nowe pliki i usuwając pliki nie występujące już w nowej wersji.

B.7. Tworzenie skrótu do repozytorium

Jeśli potrzebujecie często otwierać przeglądarkę repozytorium w określonym miejscu, można utworzyć skrót na pulpicie za pomocą interfejsu automatyzacji TortoiseProc. Wystarczy utworzyć nowy skrót i ustawić element docelowy:

```
TortoiseProc.exe /command:repobrowser /path:"url/do/repozytorium"
```

Oczywiście trzeba podać prawdziwy adres URL repozytorium.

B.8. Ignorowanie plików, które już są pod kontrolą wersji

Jeśli przypadkowo dodano kilka plików, które powinny zostać zignorowane, jak się ich pozbyć z kontroli wersji bez utraty? A może masz własny plik konfiguracyjny IDE, który nie jest częścią projektu, ale który wymagał długiej konfiguracji do odpowiadającej Wam postaci.

Jeśli nie zatwierdziliście jeszcze dodania, wszystko co musicie zrobić, to użyć TortoiseSVN → Cofnij dodanie... by cofnąć operację dodania. Następnie należy dodać plik(i) do listy ignorowanych aby nie mógł zostać dodany później przez pomyłkę.

Jeśli pliki znajdują się już w repozytorium, muszą zostać usunięte z repozytorium i dodane do listy ignorowanych. Na szczęście TortoiseSVN posiada wygodny skrót do tego celu. TortoiseSVN → Odwersjonuj i dodaj do ignorowanych najpierw zaznacza plik/folder do usunięcia z repozytorium, pozostawiając lokalną kopię. Dodaje również ten element do listy ignorowanych tak, że nie zostanie dodany z powrotem do Subversion przez pomyłkę. Po wykonaniu tych czynności wystarczy zatwierdzić folder nadrzędny.

B.9. Usunięcie kontroli wersji z kopii roboczej

Jeśli macie kopię roboczą, którą chcecie przekonwertować z powrotem na zwykłe drzewa folderów bez katalogów `.svn`, możecie po prostu wyeksportować je do siebie. Czytajcie Sekcja 4.27.1, „Usunięcie kopii roboczej z kontroli wersji” aby dowiedzieć się jak.

B.10. Usunięcie kopii roboczej

Jeśli macie kopię roboczą, która nie jest już potrzebna, w jaki sposób się jej elegancko pozbyć? Proste - wystarczy tylko usunąć ją w eksploratorze Windows! Kopie robocze są prywatnymi jednostkami lokalnymi i są samodzielne. Usuwanie kopii roboczej w Eksploratorze Windows nie ma w ogóle wpływu na dane w repozytorium.

Dodatek C. Użyteczne porady dla administratorów

Ten dodatek zawiera rozwiązania problemów/pytań, które pojawiają się, gdy jest się odpowiedzialnym za wdrażanie TortoiseSVN na wielu komputerach klienckich.

C.1. Wdrażanie TortoiseSVN poprzez zasady grup

Instalator TortoiseSVN przychodzi jako plik MSI, co oznacza, że nie powinno być problemów podczas dodania pliku MSI do zasad grupy kontrolera domeny.

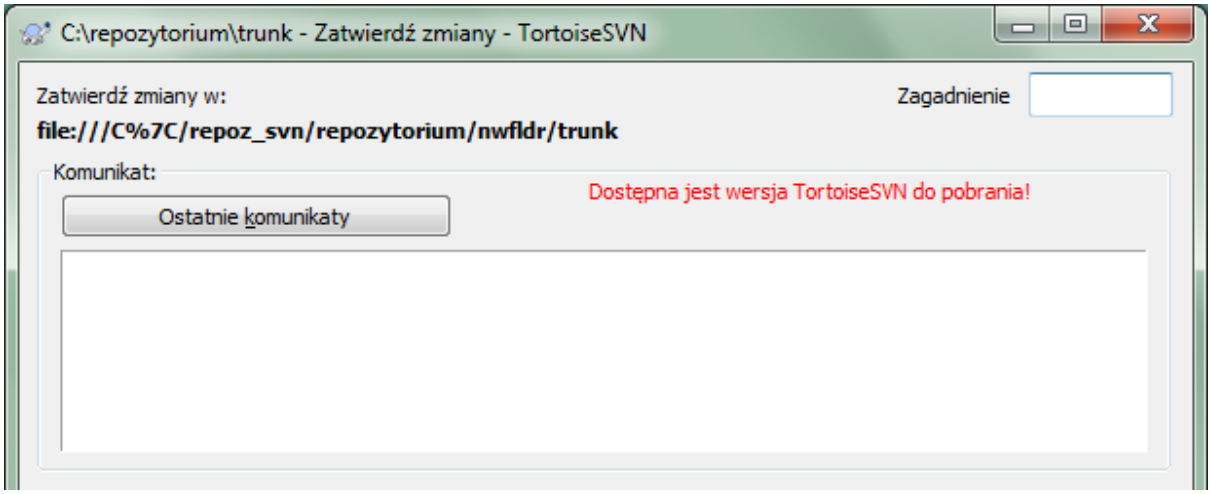
Dobrą solucję jak to zrobić można znaleźć w artykule 314934 bazy wiedzy firmy Microsoft: http://support.microsoft.com/?kbid=314934.

TortoiseSVN musi zostać zainstalowany w obszarze *Konfiguracja komputera*, a nie *Konfiguracja użytkownika*. To dlatego, że TortoiseSVN wymaga bibliotek CRT i DLLi MFC, które mogą być rozmieszczane tylko *na komputer* a nie *na użytkownika*. Jeśli naprawdę trzeba zainstalować TortoiseSVN użytkownikom z osobna, należy najpierw zainstalować MFC i wersję 12 pakietu CRT Microsoft na każdym komputerze, na którym należy zainstalować TortoiseSVN użytkownikom.

Można dostosować plik MSI jeśli trzeba, by wszyscy wasi użytkownicy otrzymali te same ustawienia. Ustawienia TSVN są przechowywane w rejestrze pod `HKEY_CURRENT_USER\Software\TortoiseSVN` a podstawowe ustawienia Subversion (od których zależą klienci Subversion) są przechowywane w plikach konfiguracji `%APPDATA%\Subversion`. Jeśli potrzebujecie pomocy przy dostosowaniu MSI, skorzystajcie z jednego z forów transformacji MSI lub przeszukajcie sieć względem „MSI transform”.

C.2. Przekierowanie sprawdzenia nowej wersji

TortoiseSVN sprawdza czy jest już nowa wersja co kilka dni. Jeśli jest dostępna nowsza wersja, jest wyświetlane powiadomienie w oknie dialogowym zatwierdzenia.



Rysunek C.1. Okno zatwierdzenia, wyświetlanie powiadomienia o nowej wersji

Jeśli jesteś odpowiedzialny za wielu użytkowników w domenie, możesz wymóc na użytkownikach do korzystania tylko z wersji zatwierdzonej przez siebie a nie zawsze żądać instalowania najnowszej wersji. Prawdopodobnie nie życzysz sobie wyświetlania powiadomienia o aktualizacji, aby użytkownicy nie wykonywali aktualizacji natychmiast.

Wersja 1.4.0 TortoiseSVN i późniejsze pozwalają przekierować sprawdzenie nowych wersji na serwer intranetowy. Można ustawić klucz rejestru `HKCU\Software\TortoiseSVN\UpdateCheckURL` (wartość

ciągu znaków) na adres URL wskazujący plik tekstowy w intranecie. Ten plik tekstowy musi mieć następujący format:

```
1.9.1.6000
Nowa wersja TortoiseSVN jest dostępna do pobrania!
http://192.168.2.1/downloads/TortoiseSVN-1.9.1.6000-svn-1.9.1.msi
```

Pierwsza linia to zapis numeru wersji. Trzeba upewnić się, że jest ona zgodna z właściwym numerem wersji pakietu instalacyjnego TortoiseSVN. Następna linia to niestandardowy tekst, wyświetlany w oknie zatwierdzenia. Można wpisać, co się chce. Należy jednak pamiętać, że przestrzeń w oknie zatwierdzenia jest ograniczona. Zbyt długi komunikat zostanie obcięty! Trzecia linia to adres URL nowego pakietu instalacyjnego. Ten URL jest otwierany gdy użytkownik kliknie na etykiecie niestandardowej wiadomości w oknie zatwierdzenia. Można również przekierować użytkownika na stronę w sieci zamiast bezpośrednio do pliku MSI. Adres URL jest otwierany w domyślnej przeglądarce, zatem wskazanie strony internetowej skutkuje otwarciem jej i pokazaniem użytkownikowi. Jeśli zostanie wskazany pakiet MSI , przeglądarka poprosi użytkownika o zapisanie lokalnie pliku MSI.

C.3. Ustawianie zmiennej środowiskowej `SVN_ASP_DOT_NET_HACK`

Począwszy od wersji 1.4.0, instalator TortoiseSVN nie zapewnia już użytkownikowi możliwość ustawienia zmiennej środowiskowej `SVN_ASP_DOT_NET_HACK`, jako że była przyczyną wielu problemów i nieporozumień dla użytkowników, którzy zawsze instalują *wszystko* bez względu, czy wiedzą, po co to jest.

Jednak funkcja jest wciąż dostępna w TortoiseSVN i innych klientach svn. By ją włączyć musicie ustawić zmienną systemową o nazwie `ASPDOTNETHACK` na `1`. Właściwie wartość zmiennej systemowej nie ma znaczenia: jeśli zmienna istnieje, funkcja jest aktywna.

Ważne

Należy pamiętać, że ten hack konieczny jest tylko wtedy, gdy nadal używasz VS.NET2002. Wszystkie późniejsze wersje Visual Studio do *nie* wymagają tej sztuczki aby być aktywowane! Więc jeśli nie używasz tak przestarzałego narzędzia, NIE UŻYWAJ TEGO!

C.4. Wyłączanie pozycji menu kontekstowego

Począwszy od wersji 1.5.0, TortoiseSVN pozwala wyłączyć (a faktycznie ukryć) pozycje w menu kontekstowym. Ponieważ jest to funkcja, której nie należy używać pochopnie, a tylko wtedy, gdy istnieje ważny powód, nie ma do tego GUI i trzeba to zrobić bezpośrednio w rejestrze. Można w ten sposób wyłączyć niektóre polecenia użytkownikom, którzy nie powinni ich używać. Ale proszimy pamiętać, że tylko wpisy menu kontekstowego w *eksploratorze* zostają ukryte, zaś same polecenia są nadal dostępne w inny sposób, np. z linii poleceń a nawet z innych okien dialogowych w TortoiseSVN!

Klucze rejestru, które przechowują informacje, jakie menu kontekstowe powinny zostać pokazane `HKEY_CURRENT_USER\Software\TortoiseSVN\ContextMenuEntriesMaskLow` i `HKEY_CURRENT_USER\Software\TortoiseSVN\ContextMenuEntriesMaskHigh`.

Każdy z tych wpisów rejestru to wartość `DWORD`, w której każdy bit odpowiada konkretnej pozycji menu. Ustawiony bit oznacza że odpowiednia pozycja menu jest nieaktywna.

Wartość	Pozycja menu
0x0000000000000001	Pobierz
0x0000000000000002	Uaktualnij
0x0000000000000004	Zatwierdź
0x0000000000000008	Dodaj

Wartość	Pozycja menu
0x0000000000000010	Wycofaj zmiany
0x0000000000000020	Uporządkuj
0x0000000000000040	Rozwiąż konflikt
0x0000000000000080	Przełącznik
0x0000000000000100	Import
0x0000000000000200	Eksport
0x0000000000000400	Twórz repozytorium tutaj
0x0000000000000800	Gałąź/etykieta
0x0000000000001000	Scalanie
0x0000000000002000	Usuń
0x0000000000004000	Zmień nazwę
0x0000000000008000	Uaktualnij do wersji
0x0000000000010000	Porównaj
0x0000000000020000	Pokaż dziennik
0x0000000000040000	Edytuj konflikty
0x0000000000080000	Zmień lokalizację
0x0000000000100000	Sprawdź zmiany
0x0000000000200000	Ignoruj
0x0000000000400000	Przeglądarka repozytorium
0x0000000000800000	Adnotuj
0x0000000001000000	Twórz plik poprawek
0x0000000002000000	Zastosuj poprawkę
0x0000000004000000	Wykres wersji
0x0000000008000000	Blokada
0x0000000010000000	Odblokuj
0x0000000020000000	Atrybuty
0x0000000040000000	Porównaj z URL
0x0000000080000000	Usuń niewersjonowane elementy
0x0000000100000000	Scalaj wszystko
0x0000000200000000	Porównaj z poprzednią wersją
0x0000000400000000	Wklej
0x0000000800000000	Konwertuj kopię roboczą
0x0000001000000000	Porównaj później
0x0000002000000000	Porównaj z 'nazwa pliku'
0x0000004000000000	Plik różnicowy
0x2000000000000000	Ustawienia
0x4000000000000000	Pomoc
0x8000000000000000	Informacje o

Tabela C.1. Pozycje menu i ich wartości

Przykład: aby wyłączyć pozycje menu „Zmień lokalizację”, „Usuń niewersjonowane elementy” i „Ustawienia”, należy dodać wartości przypisane do pozycji w ten sposób:

```
0x0000000000080000
+ 0x0000000080000000
+ 0x2000000000000000
= 0x2000000080080000
```

Niższa wartość `DWORD` (`0x80080000`) musi być przechowywana w `HKEY_CURRENT_USER\Software\TortoiseSVN\ContextMenuEntriesMaskLow`, zaś wyższa wartość `DWORD` (`0x20000000`) w `HKEY_CURRENT_USER\Software\TortoiseSVN\ContextMenuEntriesMaskHigh`.

Aby włączyć pozycje w menu, wystarczy usunąć te dwa klucze rejestru.

Dodatek D. Automatyzacja TortoiseSVN

Ponieważ wszystkie polecenia TortoiseSVN są kontrolowane przez parametry wiersza polecenia, można zautomatyzować za pomocą skryptów wsadowych lub uruchamiać konkretne polecenia i okna dialogowe z innych programów (np. ulubionego edytora tekstu).

Ważne

Pamiętaj, że TortoiseSVN jest GUI klienta, a ten przewodnik automatyki pokazuje jak wywołać okna dialogowe TortoiseSVN aby gromadzić dane wprowadzone przez użytkownika. Jeśli chcesz napisać skrypt, który nie wymaga żadnego wejścia, należy użyć oficjalnego klienta linii poleceń Subversion.

D.1. Polecenia TortoiseSVN

Program GUI TortoiseSVN nosi nazwę `TortoiseProc.exe`. Wszystkie polecenia są określone przez parametr `/command:abcd`, gdzie `abcd` jest wymaganą nazwą polecenia. Większość z tych poleceń wymaga co najmniej jednego argumentu ścieżki, który jest podawany przez `/path:"jakaś\ścieżka"`. W poniższej tabeli polecenie odnosi się do parametru `/command:abcd` a ścieżka odnosi się do parametru `/path:"jakaś\ścieżka"`.

Jest specjalne polecenie nie wymagające parametru `/command:abcd`, gdy jednak nie wskazano nic w linii poleceń, jest uruchamiany monitor projektu. Jeżeli podano opcję `/tray`, monitor projektu uruchamiany jest ukryty i dodaje tylko ikonę w zasobniku systemowym.

Ponieważ niektóre polecenia mogą przyjmować listę ścieżek docelowych (np. zatwierdzenie kilku konkretnych plików) parametr `/path` może potrwać kilka ścieżek oddzielonych znakiem `*`.

Można również określić plik, który zawiera listę ścieżek oddzielonych znakami nowej linii. Plik musi być w formacie UTF-16, bez *BOM* [https://en.wikipedia.org/wiki/Byte-order_mark]. eśli wskażecie taki plik, użyjcie `/pathfile` zamiast `/path`. Aby TortoiseProc usunął ten plik po wykonaniu polecenia, można przekazać parametr `/deletepathfile`. Bez wpisania `/deletepathfile`, konieczne jest usunięcie ręczne tego pliku lub pozostanie on po operacji.

Okno dialogowe postępu, które jest używana do zatwierdzenia, aktualizacji i wielu innych poleceń zwykle pozostaje otwarte po zakończeniu polecenia dopóki użytkownik nie naciśnie przycisku OK. Można to zmienić poprzez zaznaczenie odpowiedniej opcji w oknie ustawień. Ale użycie tego ustawienia spowoduje zamknięcie okna dialogowego postępu, niezależnie czy uruchomiono polecenie z pliku wsadowego czy z menu kontekstowego TortoiseSVN.

Aby określić inną lokalizację pliku konfiguracji, użyć należy parametru `/configdir:"ścieżka\do\folderu\konfiguracji"`. Nadpisze to domyślną ścieżkę, w tym ustawienia rejestru.

Aby zamknąć automatycznie okno postępu na końcu polecenia bez wykorzystania stałych ustawień można przekazać parametr `/closeonend`.

- `/closeonend:0` nie zamykać okna automatycznie
- `/closeonend:1` zamknąć automatycznie jeśli brak błędów
- `/closeonend:2` zamknąć automatycznie jeśli brak błędów i konfliktów
- `/closeonend:3` zamknąć automatycznie jeśli brak błędów, konfliktów i scaleń

Aby zamknąć okno dialogowe postępu dla operacji lokalnych, jeśli nie ma błędów i konfliktów, należy przekazać parametr `/closeonend`.

Poniższa tabela zawiera listę wszystkich poleceń, które można wykonać za pomocą wiersza poleceń TortoiseProc.exe. Jak opisano powyżej, powinny one być stosowane w postaci `/command:abcd`. Wewnątrz tabeli przedrostek `/command` jest pomijany w celu zaoszczędzenia miejsca.

Polecenie	Opis
:about	Pokazuje okno informacji o programie. Jest ono pokazywane również jeśli nie podano polecenia.

:log	Otwiera okno dziennika. `/path` wskazuje plik lub folder dla którefo ma zostać wyświetlony dziennik. Mogą zostać ustawione dodatkowe opcje:
	• `/startrev:xxx`,
	• `/endrev:xxx`,
	• `/strict` włącza pole wyboru 'zatrzymaj-na-kopii',
	• `/merge` włącza pole wyboru 'włącznie z wersjami scalonymi',
	• `/datemin:"{ciągdaty}"` ustawia datę początkową filtrowania, a
	• `/datemax:"{ciągdaty}"` ustawia datę końcową filtrowania. Format daty jest taki sam jak używany przez svn dla dat wersji.
	• `/findstring:"filtrowanyciag"` wypełnia filtrowany tekst,
	• `/findtext` wymusza na filtrze wyszukiwanie tekstu a nie wyrażenia regularnego lub
	• `/findregex` wymusza na filtrze użycie wyrażenia regularnego a nie zwykłego wyszukiwania tekstowego, a
	• `/findtype:X` gdzie X to numer pomiędzy 0 a 511. Liczby te stanowią sumę następujących opcji:
	  • `/findtype:0` filtrować według wszystkiego
	  • `/findtype:1` filtrować według wiadomości
	  • `/findtype:2` filtrować według ścieżek
	  • `/findtype:4` filtrować według autorów
	  • `/findtype:8` filtrować według rewizji
	  • `/findtype:16` niewykorzystane
	  • `/findtype:32` filtrować według ID błędów
	  • `/findtype:64` niewykorzystane
	  • `/findtype:128` filtrować według daty
	  • `/findtype:256` filtrować według zakresu dat
	• Jeśli wskazano `/outfile:sciezka\do\pliku`, wskazane wersje zostaną zapisane dowskazanego pliku po zamknięciu okna. Wersje są zapisywane w tym samym formacie jaki został użyty w oknie scalania.
	An svn date revision can be in one of the following formats:
	• {2006-02-17}
	• {15:30}
	• {15:30:00.200000}

Polecenie	Opis
	• {"2006-02-17 15:30"} • {"2006-02-17 15:30 +0230"} • {2006-02-17T15:30} • {2006-02-17T15:30Z} • {2006-02-17T15:30-04:00} • {20060217T1530} • {20060217T1530Z} • {20060217T1530-0500}
:checkout	Otwiera okno dialogowe pobrania. `/path` określa katalog docelowy a `/url` określa adres URL, z którego wykonywane jest pobranie. Jeśli podasz klucz `/blockpathadjustments`, automatyczne korekty ścieżki pobierania są blokowane. `/revision:XXX` określa wersje do pobrania.
:import	Otwiera okno importu. `/path` określa katalog z danymi do importu. Można również określić przełącznik `/logmsg` by przekazać predefiniowane opisy zmian do okna importu. Lub, jeśli nie chce się przekazać opisu zmiany z linii poleceń, należy użyć `/logmsgfile:ścieżka`, gdzie `ścieżka` wskazuje na plik zawierający opis zmiany.
:update	Aktualizuje kopię roboczą w `/path` do HEAD. Jeśli dodano opcję `/rev` to wyświetlane jest okno dialogowe z pytaniem użytkownika, do której wersji zastosować uaktualnienie. By uniknąć okienka wskaż numer wersji `/rev:1234`. Inne opcje to `/nonrecursive`, `/ignoreexternals` i `/ignoreexternals`. `/stickydepth` wskazuje, że głębokość powinna być określona ściśle, tworząc rzadkie pobieranie. Można ustawić `/skipprechecks` by pominąć wszystkie sprawdzenia wykonywane przed aktualizacją. Jeśli go wskazano, zostaje zablokowany przycisk Pokaż dziennik, zaś menu kontekstowe pokazywania różnic pozostaje również zablokowane po aktualizacji.
:commit	Otwiera okno dialogowe zatwierdzenia. `/path` określa katalog docelowy lub listę plików do zatwierdzenia. Można również określić przełącznik `/logmsg` dla przekazania predefiniowanego opisu zmiany do okna zatwierdzenia. Lub, jeśli nie chcecie przekazać opisu zmiany z linii poleceń, należy użyć `/logmsgfile:ścieżka`, gdzie `ścieżka` wskazuje na plik zawierający opis zmiany. Aby wstępnie wypełnić pole ID błędu (w przypadku prawidłowej konfiguracji integracji z trackerami błędów), można użyć `/bugid:"tutaj id błędu"`.
:add	Dodaje pliki z `/path` do kontroli wersji.
:revert	Wycofuje lokalne modyfikacje kopii roboczej. `/path` informuje, jakie elementy przywrócić.
:cleanup	Usuwa przerwane lub anulowane polecenia i odblokowuje kopię roboczą w `/path`. Konieczne jest wykonanie `/cleanup` by przeprowadzić rzeczywiste czyszczenie. Użyjcie `/noui` aby zapobiec pojawieniu się okna wyniku (mówiącego o zakończeniu oczyszczenia albo wyświetlającego komunikat błędu). `/noprogressui` wyłącza również okno dialogowe postępu. `/nodlg` wyłącza wyświetlenie okna oczyszczania, w którym użytkownik może wybrać, gdzie dokładnie należy wykonać uporządkowanie. Dostępne działania mogą być określone przy użyciu opcji `/cleanup` dla statusu uporządkowania, `/breaklocks` by złamać wszystkie blokady, `/revert` by

Polecenie	Opis
	wycofać niezatwierdzone zmiany, `/delunversioned`, `/delignored`, `/refreshshell`, `/externals`, `/fixtimestamps` and `/vacuum`.
:resolve	Oznacza skonfliktowany plik podany w `/path` jako rozwiązany. Jeśli jest podane `/noquestion`, wtedy rozwiązanie jest wykonywane bez poprzedzającego pytania, czy naprawdę należy to zrobić.
:repocreate	Tworzy repozytorium w `/path`
:switch	Otwiera okno przełączenia. Folder `/path` określa miejsce docelowe zaś `/url` to adres URL do przełączenia.
:export	Eksportuje kopię roboczą z `/path` do innego katalogu. Jeśli `/path` wskazuje na niewersjonowany folder, zostaniecie zapytani o adres URL do eksportu do katalogu z `/path`. Jeśli podacie klucz `/blockpathadjustments`, automatyczna korekta ścieżki eksportu zostanie zablokowana.
:dropexport	Eksportuje kopię roboczą z `/path` do folderu wskazanego w `/droptarget`. To działanie nie wykorzystuje okna dialogowego eksportu, ale wykonuje eksport bezpośrednio. Opcja `/overwrite` wskazuje, że istniejące pliki zostaną nadpisane bez potwierdzenia, zaś opcja `/autorename`, że jeśli pliki istnieją, pliki eksportowane zostaną przemianowane bu uniknąć nadpisania. Opcja `/extended` może zawierać `localchanges` by wyeksportować tylko pliki zmienione lokalnie, albo `unversioned` by wyeksportować również wszystkie elementy niewersjonowane.
:dropvendor	Copies the folder in `/path` recursively to the directory specified in `/droptarget`. New files are added automatically, and missing files get removed in the target working copy, basically ensuring that source and destination are exactly the same. Specify `/noui` to skip the confirmation dialog, and `/noprogressui` to also disable showing the progress dialog.
:merge	Otwiera okno łączenia. `/path` wskazuje na folder docelowy. Przy łączeniu zakresu rewizji, dostępne są następujące opcje: `/fromurl:URL`, `/revrange:ciąg`. Przy łączeniu dwóch drzew repozytoriów, dostępne są następujące opcje: `/fromurl:URL`, `/tourl:URL`, `/fromrev:xxx` i `/torev:xxx`.
:mergeall	Otwiera okno dialogowe scalenia wszystkiego. `/path` wskazuje folder docelowy.
:copy	Wywołuje okno dialogowe gałęzi/etykiety. Zapis `/path` określa kopię roboczą, z której kopiowana jest gałąź/etykieta. Natomiast `/url` to docelowy adres URL. Jeśli adres URL zaczyna się od ^ przyjmuje się, że jest on względny w odniesieniu do folderu głównego repozytorium. Aby wstępnie zaznaczyć opcję <placeholder-4> możecie przekazać przełącznik `/switchaftercopy`. Aby zaznaczyć opcję <placeholder-6> przekazać należy przełącznik `/makeparents`. Można również podać przełącznik `/logmsg` by przesłać w linii poleceń predefiniowany komunikat dzienika do okna gałęzi/etykiety. Istnieje też możliwość, jeśli nie chcecie pisać komunikatów w linii poleceń, użycia `/logmsgfile:ścieżka`, gdzie `ścieżka` wskazuje na plik zawierający komunikat dziennika.</placeholder-6> </placeholder-4>
:settings	Otwiera okno dialogowe ustawień.
:remove	Usuwa plik(i) w `/path` z kontroli wersji.
:rename	Zmienia nazwę pliku w `/path`. Nowa nazwa dla pliku, jest wymagana w oknie dialogowym. Aby uniknąć pytania o zmianę nazwy podobnych plików w jednym kroku, podaj `/noquestion`.
:diff	Uruchamia zewnętrzne narzędzie porównania wskazane w ustawieniach TortoiseSVN. `/path` wskazuje pierwszy plik. jeśli ustawiono opcję `/path2`, program porównujący uruchamia się z tymi dwoma plikami. jeżeli pominięto `/path2`, porównywanie wykonuje się pomiędzy plikiem z `/path` a jego

Polecenie	Opis
	BASE. Jeśli wskazany plik ma również zmienione atrybuty, zewnętrzne narzędzie porównywania jest odpalane dla każdego zmienionego atrybutu. By do tego nie dopuścić, przekażcie opcję `/ignoreprops`. By jawnie wskazać numery wersji należy podać `/startrev:xxx` i `/endrev:xxx`, a przy opcjonalnej wersji wieszakowej użyć `/pegrevision:xxx`. Jeżeli ustawiono `/blame` a nie wskazano `/path2`, to porównywanie zostaje zakończone przez adnotowanie na początku plików ze wskazanymi wersjami. Parametr `/line:xxx` wskazuje linię od której zaczyna się wyświetlać porównywanie.
:shelve	Shelves the specified paths in a new shelf. The option `/shelfname:name` specifies the name of the shelf. An optional log message can be specified with `/logmsg:message`. If option `/checkpoint` is passed, the modifications of the files are kept.
:unshelve	Applies the shelf with the name `/shelfname:name` to the working copy path. By default the last version of the shelf is applied, but you can specify a version with `/version:X`.
:showcompare	W zależności od URLi i wersji do porównania, pokazywane są albo plik różnicowy (jeśli ustawiona jest opcja `unified`), okno z listą plików, które uległy zmianie albo jeśli adresy URL wskazują pliki, uruchamia przeglądarkę różnic dla tych dwóch plików. Opcje `url1`, `url2`, `revision1` i `revision2` muszą być określone. Opcje `pegrevision`, `ignoreancestry`, `blame` i `unified` są opcjonalne. Jeżeli wskazany adres url ma również zmienione atrybuty, zewnętrzne narzędzie różnicujące zostanie uruchomione również dla każdego zmienionego atrybutu. By temu zapobiec, wykonajcie z opcją `/ignoreprops`. If a unified diff is requested, an optional `prettyprint` option can be specified which will show the merge-info properties in a more user readable format.
:conflicteditor	Uruchamia edytor konfliktu określony w ustawieniach TortoiseSVN z odpowiednimi plikami dla pliku konfliktu w `/path`.
:relocate	Otwiera okno zmiany lokalizacji. `/path` określa ścieżkę kopii roboczej do przeniesienia.
:help	Otwiera plik pomocy.
:repostatus	Otwiera okno dialogowe sprawdź zmiany. `/path` określa katalog w kopii roboczej. Jeśli jest określony `/remote`, okno łączy się z repozytorium bezpośrednio po starcie, jak gdyby użytkownik kliknął przycisk Sprawdź repozytorium.
:repobrowser	Uruchamia okno przeglądarki repozytorium, wskazujące na adres URL kopii roboczej podany w `/path` lub `/path` wskazuje bezpośrednio na adres URL. Dodatkowa opcja `/rev:xxx` może być używana do określenia wersji, którą przeglądarka repozytorium powinna pokazać. Jeśli `/rev:xxx` jest pominięta, przyjmowana jest domyślnie HEAD. Jeśli `/path` wskazuje na adres URL, `/projectpropertiespath:ścieżka/do/kr` określa ścieżkę, z której należy odczytać i użyć właściwości projektu. Jeśli jest określone `/outfile:ścieżka\do\pliku`, wybrany adres URL i wersja są zapisywane do tego pliku, gdy przeglądarka repozytorium jest zamknięta. Pierwsza linia w tym pliku tekstowym zawiera adres URL, druga linia wersję w formacie tekstowym.

Polecenie	Opis
:ignore	Dodaje wszystkie cele w `/path` do listy ignorowanych, tzn. dodaje własność `svn:ignore` do tych plików.
:blame	Otwiera okno dialogowe adnotacji dla pliku określonego w `/path`. Jeśli są ustawione opcje `/startrev` i `/endrev`, nie jest pokazywane okno dialogowe z pytaniem o zakres adnotacji, używane są za to wartości wersji tych opcji. Jeśli jest ustawiona opcja `/line:nnn`, TortoiseBlame otworzy się wyświetlając określony numer linii. Opcje `/ignoreeol`, `/ignorespaces` i `/ignoreallspaces` są również obsługiwane.
:cat	Zapisuje plik z adresu URL lub ścieżki w kopii roboczej podany w `/path` do lokalizacji podanej w `/savepath:ścieżka`. Wersja jest podana w `/revision:xxx`. Można tego użyć pobrania pliku o określonej wersji.
:createpatch	Creates a patch file for the path given in `/path`. To skip the file Save-As dialog you can pass `/savepath:ścieżka` to specify the path where to save the patch file to directly. To prevent the unified diff viewer from being started showing the patch file, pass `/noview`. If a unified diff is requested, an optional `prettyprint` option can be specified which will show the merge-info properties in a more user readable format.
:revisiongraph	Pokazuje wykres wersji dla ścieżki podanej w `/path`. Aby utworzyć plik obrazu z wykresu wersji dla określonej ścieżki, ale bez wyświetlania okna wykresu, przekazuje się `/output:ścieżka` ze ścieżką do pliku wyjściowego. Plik wyjściowy musi mieć rozpoznawalne rozszerzenie, by wykres wersji mógł zostać poprawnie wyeksportowany. Są to: `.svg`, `.wmf`, `.png`, `.jpg`, `.bmp` oraz `.gif`. Ponieważ wykres wersji posiada wiele opcji, które wpływają na sposób jego wyświetlenia, można również ustawić opcje podczas tworzenia pliku obrazu wyjściowego. Można przekazać te ustawienia w `/options:XXXX`, gdzie `XXXX` jest wartością dziesiętną. Najlepszym sposobem, aby znaleźć potrzebne ustawienia, jest uruchomienie wykresu wersji w zwykły sposób, ustawić wszystkie opcje interfejsu użytkownika i zamknąć wykres. Wtedy opcje wymagane do przekazania w linii poleceń można odczytać z rejestru `HKCU\Software\TortoiseSVN\RevisionGraphOptions`.
:lock	Blokuje plik lub wszystkie pliki w katalogu podanym w `/path`. Wyświetlane jest okno 'Blokada', tak więc użytkownik może wpisać komentarz dla blokady.
:unlock	Odblokowuje plik lub wszystkie pliki w katalogu podanym w `/path`.
:rebuildiconcache	Odbudowuje bufor ikony windows. Należy go używać tylko w przypadku gdy ikony windows są uszkodzone. Efektem ubocznym tego (nie do uniknięcia) jest zmiana ikon na pulpicie. Aby wyłączyć okno komunikatu, przekażcie `/noquestion`.
:properties	Pokazuje okno atrybutów dla ścieżki podanej w `/path`. Aby działać na wersjonowanych atrybutach to polecenie wymaga kopii roboczej. Atrybuty wersji można przeglądać/zmieniać jeśli `/path` to URL i wskazano `/rev:XXX`. Aby otworzyć okno dialogowe atrybutów bezpośrednio dla określonego atrybutu, przekaż nazwę atrybutu w postaci `/property:nazwa`.

Polecenie	Opis
:sync	Eksportuje/importuje ustawienia, albo gdy bieżące lub wyeksportowane ustawienia się zmienią albo jak określono. Jeśli hasło przekazano wraz z `/path`, ścieżka zostaje użyta do przechowywania i odczytu ustawień. Parametr `/askforpath` pokaże okno otwarcia/zapisu pliku by użytkownik mógł wskazać ścieżkę eksportu/importu. Jeśli nie zostały wpisane ani `/load` ani `/save`, TortoiseSVN określa czy należy wykonać eksport czy import ustawień na sprawdzając, który z plików jest późniejszy. Jeśli plik eksportu jest późniejszy niż bieżące ustawienia, ustawienia są wczytywane z pliku. Jeżeli bieżące ustawienia są późniejsze, są one eksportowane do pliku ustawień. Jeśli podano `/load`, ustawienia są importowane z pliku ustawień. Jeśli wpisano `/save`, bieżące ustawienia zostają wyeksportowane do pliku ustawień. Parametr `/local` wymusza wyeksportowanie ustawień z wraz z ustawieniami lokalnymi, tj tymi, które odwołują się do ścieżek lokalnych.

Tabela D.1. Lista dostępnych poleceń i opcji

Przykłady (każdy powinien być wpisany w jednej linii):

```
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. Uchwyt URL Tsvncmd

Możliwe jest również korzystając ze specjalnych adresów URL, aby wywołanie TortoiseProc ze strony internetowej.

TortoiseSVN rejestruje nowy protokół `tsvncmd:`, który może być wykorzystywany do tworzenia hiperłączy wykonujących polecenia TortoiseSVN. Polecenia i parametry są takie same jak podczas automatyzowania TortoiseSVN z linii poleceń.

Format adresu URL `tsvncmd:` wygląda następująco:

```
tsvncmd:command:cmd?parameter:paramvalue?parameter:paramvalue
```

z `cmd` będącym jednym z dozwolonych poleceń, `parameter` stanowiącym nazwę parametru jak `path` lub `revision` oraz `paramvalue` będącym wartością do użycia w tym parametrze. Lista dozwolonych parametrów zależy od użytego polecenia.

Następujące polecenia są dozwolone w adresach URL `tsvncmd:`:

- `:update`
- `:commit`

- `:diff`
- `:repobrowser`
- `:checkout`
- `:export`
- `:blame`
- `:repostatus`
- `:revisiongraph`
- `:showcompare`
- `:log`

Prosty przykładowy adres URL może wyglądać tak:

```
<a href="tsvncmd:command:update?path:c:\svn_wc?rev:1234">Update</a>
```

lub w przypadku bardziej złożonych:

```
<a href="tsvncmd:command:showcompare?
url1:https://svn.code.sf.net/p/stefanstools/code/trunk/StExBar/src/setup/Setup.wxs?
url2:https://svn.code.sf.net/p/stefanstools/code/trunk/StExBar/src/setup/Setup.wxs?
revision1:188?revision2:189">compare</a>
```

D.3. Polecenia TortoiseIDiff

Narzędzie porównywania obrazów ma kilka opcji wiersza polecenia, których można używać do sterowania uruchamiania narzędzia. Program nosi nazwę `TortoiseIDiff.exe` .

W poniższej tabeli przedstawiono wszystkie opcje, które mogą być przekazywane do narzędzia porównywania obrazów w linii poleceń.

Opcja	Opis
:left	Ścieżka do pliku pokazanego w lewym widoku.
:lefttitle	Ciąg znaków tytułu. Ten napis jest używany w tytule widoku obrazu zamiast pełnej ścieżki dla pliku obrazu.
:right	Ścieżka do pliku pokazanego w prawym widoku.
:righttitle	Ciąg znaków tytułu. Ten napis jest używany w tytule widoku obrazu zamiast pełnej ścieżki dla pliku obrazu.
:overlay	Jeśli określony, narzędzie porównywania obrazów przełącza się na tryb nakładania (mieszanka alfa).
:fit	Jeśli określony, narzędzie porównywania obrazów dopasuje oba obrazki.
:showinfo	Pokazuje okienko informacyjne obrazu.

Tabela D.2. Lista dostępnych opcji

Przykład (który powinien być wpisany w jednej linii):

```
TortoiseIDiff.exe /left:"c:\images\img1.jpg" /lefttitle:"image 1"
```

```
/right:"c:\images\img2.jpg" /righttitle:"image 2"
/fit /overlay
```

D.4. Polecenia TortoiseUDiff

Przeglądarka plików różnicowych ma tylko dwie opcje linii poleceń:

Opcja	Opis
:patchfile	Ścieżka do pliku różnicowego.
:p	Uruchamia tryb strumieniowy. Zunifikowane różnice są odczytywane z wejścia konsoli.

Tabela D.3. Lista dostępnych opcji

Przykłady (powinny być wprowadzone każdy w jednej linii):

```
TortoiseUDiff.exe /patchfile:"c:\diff.patch"
```

Jeżeli różnica tworzona jest innym poleceniem, można użyć TortoiseUDiff by pokazać ją bezpośrednio:

```
svn diff | TortoiseUDiff.exe /u
```

działa również jeśli pominie się parametr `/p`:

```
svn diff | TortoiseUDiff.exe
```

Dodatek E. Odsyłacze interfejsu wiersza poleceń

Czasami niniejsza instrukcja odnosi się do podstawowej dokumentacji Subversion, która opisuje Subversion w zakresie Command Line Interface (CLI - interfejsu wiersza poleceń). Aby pomóc Wam zrozumieć, co TortoiseSVN robi za kulisami, zebraliśmy wykaz poleceń CLI wraz z GUI TortoiseSVN odpowiadającego każdej z operacji.

> **Uwaga**
>
> Nawet jeśli istnieją odpowiedniki CLI tego co wykonuje TortoiseSVN, pamiętajcie, że TortoiseSVN *nie* wywołuje CLI, ale korzysta bezpośrednio z biblioteki Subversion.

Jeżeli uważacie, że znaleźliście błąd w TortoiseSVN, prosimy spróbować odtworzyć go za pomocą CLI, tak abyśmy mogli odróżnić kwestie TortoiseSVN od błędów Subversion. Ten zestaw odsyłaczy określa, jakie polecenie przetestować.

E.1. Konwencje i podstawowe zasady

W następnych opisach, URL dla lokalizacji repozytorium jest nazywany po prostu `URL`, a przykładem może być `https://svn.code.sf.net/p/tortoisesvn/code/trunk/`. Ścieżkę kopii roboczej nazywa się po prostu `PATH`, co ilustruje przykład `C:\TortoiseSVN\trunk`.

> **Ważne**
>
> Ponieważ TortoiseSVN jest rozszerzeniem powłoki Windows, nie jest w stanie używać pojęcia bieżącego katalogu roboczego. Wszystkie ścieżki kopii roboczej muszą być podawana przy użyciu ścieżki bezwzględnej, a nie ścieżki względnej.

Niektóre elementy są opcjonalne i często kontrolowane przez pola wyboru czy przyciski opcji w TortoiseSVN. Opcje te są wyświetlane w [nawiasach kwadratowych] w definicjach wiersza poleceń.

E.2. Polecenia TortoiseSVN

E.2.1. Pobierz

```
svn checkout [-depth ARG] [--ignore-externals] [-r rev] URL SCIEZKA
```

Elementy menu rozwijalnego głębokość odnoszą się do argumentu `-depth`.

Jeśli Pomiń zewnętrzne jest zaznaczony, należy użyć przełącznika `--ignore-externals`.

Zaznaczenie konkretnej wersji w GUI określa wskazanie jej po URL za pomocą przełącznika `-r`.

E.2.2. Uaktualnij

```
svn info URL_z_KR
svn update [-r rev] SCIEZKA
```

Aktualizacja wielu elementów nie jest obecnie operacją atomową w Subversion. Więc TortoiseSVN na początku znajduje wersję HEAD repozytorium, a następnie uaktualnia wszystkie elementy do danego numeru wersji, aby uniknąć tworzenia kopii roboczej mieszanych wersji.

Jeśli tylko jeden element jest wybrany do aktualizacji lub wybrane elementy nie są z tego samego repozytorium, TortoiseSVN aktualizacje tylko do HEAD.

Nie są tutaj używane opcje wiersza poleceń. Uaktualnij do wersji realizuje również polecenie update, ale oferuje więcej opcji.

E.2.3. Uaktualnij do wersji

```
svn info URL_z_KR
svn update [-r rev] [-depth ARG] [--ignore-externals] SCIEZKA
```

Elementy menu rozwijalnego głębokość odnoszą się do argumentu `-depth`.

Jeśli Pomiń zewnętrzne jest zaznaczony, należy użyć przełącznika `--ignore-externals`.

E.2.4. Zatwierdź

W TortoiseSVN, okno dialogowe zatwierdzenia korzysta z kilku poleceń Subversion. Pierwszym etapem jest sprawdzenie stanu, co określa elementy w kopii roboczej, które potencjalnie mogą być zatwierdzone. Można przejrzeć listę, porównać pliki z wersją BASE i wybrać elementy, które mają być dodane do zatwierdzenia.

```
svn status -v SCIEZKA
```

Jeśli jest zaznaczone Pokaż pliki nie objęte kontrolą wersji, TortoiseSVN będzie pokazywać również wszystkie pliki i foldery bez informacji o wersji w hierarchii kopii roboczej, biorąc pod uwagę zasady ignorowania. Właśnie ta funkcja nie ma bezpośredniego odpowiednika w Subversion, jako że polecenie `svn status` nie schodzi do folderów niewersjonowanych.

Jeśli zaznaczycie jakieś niewersjonowane pliki lub foldery, te elementy zostaną najpierw dodane do kopii roboczej.

```
svn add SCIEZKA...
```

Po kliknięciu na OK, Subversion wykonuje zatwierdzenie. Jeśli zostawiono wszystkie pola wyboru plików w stanie domyślnym, TortoiseSVN używa pojedynczego rekurencyjnego zatwierdzenia kopii roboczej. Jeśli usuniecie zaznaczenie niektórych plików, wtedy musi być zastosowane nierekursywne zatwierdzenie (`-N`), a każda ścieżka musi być określona indywidualnie w linii poleceń zatwierdzenia.

```
svn commit -m "KomunDziennika" [-depth ARG] [--no-unlock] SCIEZKA...
```

`LogMessage` reprezentuje tutaj zawartość pola edycji opisu zmiany. Może ona być pusta.

Jeśli jest zaznaczone Trzymaj blokady, należy użyć przełącznika `--no-unlock`.

E.2.5. Porównaj

```
svn diff SCIEZKA
```

Jeśli używacie Porównaj z głównego menu kontekstowego, porównujecie zmodyfikowany plik z jego wersją BASE. Wyjście z polecenia CLI powyżej również to robi i generuje dane wyjściowe w formacie unified-diff. Jednak nie jest to sposób, jakiego używa TortoiseSVN. TortoiseSVN wykorzystuje TortoiseMerge (lub programu

porównującego wybranego przez użytkownika), aby wyświetlić wizualnie różnice między plikami tekstowymi, zatem nie ma tu bezpośredniego odpowiednika CLI.

Możecie także porównać dowolne 2 pliki za pomocą TortoiseSVN, czy są one pod kontrolą wersji, czy nie. TortoiseSVN tylko zasila dwoma plikami wybrany program porównujący i pozwala mu pokazać, gdzie występują różnice.

E.2.6. Pokaż dziennik

```
svn log -v -r 0:N --limit 100 [--stop-on-copy] SCIEZKA
lub
svn log -v -r M:N [--stop-on-copy] SCIEZKA
```

Domyślnie TortoiseSVN próbuje pobrać 100 opisów zmian za pomocą metody --limit. Jeśli ustawienia zalecają mu użycie starego API, to druga forma jest używana do pobierania opisów zmian dla 100 wersji repozytorium.

Jeśli Zatrzymaj na kopii/zmianie nazwy jest zaznaczony, należy użyć przełącznika `--stop-on-copy`.

E.2.7. Sprawdź zmiany

```
svn status -v SCIEZKA
lub
svn status -u -v SCIEZKA
```

Wstępne sprawdzenie stanu działa tylko na kopii roboczej. Jeśli kliknie się na Sprawdź repozytorium to repozytorium jest również sprawdzane by określić, które pliki zostaną zmienione przez aktualizację, wymaga to przełącznika `-u`.

Jeśli jest zaznaczone Pokaż pliki nie objęte kontrolą wersji, TortoiseSVN będzie pokazywać również wszystkie pliki i foldery bez informacji o wersji w hierarchii kopii roboczej, biorąc pod uwagę zasady ignorowania. Właśnie ta funkcja nie ma bezpośredniego odpowiednika w Subversion, jako że polecenie `svn status` nie schodzi do folderów niewersjonowanych.

E.2.8. Wykres wersji

Wykres wersji jest cechą wyłącznie TortoiseSVN. Nie ma odpowiednika w kliencie linii poleceń.

Polecenia wykonane przez TortoiseSVN to

```
svn info URL_z_KR
svn log -v URL
```

dzie URL jest *korzeniem* repozytorium po czym następuje analiza zwróconych danych.

E.2.9. Przeglądarka repozytorium

```
svn info URL_z_KR
svn list [-r rev] -v URL
```

Można użyć `svn info` w celu określenia głównego folderu repozytorium, co stanowi najwyższy poziom pokazany w przeglądarce repozytorium. Nie można przejść `Do góry` powyżej tego poziomu. Również to polecenie zwraca wszystkie informacje o blokadach wyświetlane w przeglądarce repozytorium.

Wywołanie `svn list` wyświetli wykaz zawartości katalogu spod wskazanego URL i wersji.

E.2.10. Edytuj konflikty

Polecenie to nie ma odpowiednika CLI. Przywołuje TortoiseMerge lub zewnętrzne 3-drożne narzędzie porównania/scalenia by przejrzeć skonfliktowane pliki i wybrać, których linii użyć.

E.2.11. Rozwiązany

```
svn resolved SCIEZKA
```

E.2.12. Zmień nazwę

```
svn rename BIEZ_SCIEZKA NOWA_SCIEZKA
```

E.2.13. Usuń

```
svn delete SCIEZKA
```

E.2.14. Wycofaj zmiany

```
svn status -v SCIEZKA
```

Pierwszym etapem jest sprawdzenie stanu, który określa elementy w kopii roboczej, potencjalnie mogące ulec przywróceniu. Można przejrzeć listy, porównać pliki z wersją BASE i wybrać elementy, które mają być zawarte w wycofaniu.

Po kliknięciu na OK, zostanie wykonane revert Subversion. Jeśli pozostawiono wszystkie pola wyboru plików w stanie domyślnym, TortoiseSVN używa pojedynczego rekurencyjnego (`-R`) przywrócenia z kopii roboczej. Jeśli usuniecie zaznaczenie z niektórych plików, każda ścieżka musi być określona indywidualnie w przywróceniu z linii poleceń.

```
svn revert [-R] SCIEZKA...
```

E.2.15. Uporządkuj

```
svn cleanup SCIEZKA
```

E.2.16. Nakładanie blokady

```
svn status -v SCIEZKA
```

Pierwszym etapem jest sprawdzenie stanu, co określa pliki w kopii roboczej potencjalnie możliwe do zablokowania. Można wybrać elementy, które należy zablokować.

```
svn lock -m "KommunDziennika" [--force] SCIEZKA...
```

`LockMessage` reprezentuje tutaj zawartość pola edycji opisu zmiany. Może ona być pusta.

Jeśli jest zaznaczone Przejmij blokady, należy użyć przełącznika `--force`.

E.2.17. Zwalnianie blokady

```
svn unlock SCIEZKA
```

E.2.18. Gałąź/etykieta

```
svn copy -m "KomunDziennika" URL URL
lub
svn copy -m "KomunDziennika" URL@wer URL@wer
lub
svn copy -m "KomunDziennika" SCIEZKA URL
```

Okno dialogowe gałęzi/etykiety wykonuje kopię do repozytorium. Dostępne są 3 przyciski opcji:

- Wersja HEAD w repozytorium
- Określona wersja w repozytorium
- Kopia robocza

które odpowiadają 3 wariantom linii poleceń opisanym powyżej.

`LogMessage` reprezentuje tutaj zawartość pola edycji opisu zmiany. Może ona być pusta.

E.2.19. Przełącznik

```
svn info URL_z_KR
svn switch [-r rev] URL SCIEZKA
```

E.2.20. Scalanie

```
svn merge [--dry-run] --force Zrodl_URL@werN Docel_URL@werM SCIEZKA
```

Testuj scalanie wykonuje takie samo scalenie z przełącznikiem `--dry-run`.

```
svn diff Zrodl_URL@werN Docel_URL@werM
```

Plik różnicowy pokazuje operację porównania, która będzie użyta do scalenia.

E.2.21. Eksport

```
svn export [-r rev] [--ignore-externals] URL Eksportow_SCIEZKA
```

Formularz ten jest wykorzystywany, podczas uruchomienia z niewersjonowanego folderu, który to folder jest używany jako miejsce docelowe.

Eksportowanie kopii roboczej do innej lokalizacji odbywa się bez wykorzystania biblioteki Subversion, więc brak pasującego odpowiednika wiersza poleceń.

TortoiseSVN wykonuje skopiowanie wszystkich plików do nowej lokalizacji pokazując jednocześnie postęp operacji. Niewersjonowane pliki/foldery mogą opcjonalnie również zostać wyeksportowane.

W obu przypadkach, jeśli jest zaznaczony Pomiń zewnętrzne, należy użyć przełącznika `--ignore-externals`.

E.2.22. Zmień lokalizację

```
svn switch --relocate Zrodl_URL Docel_URL
```

E.2.23. Twórz repozytorium tutaj

```
svnadmin create --fs-type fsfs SCIEZKA
```

E.2.24. Dodaj

```
svn add SCIEZKA...
```

Po wybraniu folderu, TortoiseSVN najpierw skanuje go rekurencyjnie dla elementów, które mogą być dodane.

E.2.25. Import

```
svn import -m KomunDziennika SCIEZKA URL
```

`LogMessage` reprezentuje tutaj zawartość pola edycji opisu zmiany. Może ona być pusta.

E.2.26. Adnotuj

```
svn blame -r N:M -v SCIEZKA
svn log -r N:M SCIEZKA
```

Jeśli używacie TortoiseBlame aby zobaczyć adnotacje, plik dziennika jest również wymagany, aby pokazać opisy zmian w dymku. Jeśli adnotacje przeglądane są w postaci pliku tekstowego, informacja ta nie jest wymagana.

E.2.27. Dodanie do listy ignorowanych

```
svn propget svn:ignore SCIEZKA > pliktymczasowy
{edycja nowego elementu ignorowanego w pliktymczasowy}
svn propset svn:ignore -F pliktymczasowy SCIEZKA
```

Ponieważ atrybut `svn:ignore` posiada często wieloliniową wartość, jest tu pokazany jako zmieniany przez plik tekstowy, a nie bezpośrednio z linii poleceń.

E.2.28. Twórz plik poprawek

```
svn diff SCIEZKA > plik-laty
```

TortoiseSVN tworzy plik poprawki w formacie pliku różnicowego poprzez porównanie kopii roboczej z wersją BASE.

E.2.29. Zastosuj poprawkę

Stosowanie poprawek nie jest taką prostą sprawą, chyba że poprawka i kopia robocza są w tej samej wersji. Na szczęście dla Was, możecie użyć TortoiseMerge, który nie ma bezpośredniego odpowiednika w Subversion.

Dodatek F. Szczegóły realizacji

Ten dodatek zawiera bardziej szczegółowe omówienie realizacji niektórych funkcji TortoiseSVN.

F.1. Ikony nakładkowe

Każdy plik i folder ma wartość statusu Subversion, zgłoszoną przez bibliotekę Subversion. W linii poleceń klienta, są one reprezentowane przez pojedyncze kody literowe, zaś w TortoiseSVN są one wyświetlane graficznie za pomocą nakładki ikony. Ponieważ liczba nakładek jest bardzo ograniczona, każda nakładka może reprezentować jedną z kilku wartości statusu.

Nakładka *w stanie konfliktu* jest używana do reprezentowania statusu `skonfliktowany`, gdzie aktualizacja lub przełączenie spowodowało konflikty między lokalnymi zmianami i zmianami pobranymi z repozytorium. Jest również stosowany w celu wskazania statusu `niedostępny`, który może wystąpić, gdy operacja nie może zostać zakończona.

Nakładka *zmodyfikowano* reprezentuje stan `zmodyfikowane`, w którym występują lokalne modyfikacje, stan `scalono`, w którym zmiany z repozytorium zostały scalone ze zmianami lokalnymi, oraz stan `zastąpiono`, w którym plik został usunięty i zastąpiony przez inny plik o tej samej nazwie.

Nakładka *usunięto* reprezentuje stan `usunięto`, w którym element jest planowany na usunięcie lub stan `nie znaleziono`, w którym element nie jest obecny. Oczywiście element którego brakuje nie może sam mieć nakładki, ale folder macierzysty może być oznaczony, jeśli brakuje jednego z jego elementów podrzędnych.

Nakładka *dodano* jest po zwykle używana do reprezentowania statusu `dodano`, gdy element został dodany do kontroli wersji.

Nakładka *w Subversion* jest używana do reprezentowania elementu, który jest w stanie `zwykły`, lub wersjonowany element, którego stan nie jest jeszcze znany. Ponieważ TortoiseSVN wykorzystuje proces buforowania w tle by gromadzić informacje o statusie, może upłynąć kilka sekund, zanim ustawi się właściwa nakładka.

Nakładka *wymaga blokady* jest używana do wskazania, kiedy plik ma ustalony atrybut `svn:needs-lock`.

Nakładka *zablokowany* jest używana, gdy lokalna kopia robocza posiada blokadę dla tego pliku.

Nakładka *ignorowany* jest używana do reprezentowania elementu, który jest w stanie `ignorowany` ze względu na globalny wzorzec ignorowania albo atrybut `svn:ignore` folderu nadrzędnego. Ta nakładka jest opcjonalna.

Nakładka *niewersjonowany* jest używana do reprezentowania elementu w stanie `niewersjonowany`. Jest to element z folderu pod kontrolą wersji, który nie jest wersjonowany. Ta nakładka jest opcjonalna.

Jeśli item ma status Subversion `brak` (elementu nie ma w kopii roboczej), nie zostanie pokazana żadna nakładka. Jeśli wybraliście zablokowanienakładek *Ignorowany* i *Niewersjonowany*, nie pokaże się również żadna nakładka również dla takich plików.

Element może mieć tylko jedną wartość stanu Subversion. Na przykład plik może być lokalnie modyfikowany i może to być zaznaczony do usunięcia w tym samym czasie. Subversion zwraca pojedynczą wartość stanu - w tym przypadku `usunięto`. Priorytety te są zdefiniowane w samym Subversion.

Kiedy TortoiseSVN wyświetla status rekursywnie (ustawienie domyślne), każdy folder wyświetla nakładkę odzwierciedlającą jego własny status i stan wszystkich jego elementów podrzędnych. W celu wyświetlenia jednej nakładki *podsumowania*, używamy kolejności priorytetów przedstawionej powyżej, aby określić, której nakładki użyć, z nakładką *w stanie konfliktu* mającą najwyższy priorytet.

W rzeczywistości może się okazać, że nie wszystkie te ikony są używane w systemie. To dlatego, że liczba nakładek dozwolonych przez system Windows jest ograniczona do 15. Windows używa 4 z nich, a pozostałe 11 może być używane przez inne aplikacje. Jeśli nie ma wystarczającej liczby dostępnych gniazd nakładek, TortoiseSVN stara się być *Dobrym Obywatelem (TM)* i ogranicza zastosowanie swoich nakładek by dać szansę innym aplikacjom.

Ponieważ są klienty Tortoise dostępne dla innych systemów kontroli wersji, stworzyliśmy wspólny komponent, który jest odpowiedzialny za pokazywanie ikon nakładek. Szczegóły techniczne nie są ważne, wszystko co musicie wiedzieć, to że ten wspólny element umożliwia wszystkim klientom Tortoise używanie tych samych nakładek, a tym samym ograniczenie do 11 wolnych gniazd nie jest zużywane przez instalację więcej niż jednego klienta Tortoise. Oczywiście jest jedna mała wada: wszystkie klienty Tortoise używają tych samych ikon nakładkek, więc nie można dowiedzieć się z ikony nakładki, jaki system kontroli wersji reprezentuje kopia robocza.

- *Zwykły*, *zmodyfikowano* i *w stanie konfliktu* są zawsze wczytywane i widoczne.
- *Usunięto* jest wczytywany jeśli możliwe ale staje się *zmodyfikowano* jeśli brak wolnych slotów.
- *Do-odczytu* jest wczytywany jeśli możliwe ale staje się *zwykły* jeśli brak wolnych slotów.
- *Zablokowany* jest wczytywany jeśli możliwe ale staje się *zwykły* jeśli brak wolnych slotów.
- *Dodano* jest wczytywany jeśli możliwe ale staje się *zmodyfikowano* jeśli brak wolnych slotów.

Dodatek G. Pakiety językowe i sprawdzenia pisowni

Standardowy instalator obsługuje tylko angielski, ale można pobrać osobne pakiety językowe i słowniki sprawdzenia pisowni oddzielnie po instalacji.

G.1. Pakiety językowe

Interfejs użytkownika TortoiseSVN został przetłumaczony na wiele różnych języków, więc można pobrać pakiet językowy odpowiedni do własnych potrzeb. Można znaleźć pakiety językowe na naszej *stronie statusów tłumaczeń* [https://tortoisesvn.net/translation_status_dev.html]. A jeśli nie jest dostępny odpowiedni pakiet językowy, to dlaczego nie dołączyć do zespołu i zgłosić własne tłumaczenie ;-)

Każdy pakiet językowy jest spakowanym instalatorem `.msi`. Wystarczy uruchomić program instalacyjny i postępować zgodnie z instrukcjami. Po zakończeniu instalacji tłumaczenie będzie dostępne.

Dokumentacja została również przetłumaczona na wiele języków. Możecie pobrać tłumaczone podręczniki ze *strony wsparcia* [https://tortoisesvn.net/support.html] na naszej stronie internetowej.

G.2. Sprawdzanie pisowni

TortoiseSVN uses the Windows spell checker if it's available (Windows 8 or later). Which means that if you want the spell checker to work in a different language than the default OS language, you have to install the spell checker module in the Windows settings (`Settings > Time & Language > Region & Language`).

TortoiseSVN will use that spell checker if properly configured with the `tsvn:projectlanguage` project property.

In case the Windows spell checker is not available, TortoiseSVN can also use spell checker dictionaries from *OpenOffice* [https://openoffice.org] and *Mozilla* [https://mozilla.org].

Instalator automatycznie dodaje słowniki języka angielskiego USA i Wielkiej Brytanii. Jeśli potrzebujecie innych językach, najprostszą opcją jest po prostu zainstalowanie jednego z pakietów językowych TortoiseSVN. To zainstaluje odpowiednie pliki słownika, jak również lokalny interfejs użytkownika TortoiseSVN. Po zakończeniu instalacji słownik będzie również dostępny.

Ewentualnie można zainstalować słowniki własnoręcznie. Jeśli macie zainstalowane OpenOffice i Mozillę, można skopiować te słowniki, które znajdują się w folderach instalacji tych aplikacji. W przeciwnym razie należy pobrać wymagane pliki słownika z http://wiki.services.openoffice.org/wiki/Dictionaries.

Gdy już macie pliki słowników, być może potrzebujecie zmienić ich nazwy by zawierały one tylko znaki lokalizacji. Przykład:

- `en_US.aff`
- `en_US.dic`

Następnie po prostu skopiujcie je do folderu `%APPDATA%\TortoiseSVN\dic`. Jeśli tego folderu tam nie ma, musicie go utworzyć. TortoiseSVN będzie również przeszukiwać podfolder `Languages` folderu instalacji TortoiseSVN (zwykle będzie to `C:\Program Files\TortoiseSVN\Languages`); jest to miejsce, gdzie trafiają pliki z pakietów językowych. Jednak, folder %APPDATA% nie wymaga uprawnień administratora, a przez to ma wyższy priorytet. Po uruchomieniu następnym razem TortoiseSVN, moduł sprawdzania pisowni będzie dostępny.

W przypadku instalowania wielu słowników, TortoiseSVN używa tych zasad, aby wybrać z którego z nich korzystać.

1. Sprawdźcie ustawienia `tsvn:projectlanguage`. Zapoznajcie się Sekcja 4.18, „Ustawienia projektu” dla informacji o ustawianiu właściwości projektu.

2. Jeśli język projektu nie jest ustawiony lub taki język nie jest zainstalowany, wypróbujcie język odpowiadający lokalizacji Windows.

3. Jeżeli właściwa lokalizacja systemu Windows nie działa, spróbujcie z językiem „bazowym”, np. `de_CH` (Szwajcaria-niemiecki) wraca do `de_DE` (niemiecki).

4. Jeśli wszystkie z powyższych czynności powiodły, wówczas domyślnym językiem jest angielski, który jest dołączony do standardowej instalacji.

Słownik

Adnotuj	Polecenie to jest wykonywane tylko dla plików tekstowych, adnotuje ono każdą linię, aby pokazać wersję repozytorium, w której ostatnio była zmieniona, autora, który dokonał tej zmiany. Nasza realizacji GUI nazywana jest TortoiseBlame i pokazuje także datę/czas zatwierdzenia i opisy zmian po najechaniu myszką na numer wersji.
Atrybut	Oprócz wersjonowania katalogów i plików, Subversion pozwala na dodanie wersjonowanych metadanych - dalej zwanych „atrybutami” do każdego z plików i katalogów pod kontrolą wersji. Każdy atrybut ma swoją nazwę i wartość, podobnie jak klucz rejestru. Subversion ma pewne specjalne atrybuty, których używa wewnętrznie, jak `svn:eol-style`. TortoiseSVN ma też podobne, jak `tsvn:logminsize`. Możecie dodać własne atrybuty z dowolną nazwą i wybraną wartością.
Atrybut wersji (revprop)	Podobnie jak pliki mogą mieć atrybuty, tak samo może każda wersja w repozytorium. Niektóre specjalne revprop są dodawane automatycznie, gdy wersja jest tworzona, a mianowicie: `svn:date svn:author svn:log`, które reprezentują odpowiednio datę/czas zatwierdzenia, autora i opis zmiany. Te atrybuty mogą być edytowane, ale nie są one wersjonowane, więc jakiekolwiek zmiany są trwałe i nie mogą być cofnięte.
Blokada	Kiedy zakłada się blokadę na wersjonowanym elemencie, zaznacza się go w repozytorium, jako niezatwierdzalny, z wyjątkiem kopii roboczej, gdzie blokada została nałożona.
Dodaj	Polecenie Subversion, które jest używane do dodawania pliku lub katalogu do kopii roboczej. Nowe pozycje są dodawane do repozytorium podczas zatwierdzenia.
Dziennik	Pokazuje historię wersji pliku lub folderu. Znany również jako „Historia”.
Eksport	To polecenie tworzy kopię wersjonowanego folderu, tak jak kopię roboczą, ale bez lokalnych folderów `.svn`.
FSFS	Własnościowy backend systemu plików Subversion dla repozytoriów. Może być stosowany na udziałach sieciowych. Domyślny dla repozytoriów w wersji 1.2 i nowszych.
Gałąź	Określenie często stosowane w systemach kontroli wersji, aby opisać to, co się dzieje, gdy rozwój rozdziela się w danym miejscu i podąża dwiema oddzielnymi ścieżkami. Można utworzyć odgałęzienie głównej linii rozwoju w celu opracowania nowej funkcji bez konieczności czynienia głównej linii niestabilną. Można też odgałęzić stabilną wersję, na której się wykonuje się tylko poprawki błędów, podczas gdy nowe zmiany rozwojowe odbywają się na niestabilnej linii głównej. W Subversion gałąź jest zaimplementowana jako „tania kopia”.
GPO	Obiekt zasad grupy.
Historia	Pokazuje historię wersji pliku lub folderu. Znany również jako „Dziennik”.
Import	Polecenie Subversion, do zaimportowania całej hierarchii folderów do repozytorium w pojedynczej wersji.
Konflikt	Kiedy zmiany z repozytorium zostają połączone z lokalnymi modyfikacjami, czasami obie zmiany występują na tych samych liniach. W takim przypadku

	Subversion nie może automatycznie wybrać wersji do wykorzystania, a plik zostaje oznaczony, że jest w stanie konfliktu. Musicie edytować plik ręcznie i rozwiązać konflikt, zanim będzie można zatwierdzić jakiekolwiek dalsze zmiany.
Kopia	W repozytorium Subversion możecie utworzyć kopię pojedynczego pliku lub całego drzewa. Są one realizowane przez „tanie kopie”, które działają trochę jak link do oryginału w tym, że nie zajmują prawie wcale miejsca. Wykonywanie kopii zachowuje historię elementu w kopii, więc możecie śledzić zmiany dokonane przed wykonaniem kopii.
Kopia robocza	To jest lokalna „piaskownica”, obszar, gdzie pracuje się na wersjonowanych plikach i zazwyczaj znajduje się na lokalnym dysku twardym. Aby utworzyć kopię roboczą, wykonuje się polecenie „Checkout” z repozytorium a przesyła swoje zmiany z powrotem do repozytorium używając „Commit”.
Pobierz	Polecenie Subversion, które tworzy lokalną kopię roboczą do pustego katalogu przez pobranie wersjonowanych plików z repozytorium.
Poprawka	Jeśli kopia robocza zawiera zmiany w plikach tekstowych, można użyć polecenia Subversion Diff do wygenerowania jednego pliku podsumowania tych zmian w formacie pliku różnicowego. Plik tego typu jest często określany jako „Poprawka”, i może być wysłany do kogoś innego (lub na listy mailingowe) i zastosowany w innej kopii roboczej. Ktoś bez dostępu do zatwierdzenia może dokonać zmian i przesłać plik poprawki do autoryzowanego opiekuna kodu by go zastosować. Lub, jeśli nie jest się pewnym co do zmiany, można zgłaszać poprawki dla zasięgnięcia opinii innych.
Porównaj	Skrót dla „Pokaż różnice”. Bardzo przydatny, gdy trzeba dokładnie sprawdzić, jakie zmiany zostały dokonane.
Przełącznik	Tak jak „Aktualizacja-do-wersji” zmienia okno czasowe kopii roboczej by pokazać inny punkt w historii, podczas gdy „Przełącz” zmienia okno przestrzenne kopii roboczej tak, aby wskazywała na inną część repozytorium. Jest to szczególnie przydatne podczas pracy z linią główną i gałęzią, które różnią się tylko kilkoma plikami. Możecie przełączać kopię roboczą między tymi dwoma i tylko zmienione pliki zostaną przeniesione.
Repozytorium	Repozytorium jest centralnym miejscem, gdzie dane są zapisywane i przechowywane. Repozytorium może być miejscem, gdzie wiele baz danych oraz plików znajduje się w celu rozpowszechniania w sieci, albo też repozytorium może być miejscem, które jest bezpośrednio dostępne dla użytkownika bez konieczności podróżowania po sieci.
Rozwiąż konflikt	Kiedy pliki w kopii roboczej są pozostawione w stanie konfliktu po scaleniu, konflikty muszą być uporządkowane przez ludzi za pomocą edytora (lub może TortoiseMerge). Proces ten jest określany jako „rozwiązywanie konfliktów”. Gdy ten się zakończy, można oznaczyć skonfliktowane pliki jako rozwiązane, co pozwala na ich zatwierdzenie.
Scalanie	Proces, w którym zmiany z repozytorium zostały dodane do kopii roboczej bez zakłócania zmian wprowadzonych na miejscu. Czasami tych zmian nie da się pogodzić automatycznie i o kopii roboczej mówi się, że jest w konflikcie. Scalenie odbywa się automatycznie podczas aktualizacji kopii roboczej. Można również scalić konkretne zmiany od innej gałęzi za pomocą polecenia TortoiseSVN Merge.
SVN	Często używany skrót dla Subversion.

	Nazwa niestandardowego protokołu Subversion używanego przez serwer repozytorium „svnserve”.
Uaktualnij	To polecenie Subversion ściąga najnowsze zmiany z repozytorium do kopii roboczej, scalając wszelkie modyfikacje dokonane przez innych z lokalnymi zmianami z kopii roboczej.
Uporządkuj	Cytat z księgi Subversion: „ Rekurencyjnie oczyść kopię roboczą, usuwając blokady i wznawiając niedokończone operacje. Jeśli kiedykolwiek otrzymasz błąd *kopia robocza zablokowana*, wykonaj następujące polecenie, aby usunąć nieaktualne blokady i uzyskać użyteczny stan kopii roboczej. ” Zauważ, że w tym kontekście *blokada* odnosi się do blokowania lokalnego systemu plików, nie do blokad repozytorium.
Usuń	Po usunięciu wersjonowanego elementu (i zatwierdzeniu zmiany) nie istnieje on już w repozytorium po zatwierdzonej wersji. Ale oczywiście nadal istnieje we wcześniejszych wersjach repozytorium, więc wciąż można mieć do niego dostęp. W razie potrzeby można skopiować element usunięty i „wskrzesić” go wraz z historią.
Wersja	Za każdym razem podczas zatwierdzania zestawu zmian, tworzona jest jedna nowa „wersja” w repozytorium. Każda wersja reprezentuje stan drzewa repozytorium w pewnym momencie swojej historii. Jeśli chcecie cofnąć się w czasie można sprawdzić repozytorium jak wyglądało ono w wersji N. W innym znaczeniu, wersja może odnosić się do zbioru wprowadzonych zmiany, gdy wersja ta została utworzona.
wersja BASE	Aktualna wersja bazowa pliku lub folderu w *kopii roboczej*. To jest wersja, w której plik lub folder znajdował się, kiedy ostatnie pobieranie, aktualizacja lub zatwierdzenie zostało uruchomione. Wersja BASE zwykle nie jest równa wersji HEAD.
wersja HEAD	Najnowsza wersja pliku lub folderu w *repozytorium*.
Wycofaj zmiany	Subversion przechowuje lokalną „pierwotną” kopię każdego pliku jak on wyglądał podczas ostatniej aktualizacji kopii roboczej. Jeśli dokonano zmiany i zapada decyzja, by je cofnąć, można użyć polecenia „revert”, aby wrócić do pierwotnej kopii.
Zatwierdź	To polecenie Subversion jest wykorzystywane do przekazywania zmian w lokalnej kopii roboczej z powrotem do repozytorium, tworząc nową wersję repozytorium.
Zmień lokalizację	Jeśli repozytorium zmienia położenie, może dlatego, że po przeniesieniu go do innego katalogu na serwerze, lub zmieniła się nazwa domeny serwera, trzeba wykonać „relocate” dla kopii roboczej, tak aby jego adresy URL wskazywały na nową lokalizację repozytorium. Uwaga: Należy używać tego polecenia tylko jeśli kopia robocza wskazuje na to samo miejsce w tym samym repozytorium, ale repozytorium zostało przeniesione. W innych okolicznościach prawdopodobnie wymagane jest polecenie „Switch”.

Indeks

P

R

S

T

U

V

W

Z