

TortoiseMerge

**Инструмент сравнения/
слияния для Windows**

Version 1.9

**Stefan Küng
Lübbe Onken
Simon Large**

TortoiseMerge: Инструмент сравнения/слияния для Windows: Version 1.9

Stefan Küng, Lübbe Onken, Simon Large

Перевод: Vladimir Serdyuk (vsrd@users.sourceforge.net), Станислав Петраков (stannic@gmail.com), Siarhei Niakhai (s.niakhai@gmail.com)

Дата публикации 2015/06/01 20:57:36 (r26544)

Содержание

Предисловие	vi
1. TortoiseMerge бесплатен!	vi
2. Благодарности	vi
1. Введение	1
1.1. Обзор	1
1.2. История TortoiseMerge	1
2. Основная концепция	3
2.1. Просмотр и слияние различий	3
2.2. Редактирование конфликтов	3
2.3. Применение заплаток	4
3. Использование TortoiseMerge	5
3.1. Режимы просмотра	5
3.1.1. Просмотр / слияние	5
3.1.2. Применение заплаток	7
3.2. Кнопки и другие элементы управления	8
3.3. Значки состояния строки	9
3.4. Слияние и редактирование конфликтов	9
3.5. Открытие файлов	10
3.5.1. Просмотр / слияние	11
3.5.2. Применение заплаток	11
3.6. Настройки	12
3.6.1. Страница основных настроек	12
3.6.2. Страница цветовых настроек	13
A. клавиатурные сокращения	15
A.1. Клавиатурные сокращения	15
B. Автоматизация TortoiseMerge	17
B.1. Параметры командной строки TortoiseMerge	17
Глоссарий	19

Список иллюстраций

1.1. Конфликт файлов	1
2.1. Конфликт файлов	4
3.1. Однопанельный вид	5
3.2. Двухпанельный вид	6
3.3. Трёхпанельный вид	7
3.4. Список файлов в заплатке	8
3.5. Диалог открытия файлов	11
3.6. Страница основных настроек	12
3.7. Страница цветовых настроек	13

Список таблиц

В.1. Список доступных параметров командной строки	17
---	----

Предисловие



Tortoise Merge

При работе над проектом, как над проектом с открытым исходным кодом, в котором несколько человек рецензируют ваш код, так и при работе над коммерческим проектом, в котором вы продаёте компоненты с их исходным кодом, найдутся люди и заказчики, обнаружившие ошибки или желающие улучшить ваш проект. Для этого они присылают заплатки, которые вам необходимо отрецензировать и принять решение об их применении.

TortoiseMerge поможет вам в обеих задачах:

- Рецензирование заплаток
- Применение заплаток

TortoiseMerge также окажет помощь в улаживании конфликтов в файлах, которые могут возникнуть, когда вы работаете с системой управления версиями и если кто-либо ещё работает с теми же файлами, что и вы.

1. TortoiseMerge бесплатен!

TortoiseMerge бесплатен. Вам не нужно за него платить, и вы можете применять его любым удобным вам способом. Он разработан под лицензией GPL.

TortoiseMerge - проект с открытым исходным кодом. Это означает, что Вы имеете полный доступ на чтение к исходному коду этой программы. Вы можете просмотреть его по ссылке <https://sourceforge.net/p/tortoisesvn/code/HEAD/tree/>. Самая последняя версия (над которой мы работаем в данный момент) находится в /trunk/, ранее выпущенные версии находятся в /tags/.

Вознаграждение

Хотя TortoiseSVN и TortoiseMerge и бесплатны, вы можете поспособствовать разработчикам, присылая заплатки и активно участвуя в разработке. Вы также можете помочь нам, подбодрив нас в то нескончаемое время, которое мы проводим за нашими компьютерами.

Работая над TortoiseSVN мы любим слушать музыку. И так как мы тратим много времени на проект, то нам надо *очень много* музыки. Поэтому мы составили несколько списков пожеланий с нашими любимыми музыкальными CD и DVD: <http://tortoisesvn.net/donate.html> Посмотрите, пожалуйста, также на список людей, которые участвуют в проекте отправляя патчи или переводы.

2. Благодарности

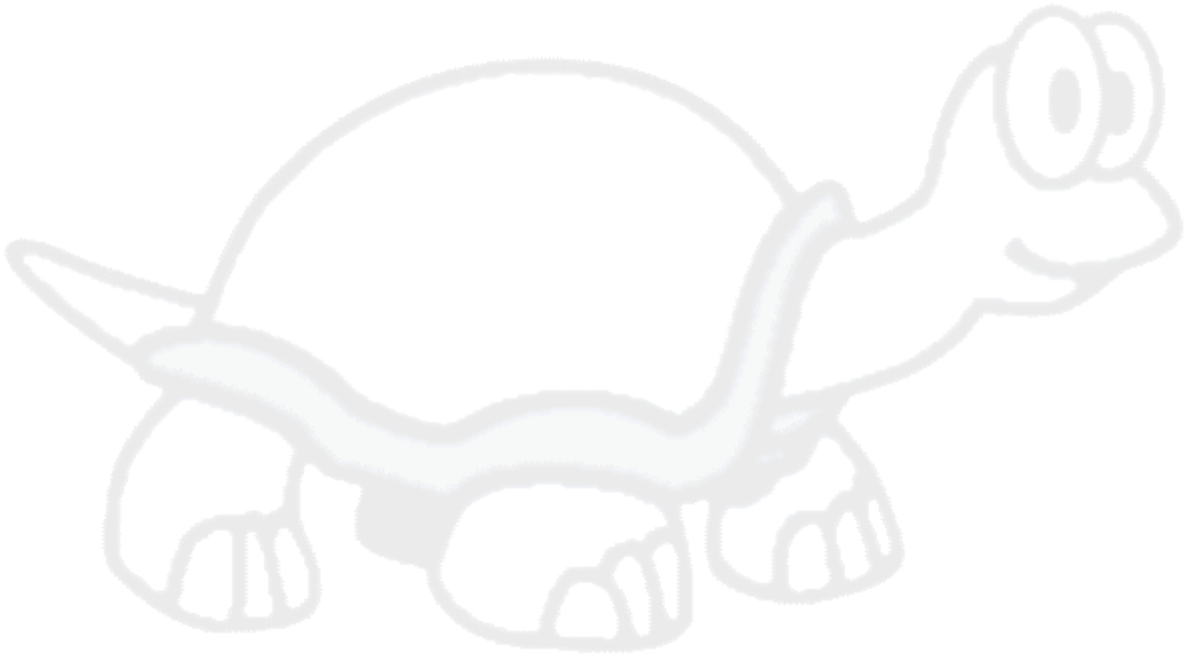
Стефану Кунгу (Stefan Küng)
за тяжёлый труд над TortoiseMerge

Люббе Онкену (Lübbe Onken)
за прекрасные значки, логотипы, отлов ошибок и ведение документации

Проекту Tigris Style (The Tigris Style project)
за некоторые стили, использованные в этой документации

Нашим помощникам
за исправления, сообщения об ошибках и новые идеи. И за помощь, оказанную другим в виде ответов в нашем списке рассылки

Нашим дарителям
за многие часы удовольствия от присланной нам музыки



Глава 1. Введение

1.1. Обзор

TortoiseMerge - это бесплатное приложение с открытым исходным кодом. Оно позволяет просматривать различия в текстовых файлах, производить слияние этих изменений, и даже рецензировать и применять объединённые файлы различий, часто называемые *заплатами*.

В процессе работы над текстовыми файлами, например файлами исходных кодов на вашем любимом языке программирования, или html/xml файлами для документации или веб-сайта, часто возникает ситуация, когда вам надо сравнить различные версии этих файлов. Иногда другую версию файла вы получаете от кого-то ещё, иногда вы просто желаете посмотреть сделанные вами изменения.

Если вы работаете с системой управления версиями (например, Subversion), тогда вы иногда получаете *конфликты* при обновлении вашей рабочей копии. Это происходит, если кто-либо другой произвёл изменения в тех же частях файла, над которыми работали и вы в это время. Тогда вы должны уладить эти конфликты вручную, - и в этой ситуации TortoiseMerge может оказать помощь.

Следующий рисунок показывает взаимоотношения между файлами, вовлечёнными в конфликт:

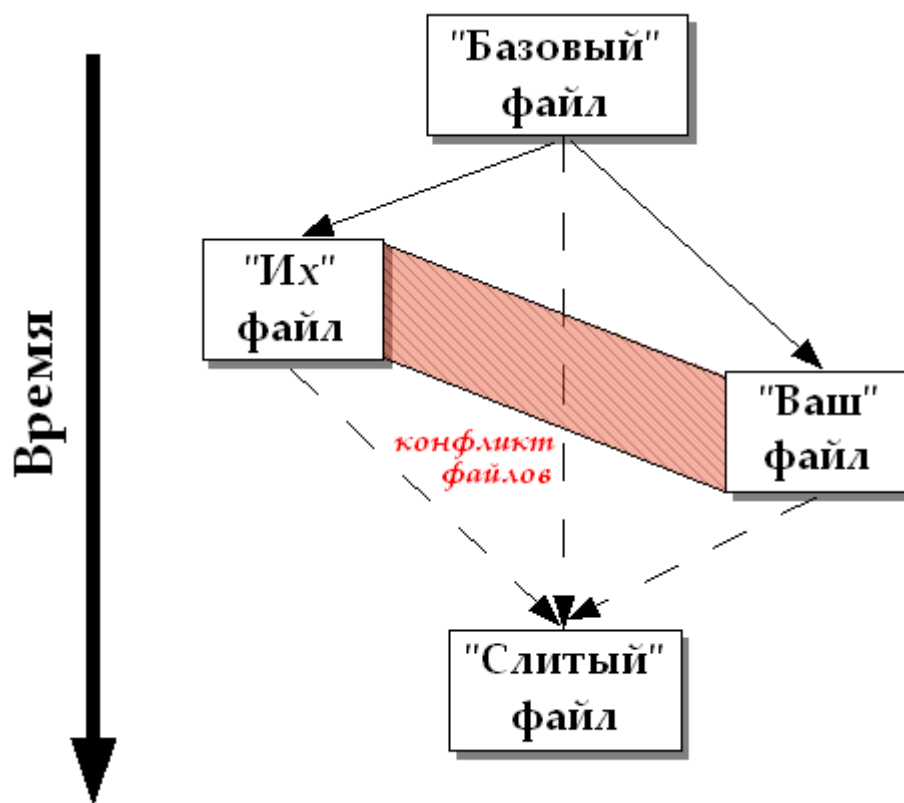


Рисунок 1.1. Конфликт файлов

1.2. История TortoiseMerge

В процессе работы над *TortoiseSVN* [<http://tortoisesvn.tigris.org>] мы обнаружили, что когда пользователи предоставляют заправки для улучшения проекта или же для исправления ошибок, надо приложить массу усилий только чтобы применить эти заправки. И много раз эти заправки были устаревшими и не могли быть применены при помощи инструмента командной строки для применения заплаток <http://>

unxutils.sourceforge.net/ поскольку мы продолжали работать над этими файлами и иногда эти файлы были уже изменены.

Именно в этом была основная причина, почему в TortoiseSVN изначально не была реализована команда TortoiseSVN → Создать заплатку: мы предпочитали, чтобы пользователи присылали нам файлы целиком, поскольку так заплатки было проще применять.

Итак, мы потратили много времени на поиски в Интернет инструмента с графическим интерфейсом, который мог бы применять файлы заплаток, но всё, что мы обнаружили - это то, что такого инструмента нет. Короче говоря: мы решили, что если никто не может сделать такой инструмент, то мы должны написать его сами. Так родился TortoiseMerge.

Поскольку TortoiseSVN уже использует библиотеку Subversion, которая обладает хорошим механизмом сравнения, было вполне естественно использовать эту библиотеку вместо GNU diff.

В Subversion 1.7 появилась команда *svn patch*, которая позволяет применить заплатку к рабочей копии. TortoiseMerge сейчас использует те же функции, что и Subversion во избежание разного поведения.

Глава 2. Основная концепция

У TortoiseMerge есть несколько различных применений:

- Просмотр различий между двумя файлами и внесение изменений путём слияния в один из них, или удаление нежелательных изменений.
- Редактирование конфликтов между локальными изменениями и изменениями из хранилища Subversion после обновления.
- Применение файлов заплаток и их пересмотр.

2.1. Просмотр и слияние различий

В этом режиме вы сравниваете два файла. Файл в левой панели (иногда обозначаемый как *их*) считается первоисточником, а в правой панели - изменённым файлом (он иногда обозначается как *мой*).

Вы можете делать простые изменения (на уровне строк) в файле из правой панели, включая:

- Возвращение изменённых строк обратно к тому виду, в котором они отображаются в левой панели.
- Использование совокупности обоих блоков, либо *их* перед *моим*, либо *мой* перед *их*.

Вы также можете редактировать файл в правой панели, как в текстовом редакторе. Отредактированные строки помечаются символом карандаша. Обратите внимание: если вы желаете произвести какие-либо из описанных выше изменений целых строк/блоков, лучше их сделать сначала, поскольку после того, как вы начнёте редактировать файл сами, TortoiseMerge будет больше не в состоянии отслеживать взаимосвязи с первоначальным файлом.

2.2. Редактирование конфликтов

Это также иногда называется трёхсторонним слиянием, и показывается в трёх панелях. Однако, на самом деле вовлечёнными оказываются четыре файла; файл, который не отображается - это общий базовый файл, последний общий предок тех двух файлов, которые конфликтуют. Взаимоотношения между тремя файлами объясняет нижеприведённый рисунок:

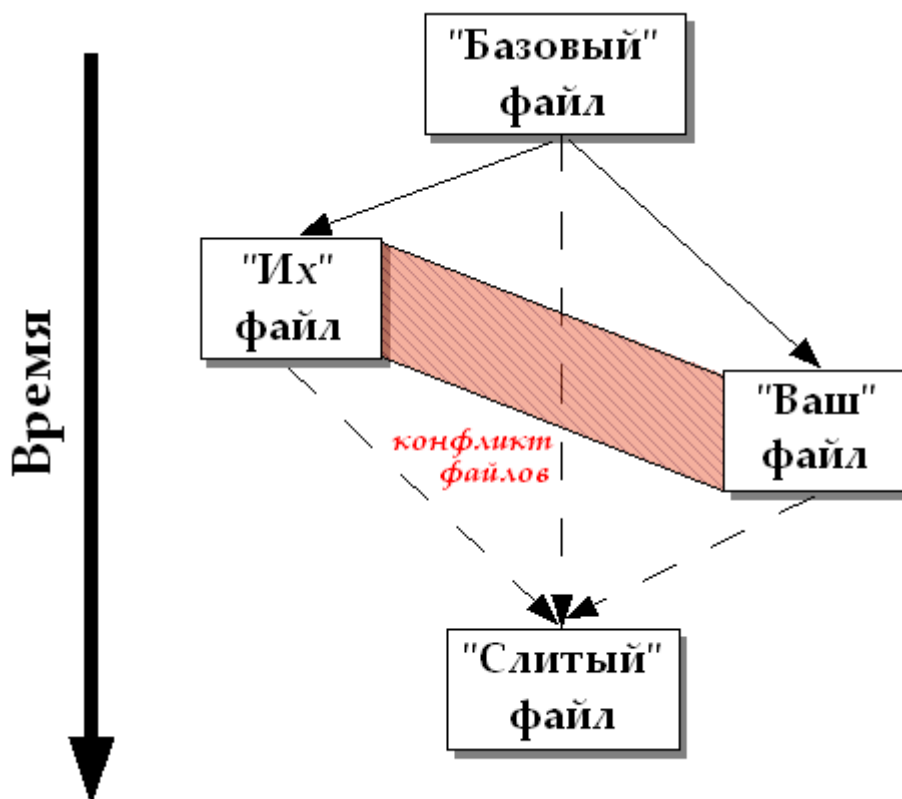


Рисунок 2.1. Конфликт файлов

Базовый файл представляет старейшую версию файла, в которую вы и они начали вносить изменения. Мой представляет базовый файл со всеми *вашими* изменениями, и их - это файл, который содержит все изменения, сделанные *другими*. В левой панели показываюются изменения *их* файла относительно базового, а в правой - изменения *моего* файла относительно базового. В нижней панели - результирующий файл, в котором вы и пытаетесь уладить конфликты.

В режиме улаживания конфликтов вы можете использовать блоки из *Моего*, *Их*, или обоих файлов. Но в этом случае изменения будут показаны в нижней панели.

2.3. Применение заплаток

Файл заплатки - это файл объединённых различий Subversion, содержащий информацию, необходимую для применения изменений к набору файлов. Заплатка может быть предоставлена другим разработчиком, чтобы вы могли посмотреть сделанные им изменения и, возможно, зафиксировать их в хранилище. Или, она может быть создана внутри TortoiseSVN при сравнении двух папок. Это случается при сравнении ревизии из хранилища с рабочей папкой, или при сравнении двух различных путей в хранилище.

В любом случае TortoiseMerge покажет небольшое окно, в котором перечислены входящие в заплатку файлы. По двойному щелчку на одном из этих файлов он будет подгружен и к нему будут применены изменения. В левой панели показывается содержимое исходного файла, а в правой - этот же файл после применения заплатки.

Вам доступны те же возможности редактирования файла в правой панели, что и в режиме сравнения/редактирования.

Глава 3. Использование TortoiseMerge

3.1. Режимы просмотра

В TortoiseMerge есть три основных режима просмотра: однопанельный, двухпанельный и трёхпанельный. Одно/двухпанельный режим используется для просмотра изменений, трёхпанельный - для улаживания конфликтов.

3.1.1. Просмотр / слияние

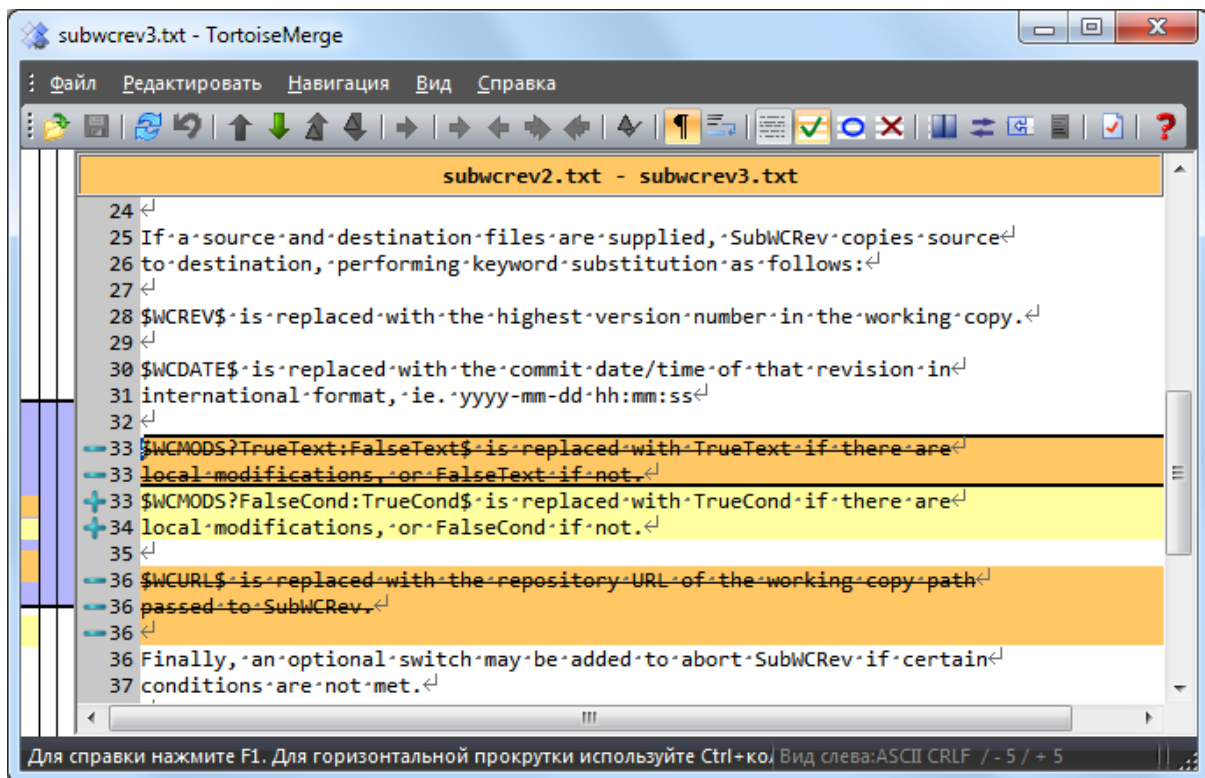


Рисунок 3.1. Однопанельный вид

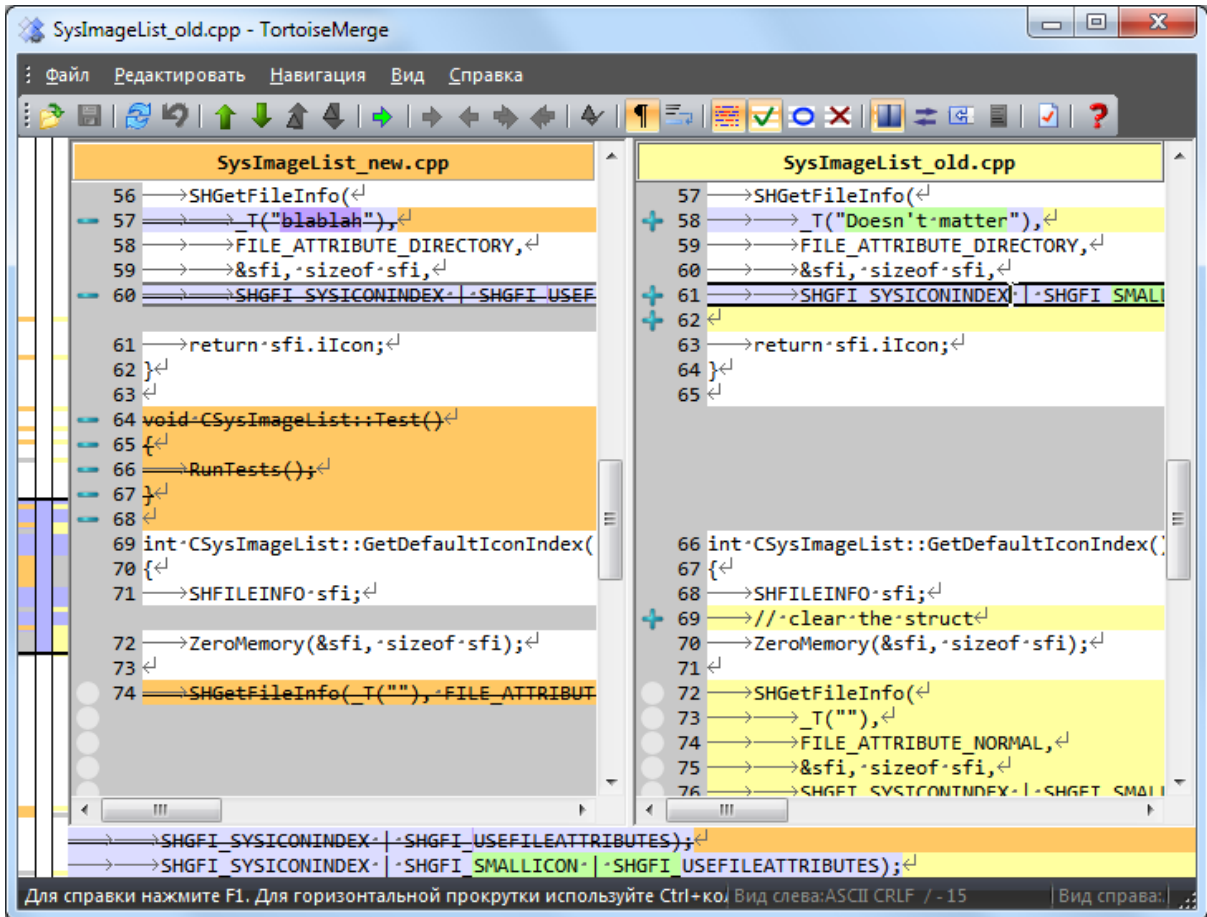


Рисунок 3.2. Двухпанельный вид

Двухпанельный вид имеет некоторые возможности, недоступные в однопанельном:

- Изменения внутри модифицированных строк показываются разными цветами. Добавленные части строк отображаются жёлтым цветом, удалённые части обозначаются при помощи тёмно-коричневой вертикальной линии в строке. Конечно же, вы можете настроить используемые здесь цвета. Посмотрите на рисунок выше, чтобы понять, как это работает.
- Реорганизация кода часто приводит к множеству изменений в непечатаемых знаках (пробелы, табуляции, новые строки) без реального изменения кода. Например, вы разбили очень длинную строку на несколько, или же объединили несколько строк в одну.

Такого рода изменения обозначаются белым кружком слева. И если вы видите такой кружок, вы уже знаете, что здесь нет реальных изменений кода и изучать этот блок более внимательно незачем.

- Ручное редактирование файла в правой панели возможно в двухпанельном виде. Такие изменения помечаются символом карандаша. Ручное редактирование файла в левой панели также возможно при нажатой кнопке Разрешить Редактирование и активной левой панели.

Если вы желаете сравнить/слить три файла, TortoiseMerge покажет различия в трёхпанельном виде. Этот вид также используется, когда вы улаживаете конфликты в файлах.

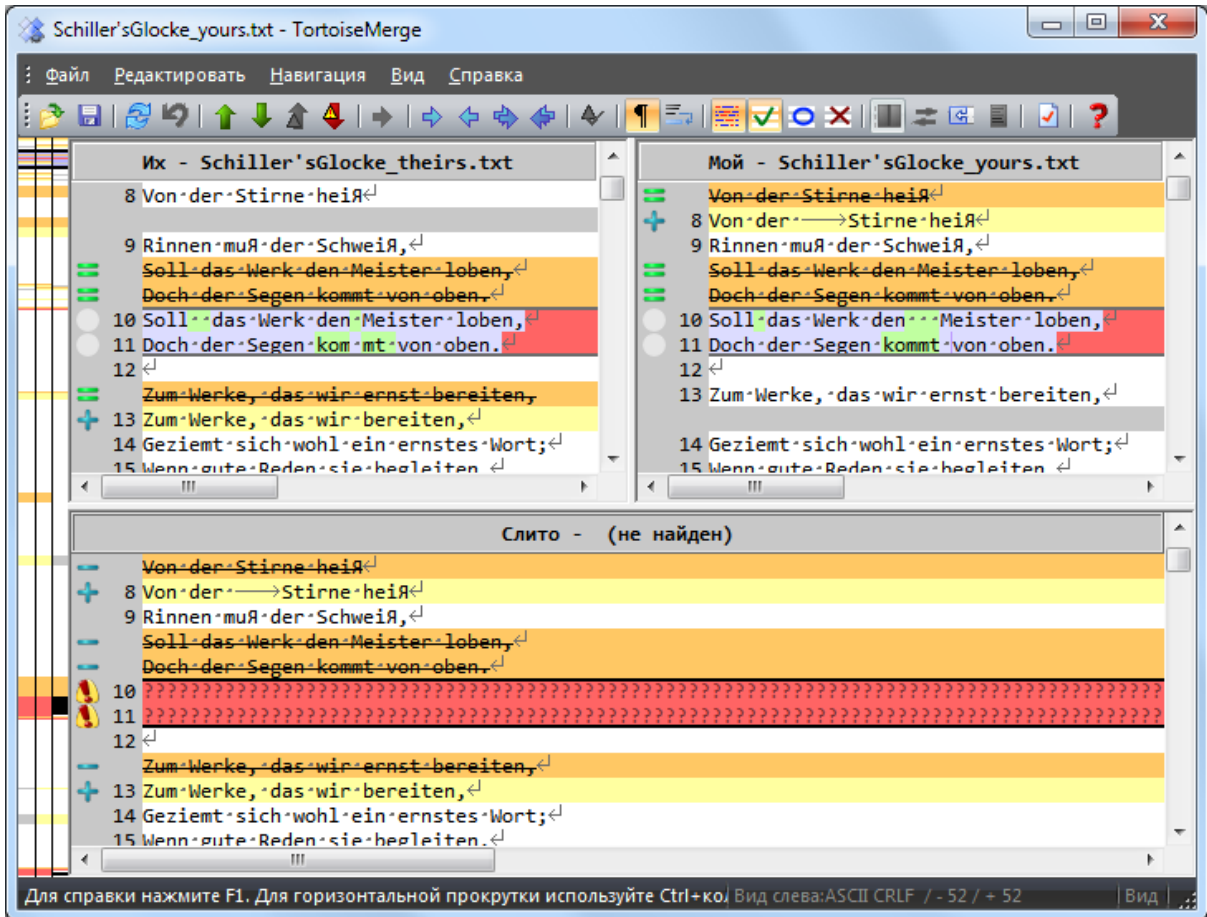


Рисунок 3.3. Трёхпанельный вид

Левая панель показывает различия между *Их* файлом и *Базовым*, а правая панель - между *Моим* файлом и *Базовым*. Нижняя панель показывает результат слияния *Базового* файла, *Моего* файла и *Их* файла с возможными конфликтами.

При наведении указателя мыши на заголовок панели появляется подсказка с именем используемого для сравнения файла, отображаемого в этой панели.

Ручное редактирование файла в левой или правой панели также возможно при нажатой кнопке **Разрешить Редактирование** в активной панели.

3.1.2. Применение заплаток

После того, как TortoiseMerge проанализирует файл заплатки, будет показано небольшое окно со списком всех файлов, для которых в заплатке есть изменения.

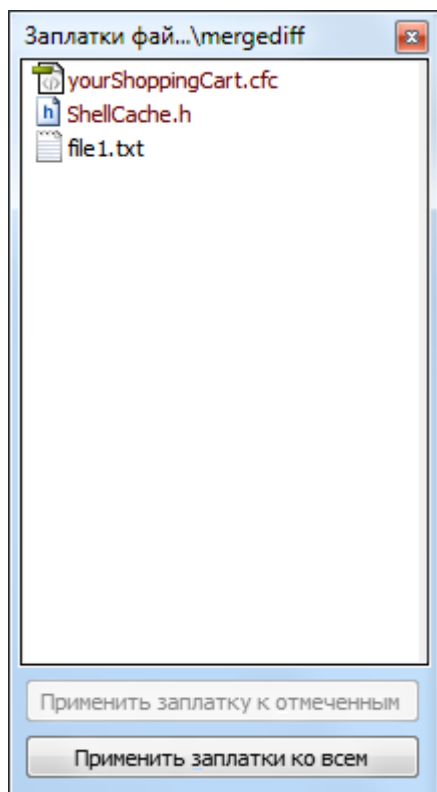


Рисунок 3.4. Список файлов в заплатке

Если имя файла показано чёрным, тогда заплатка может быть применена без проблем. Это означает, что файл не устарел по отношению к заплатке. Однако, если имя файла показано красным, тогда заплатка не может быть применена напрямую, так как вы уже изменили этот файл.

Окно файла заплатки имеет контекстное меню, позволяющее предварительно посмотреть на результат применения заплатки к текущему файлу (применить без сохранения), применить и сохранить изменения для выбранного файла, или применить и сохранить изменения для *всех* файлов в списке. Двойной щелчок запускает предварительный просмотр.

3.2. Кнопки и другие элементы управления

В зависимости от настроек вы видите панель инструментов или ленту со множеством кнопок. При наведении указателя мыши на кнопку показывается всплывающая подсказка поясняющая функцию кнопки.

Слева находится полоса местоположения. Она обеспечивает быстрое наглядное определение того, где находятся изменения в файле. Полоса состоит из трёх столбцов. Левый столбец соответствует левой панели, правый столбец - правой, центральный - нижней (если присутствует). В однопанельном виде используется только левый столбец. Полоса местоположения также может использоваться как полоса прокрутки для одновременной прокрутки всех окон.

Если вы сделаете двойной щелчок на слове, то после этого каждое упоминание этого слова будет подсвечено во всем документе, в главной панели и в полосе навигации. Повторный двойной щелчок на слове удаляет подсветку.

Если вы щёлкнете на левой границе, или трижды щёлкнете на строке то будет выбрана вся строка.

Внизу окна расположена строка состояния. В ней отображается количество добавленных и удаленных строк в Их и Моих и количество оставшихся неулаженных конфликтов.

Строка статуса также содержит элементы управления с выпадающим меню, которые показывают как обрабатываются файлы:

Кодировка

Кодировка определяет как символы в панели загружаются/сохраняются и отображаются. Наиболее распространенная кодировка на английском это ASCII (что обозначает локальную кодировку языка операционной системы), но вы можете изменить её на UTF8, UTF16LE, UTF16BE, UTF32LE и UTF32BE, обе с или без метки порядка байтов (BOM).

Концы строк

Наиболее распространённый символ окончания строки в Windows это CRLF, но вы можете изменить его на тот, который вам нравится. Обратите внимание, что если вы измените окончания строк, то *все* окончания строк во всём файле изменятся, даже если при загрузке окончания строк были не одинаковы.

Размер табул.

Опция над меню поля со списком показывает символы табуляции или пробелы вставляются при нажатии клавиши табуляции. Если включена опция *символ "умной" табуляции*, используется алгоритм для определения, что подойдёт лучше.

Размер табуляции определяет сколько символов пробела вставляется при редактировании и нажатии символа табуляции, или на сколько символов сдвигается следующее слово, когда встречается символ табуляции.

3.3. Значки состояния строки

Изменённые строки помечаются различными значками для обозначения типа произведённого изменения.



Строка была добавлена.



Строка была удалена.



Изменение было отменено путём возвращения первоначального содержимого строки.



Эта строка содержит изменения только пробельных символов. Если помечено несколько последовательных строк, то это может означать, что абзац был переразбит на строки заново и что некоторые слова могли переместиться на соседние строки.



Строка была отредактирована вручную с использованием TortoiseMerge в качестве текстового редактора.



Эта строка находится в конфликте.



В этой строке есть конфликт, который не оказывает влияния на результат из-за настроек обработки пробельных символов или завершений строк.



Эта строка обнаружена как перемещённая из / в другое местоположение.

3.4. Слияние и редактирование конфликтов

TortoiseMerge не только *показывает* вам различия между файлами, но и позволяет вам улаживать конфликты или применять изменения.

Если вы в режиме просмотра с двумя панелями, то вы можете только редактировать файл на правой панели (Мой). Для применения изменений сделанных в левом файле (Их), сделайте правый щелчок на измененных строках и выберите Context Menu → Использовать 'их' текстовый блок. После этого изменения из левого файла добавляются в правый файл.

Иногда вам на самом деле необходимы оба текстовых блока, и контекстное меню предлагает Контекстное меню → Использовать оба текстовых блока (этот в начале) и Контекстное меню → Использовать оба текстовых блока (этот в конце).

Вы можете редактировать получившийся файл точно также, как и в текстовом редакторе. Отредактированные строки помечаются символом карандаша. Обратите внимание: если вы желаете произвести какие-либо из описанных выше изменений целых строк/блоков, лучше их сделать сначала, поскольку после того, как вы начнёте редактировать файл сами, TortoiseMerge будет больше не в состоянии отслеживать взаимосвязи с первоначальным файлом.

Если вы в режиме просмотра с тремя панелями (иногда называемый *вид слияния*), то вы можете редактировать только файл в нижней части (Слит0). Как в двухпанельном режиме, вы можете сделать правый щелчок на конфликтующих строках и выбрать Context Menu → Использовать 'их' текстовый блок или Context Menu → Использовать 'мой' текстовый блок. В дополнение, если хотите *оба* блока, то можете выбрать Context Menu → Использовать 'мой' текстовый блок перед 'их' или Context Menu → Использовать 'их' текстовый блок перед 'моим'. В соответствии с выбранными вами командами изменения будут в результирующем Слит0 файле.

Иногда может быть так, что в Subversion файл помечен как конфликтующий, но при просмотре его при помощи TortoiseMerge никаких конфликтов не показывается. Это может быть из-за выбранного метода обработки пробельных символов: если вы выбрали игнорирование изменений завершений строк или пробельных символов, то строки будут помечены значком *Конфликт-игнорируемое*. Для улаживания конфликта вам всё равно придётся выбрать, какой из вариантов вы желаете использовать.



Важно

Обратите внимание, при повторном использовании в TortoiseMerge тех же файлов любые изменения в вашей рабочей копии, сделанные TortoiseMerge или вручную, *будут потеряны* и файл окажется таким, каким он был при начале улаживания конфликта.

3.5. Открытие файлов

При запуске TortoiseMerge без указания параметров в командной строке вам надо будет открыть файлы вручную, используя Файл → Открыть.

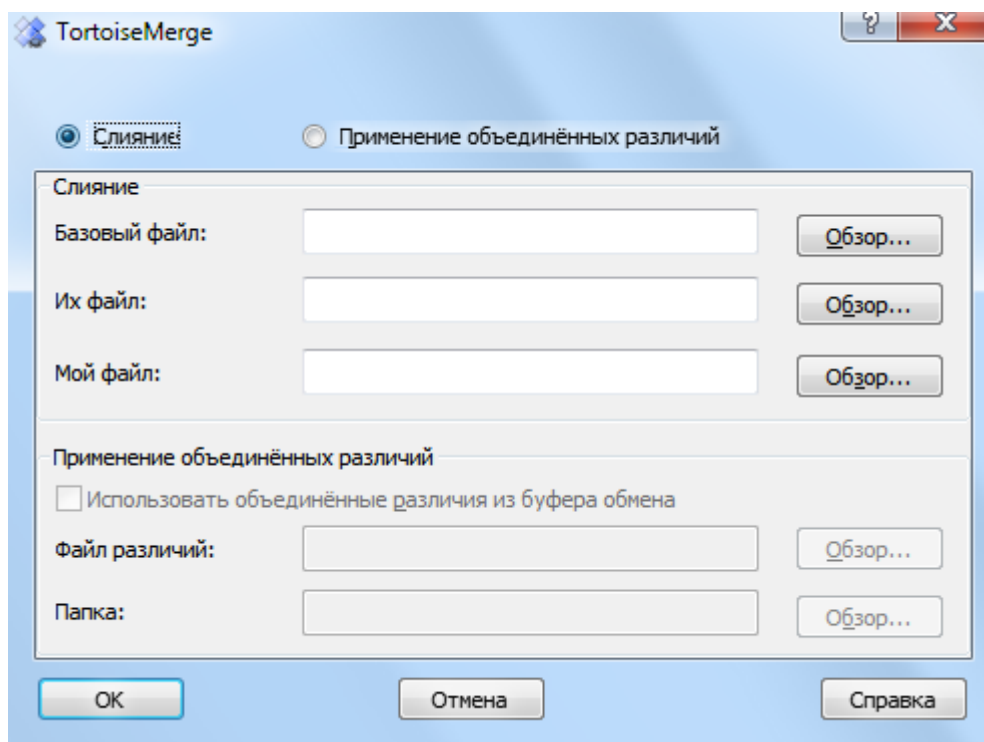


Рисунок 3.5. Диалог открытия файлов

Первое, что вы должны сделать - решить, желаете ли вы только сравнить/слить файлы, или же вы хотите применить заплатку. В зависимости от того, что вы выберете, станут доступны соответствующие поля ввода и кнопки обзора.

3.5.1. Просмотр / слияние

Если вы желаете сравнить/слить файлы, вам надо задать по крайней мере два из трёх возможных путей к *базовому*, *моему* и *их* файлам. Если вы выберете только два файла, то TortoiseMerge покажет различия между этими двумя файлами или в двухпанельном, или в однопанельном.

Если вы желаете слить три файла, TortoiseMerge покажет различия в трёхпанельном виде. Этот вид обычно используется, когда вы улаживаете конфликты в файлах. Выходной файл в этом случае не имеет названия и вам необходимо использовать **Файл** → **Сохранить как...** для сохранения результата.

3.5.2. Применение заплаток

Если вы желаете применить заплатку, вы должны задать путь к файлу заплатки, а также путь к папке, к которой этот файл заплатки должен быть применён.

3.6. Настройки

3.6.1. Страница основных настроек

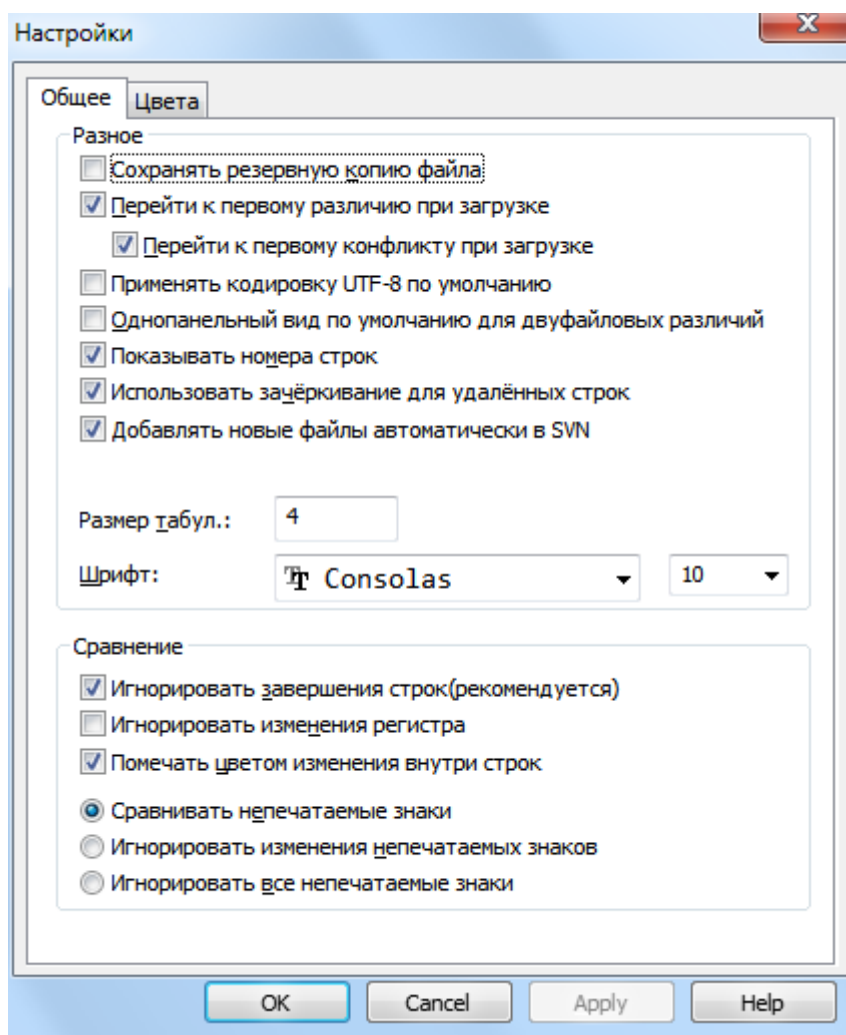


Рисунок 3.6. Страница основных настроек

Большинство настроек не требуют объяснений, но некоторые моменты необходимо прояснить.

Сохранять резервную копию файла переименовывает исходный файл из рабочей копии в `filename.bak` перед сохранением изменений.

Применять кодировку UTF-8 по умолчанию если установлено, то файлы ANSI загружаются в кодировке UTF-8 и в ней же сохраняются после редактирования.

Максимальная длина строки для внутрискрочных различий TortoiseMerge может замедляться при отображении внутрискрочных различий для очень длинных строк. Поэтому только для строк короче 3000 символов показаны внутрискрочные различия. Вы можете здесь изменить это значение.

Игнорировать завершения строк скрывает изменения, появившиеся только из-за различий в типах перевода строки.

Игнорировать изменения регистра скрывает изменения, появившиеся только из-за различий в регистре символов. Это может пригодиться для таких приложений, как Visual Basic, который изменяет регистр в именах переменных без предупреждения.

3.6.2. Страница цветовых настроек

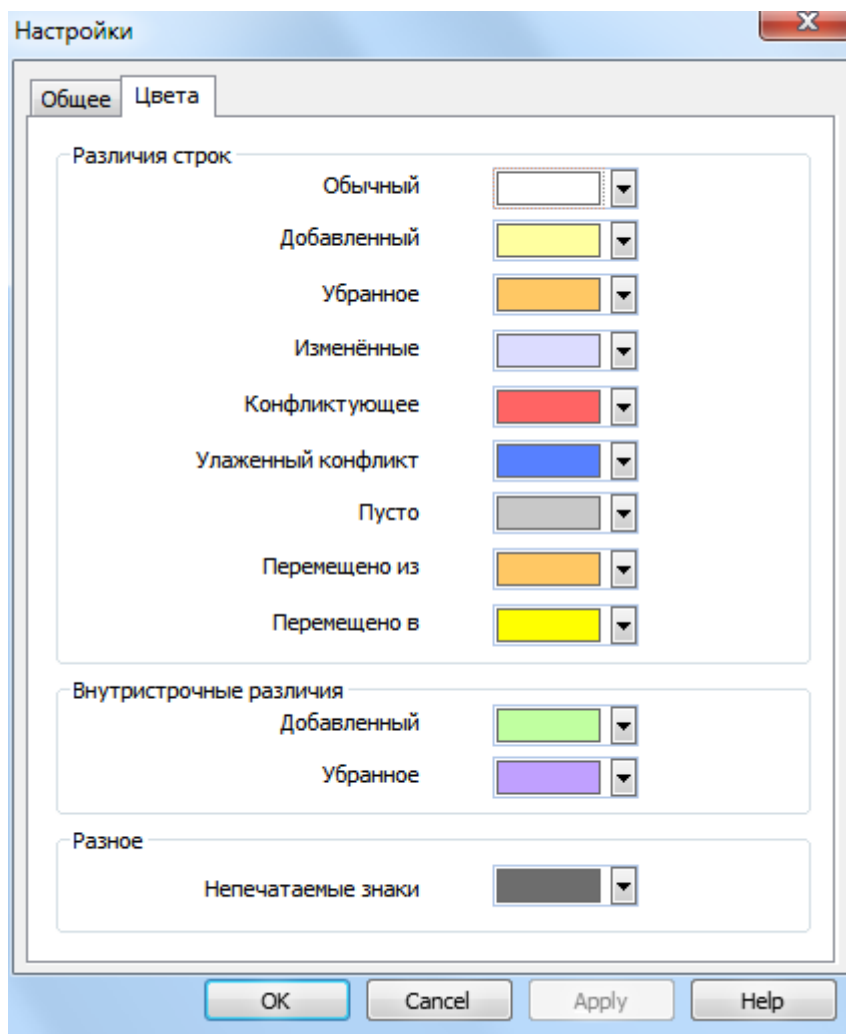


Рисунок 3.7. Страница цветовых настроек

Этот диалог позволяет вам выбрать разные фоновые цвета для выделения различных изменений строк.

Нормальное

Все неизменённые строки, или строки, где изменения скрыты.

Добавлено

Строки, которые были добавлены.

Убранное

Строки, которые были удалены.

Изменено

Строки с относительно малым количеством изменений, показываемые с внутристрочными различиями. Этот цвет используется для *неизменённой* части содержимого строки. Изменённые части показываются цветами для добавленного и удалённого текста внутри строк, описанными ниже. Если не включена опция **Помечать цветом изменения внутри строк**, этот цвет не будет использоваться и изменённые строки всегда будут показываться как заменённые (старая удалена, новая добавлена).

Конфликтующее

Строки, которые были изменены в обоих файлах.

Улаженный конфликт

Строки, изменённые в обоих файлах, для которых уже выбрано, какая из версий будет использована.

Отсутствующее

Строки, которые были добавлены в противоположной панели и которые отсутствуют в текущей.

Текст, добавленный внутри строки

При показе внутрискочных различий этим цветом отображается добавленный текст.

Текст, удалённый внутри строки

При показе внутрискочных различий этим цветом отображается удалённый текст.

Непечатаемые знаки

Символы, используемые для отображения непечатаемых знаков, показываются в цвете, отличном от цвета основного текста.

Приложение А. клавиатурные сокращения

Список клавиатурных сокращений и команд.

А.1. Клавиатурные сокращения

Ctrl-Q, Ctrl-W, Escape

Выйти из программы

Ctrl-C

Скопировать выбранный текст в буфер обмена

Ctrl-X, Shift-Del

Вырезать выбранный текст в буфер обмена

Ctrl-V, Shift-Insert

Вставить выбранный текст из буфера обмена

Ctrl-Z, Alt-Backspace

Отменить последние изменения

Ctrl-F

Открыть диалог для текстового поиска и замены

Ctrl-O

Открыть файлы для сравнения/слияния

Ctrl-S

Сохранить изменения

Ctrl-Shift-S

Сохранить как...

F7

Перейти к следующему различию

Shift-F7

Перейти к предыдущему различию

F8

Перейти к следующему конфликту

Shift-F8

Перейти к предыдущему конфликту

Ctrl-D

Переключиться между одно- и двухпанельным видами

Ctrl-R

Перечитать файлы и отменить все сделанные изменения

Ctrl-T

Переключить режим отображения непечатаемых знаков

Ctrl-L

Переключает между сворачиванием неизменных секций или нет

Ctrl-P

Переключить между способами переноса строк

Ctrl-G

Перейти к строке

Ctrl-A

Выбрать весь текст

Ctrl-P

Переключить между способами переноса строк

Ctrl-U

Переключает виды

Ctrl-колёсико мыши

Прокрутка экрана влево/вправо

Ctrl-Tab

Переключить между левым/правым/нижним видом

Приложение В. Автоматизация TortoiseMerge

TortoiseMerge можно запустить с параметрами командной строки, и тогда не надо будет указывать файлы в диалоге Открыть. Это также может пригодиться, если вам надо вызвать TortoiseMerge из другого приложения.

В.1. Параметры командной строки TortoiseMerge

Большинство параметров требуют дополнительной информации, вроде пути или какой-то другой строки. В этом случае, добавьте к параметру ':' и после этого укажите строку/путь. Пример:

```
/base:"c:\folder\my base file.txt"
```

Команда	Описание
/?	Показывает диалоговое окно с наиболее важными параметрами командной строки.
/help	То же самое, что и ?.
/base	Задаёт <i>базовый</i> файл, используемый в трёхстороннем сравнении. Это общий предок сравниваемых файлов, хотя он и не показывается в отдельном окне. При двухстороннем сравнении это левый файл.
/basename	Имя 'базового' файла, которое показывается в заголовке панели вместо пути к файлу. При трёхстороннем сравнении оно показывается во всплывающей подсказке заголовка панели.
/basereflectedname	Имя используемое в шаблонах editorconfig.
/theirs	Задаёт <i>их</i> файл, используемый в трёхстороннем сравнении; показывается в левой панели.
/theirsname	Имя 'их' файла, которое показывается в заголовке панели вместо пути к файлу.
/theirsreflectedname	Имя используемое в шаблонах editorconfig.
/mine	Задаёт <i>мой</i> файл, используемый в трёхстороннем сравнении; показывается в правой панели. При двухстороннем сравнении это правый файл.
/minename	Имя 'моего' файла, которое показывается в заголовке панели вместо пути к файлу.
/minereflectedname	Имя используемое в шаблонах editorconfig.
/merged	Определяет результирующий <i>объединённый</i> файл используемый в различиях. Это путь файла, в котором сохраняются результаты слияния или разрешения конфликтов. Если это не задано для трёхстороннего сравнения, то TortoiseMerge спросит у пользователя куда сохранить результат. Если это не задано для двухстороннего сравнения, то TortoiseMerge автоматически использует путь файла, отображаемого в правой панели, как путь сохранения.
/mergedname	Имя 'слитого' файла, которое показывается в заголовке панели вместо пути к файлу.
/mergedreflectedname	Имя используемое в шаблонах editorconfig.

Команда	Описание
/patchpath	Путь, к которому необходимо применить заплатку. Если путь не указан, TortoiseMerge попытается найти этот путь самостоятельно по соответствию путям в заплатке, но это может занять <i>очень продолжительное</i> время.
/saverequired	Если задано, то TortoiseMerge выдаст запрос на сохранение перед выходом даже если пользователь не изменял файлы.
/saverequiredonconflicts	Если задано, то TortoiseMerge выдаст запрос на сохранение перед выходом если найдены конфликты, даже если пользователь не изменял файлы.
/patchoriginal	Имя исходного файла для применения заплатки. Используется в заголовке панели.
/patchpatched	Имя результирующего файла с применённой заплаткой. Используется в заголовке панели.
/diff	Путь к файлу заплатки/различий, который будет применяться к папке.
/oneway	Заставляет TortoiseMerge запускаться в однопанельном режиме вместо режима, указанного пользователем в настройках.
/reversedpatch	Меняет местами левую и правую панели с указанными для сравнения двумя файлами.
/createunifieddiff	Создаёт файл объединённых различий (файл заплатки) для двух файлов, указанных /origfile:"путь_к_исходному_файлу" и /modifiedfile:"путь_к_изменённому_файлу". Путь к создаваемому файлу указывается в /outfile:"путь_к_создаваемому_файлу_заплатки". Если параметр /outfile не задан, то будет показан диалог сохранения файла, чтобы пользователь мог выбрать, куда сохранить файл заплатки. Обратите внимание: если указан параметр /createunifieddiff, то все остальные параметры игнорируются.
/line	Задаёт номер строки, к которой будет осуществлён переход после загрузки файлов.
/readonly	Предотвращает файл от редактирования. Это означает, что возможности редактирования TortoiseMerge отключены.

Таблица В.1. Список доступных параметров командной строки

Можно также указать в командной строке просто имена файлов для совместимости с другими программами сравнения. Формат командной строки для этой упрощённой формы выглядит следующим образом:

```
TortoiseMerge BaseFilePath MyFilePath [ TheirFilePath ]
```

Если указаны два файла, они сравниваются друг с другом. Если указано три файла, то первый используется как базовый, а остальные два сравниваются с ним трёхсторонним сравнением.

Глоссарий

FSFS	Собственная внутренняя файловая система Subversion для хранилищ. Может быть использована на сетевых разделяемых ресурсах. Используется по умолчанию для хранилищ, начиная с версии 1.2.
GPO	Объект групповой политики
SVN	Часто используемое сокращение для Subversion. Имя собственного протокола Subversion, используемого сервером хранилища «svnserve».
Авторство (Blame)	Эта команда предназначена только для текстовых файлов. Она снабжает каждую строку информацией о том, в какой ревизии хранилища строка была последний раз изменена, и об авторе, выполнившим это изменение. Наша реализация с графическим интерфейсом пользователя называется TortoiseBlame и показывает также дату/время фиксации и сообщение журнала при наведении указателя мыши на номер ревизии.
Базовая ревизия (BASE)	Текущая базовая ревизия файла или папки в вашей <i>рабочей копии</i> . Эта ревизия, в которой файл или папка были при последнем извлечении, обновлении или фиксации. Базовая ревизия обычно не равна ведущей (HEAD) ревизии.
Блокировка	Когда вы выполняете блокировку версированного элемента, вы помечаете его в хранилище как недоступный для фиксации никому, кроме той рабочей копии, которая произвела блокировку.
Ведущая ревизия (HEAD)	Последняя ревизия файла или папки в <i>хранилище</i> .
Добавить	Команда Subversion, которая используется для добавления файла или папки в вашу рабочую копию. Эти новые элементы будут добавлены в хранилище при фиксации.
Журнал	Показывает историю ревизий файла или папки. Также известен как «История».
Заплата (Patch)	Если рабочая копия содержит изменения только в текстовых файлах, можно воспользоваться командой 'Различия' Subversion для генерации одного сводного файла, содержащего все эти изменения в формате объединённых различий. Файлы этого типа часто называются «Заплатами», и они могут быть отправлены по электронной почте кому-либо ещё (или в список рассылки) и применены к другой рабочей копии. Кто-нибудь без возможности фиксировать изменения может подготовить изменения и отправить файл заправки для применения лицу, обладающему правом фиксации. Или, если нет уверенности в изменении, можно отправить заправку другим для рецензирования.
Извлечь	Команда Subversion, создающая локальную рабочую копию в пустой папке путём загрузки версированных файлов из хранилища.
Импорт	Команда Subversion для импорта в хранилище целой иерархии папок за одну ревизию.
История	Показывает историю ревизий файла или папки. Также известна как «Журнал».
Конфликт	Когда изменения из хранилища сливаются с локальными, иногда случается, что они затрагивают одни и те же строки. В этом случае Subversion не может самостоятельно решить, какую из версий

	<p>использовать и файл считается находящимся в конфликтном состоянии. Вы должны вручную отредактировать файл и уладить конфликты перед тем, как вы сможете зафиксировать любые дальнейшие изменения.</p>
Копирование	<p>В хранилище Subversion вы можете создать копию одного файла или целого дерева. Это реализовано через «лёгкие копии», которые ведут себя подобно ссылке на оригинал: они почти не занимают места. Создание копии сохраняет историю элемента в копии, так что вы можете отследить изменения, сделанные до создания копии.</p>
Обновить	<p>Эта команда Subversion вносит последние изменения из хранилища в вашу рабочую копию, сливая все сделанные другими изменения с локальными изменениями из рабочей копии.</p>
Ответвление	<p>Термин, часто используемый в системах управления версиями для описания ситуации, когда разработка разветвляется в определённой точке и следует по двум различным путям. Можно создать ответвление из основной линии разработки - для реализации новой возможности без приведения главной линии в нестабильное состояние, или можно ответить стабильную версию, в которой делаются только исправления ошибок, в то время как новые разработки ведутся в нестабильном стволе. В Subversion ответвления реализованы как «лёгкие копии».</p>
Очистка	<p>Цитата из Книги о Subversion: «Рекурсивно очищает рабочую копию, удаляя блокировки и возобновляя незаконченные действия. Если вы когда-либо столкнётесь с ошибкой <i>рабочая копия заблокирована</i>, то запустите эту команду для удаления подвисших блокировок и приведения вашей рабочей копии в работоспособное состояние.» Заметьте, что в этом контексте <i>блокировка</i> относится к блокированию в локальной файловой системе, а не в хранилище.</p>
Параметр	<p>Как «обновить-до-ревизии» изменяет временное окно, через которое смотрит рабочая копия, на другую точку в истории, так и «переключить» изменяет пространственное окно рабочей копии так, чтобы оно указывало на другую часть хранилища. Это особенно полезно при работе в стволе и ответвлениях, отличающихся всего несколькими файлами. Вы можете переключать вашу рабочую копию между двумя местами, и передаваться будут только изменённые файлы.</p>
Перебазировать	<p>Когда ваше хранилище перемещается, возможно из-за переноса в другую папку на сервере, или изменяется доменное имя сервера, вам необходимо «перебазировать» вашу рабочую копию, чтобы содержащийся в ней URL хранилища указывал на новое местоположение.</p> <p>Обратите внимание: вы должны использовать эту команду только в случае, если ваша рабочая копия относится к тому же местоположению в том же хранилище, но само хранилище было перемещено. В любом другом случае вам, вероятно, вместо этой команды необходимо использовать команду «Переключить».</p>
Рабочая копия	<p>Это ваша локальная «песочница», область, где вы работаете с версионными файлами, и она обычно находится на вашем локальном жёстком диске. Вы создаёте рабочую копию путём «извлечения» из хранилища, и возвращаете изменения назад в хранилище при помощи «фиксации».</p>
Различие	<p>Обозначение для «Показать различия». Очень полезно, когда вы желаете увидеть, какие именно изменения были сделаны.</p>
Ревизия	<p>Каждый раз при фиксации набора изменений создаётся новая «ревизия» в хранилище. Каждая ревизия представляет состояние дерева хранилища</p>

в определённой точке его истории. При желании, вы можете вернуться назад во времени и изучить хранилище, каким оно было в ревизии N.

Иначе говоря, на ревизию можно ссылаться как на набор изменений, которые были сделаны при создании этой ревизии.

Свойство	Помимо версирования ваших папок и файлов, Subversion позволяет добавлять версированные метаданные, - называемые «свойствами», - к каждому версированному файлу или папке. У каждого свойства есть имя и значение, почти как у ключа реестра. В Subversion есть несколько специальных свойств для внутреннего использования, таких как <code>svn:eol-style</code> . И у TortoiseSVN тоже есть такого рода свойства, вроде <code>tsvn:logminsize</code> . Вы можете добавить ваши собственные свойства с именем и значением по вашему выбору.
Свойство ревизии (revprop)	Так же как и файлы, каждая ревизия в хранилище может обладать свойствами. Некоторые специальные свойства добавляются автоматически при создании ревизии, а именно <code>svn:date</code> <code>svn:author</code> <code>svn:log</code> , представляющие дату/время фиксации, того, кто её произвёл и сообщение журнала, соответственно. Эти свойства могут быть отредактированы, но они не версируются, поэтому любое изменение является постоянным и не может быть отменено.
Слияние	Процесс, при помощи которого изменения из хранилища добавляются в вашу рабочую копию без разрушения уже сделанных в ней изменений. Иногда изменения не могут быть приведены в соответствие автоматически, и рабочая копия считается находящейся в состоянии конфликта. Слияние случается автоматически при обновлении рабочей копии. Вы можете слить отдельные изменения из другого ответвления при помощи команды слияния TortoiseSVN.
Убрать изменения	Subversion локально сохраняет «нетронутую» копию каждого файла, каким этот файл был при последнем обновлении рабочей копии. Если вы сделали какие-то изменения и решили их откатить, вы можете использовать команду «Убрать изменения» для того, чтобы вернуться к нетронутой копии.
Удалить	Когда вы удаляете версированный элемент (и фиксируете это изменение), элемент больше не существует в хранилище после зафиксированной ревизии. Но, конечно, он всё ещё существует в более ранних ревизиях хранилища, так что вы всё ещё можете получить к нему доступ. При необходимости, вы можете скопировать удалённый элемент и «воскресить» его вместе с историей.
Уладить	В случае, когда файлы в рабочей копии находятся в состоянии конфликта после слияния, эти конфликты должны быть разрешены человеком при помощи редактора (или, возможно, TortoiseMerge). Этот процесс называется «улаживанием конфликтов». После его выполнения вы можете отметить конфликтующие файлы как улаженные, что позволит их зафиксировать.
Фиксировать	Эта команда Subversion используется для передачи изменений из вашей рабочей копии обратно в хранилище, при этом создавая в нём новую ревизию.
Хранилище	Хранилище - центральное место, где данные хранятся и обслуживаются. Хранилище может быть местом, в котором размещаются несколько баз данных или файлов для распространения по сети, или хранилище

Экспорт	<p>может быть местом, которое непосредственно доступно пользователю без необходимости блуждания по сети.</p> <p>Эта команда создаёт копию версированной папки, почти как рабочая копия, но без локальных папок <code>.svn</code>.</p>
---------	---