

TortoiseMerge

**Ein Vergleichs- und
Konflikteditor für Windows**

Version 1.12

**Stefan Küng
Lübbe Onken
Simon Large**

TortoiseMerge: Ein Vergleichs- und Konflikteditor für Windows: Version 1.12

von Stefan Küng, Lübbe Onken und Simon Large
Übersetzung: Stefan Küng, Lübbe Onken

Veröffentlicht 2019/03/11 21:25:21 (r28524)

Inhaltsverzeichnis

Vorwort	vi
1. TortoiseMerge ist frei!	vi
2. Danksagung	vi
1. Einleitung	1
1.1. Übersicht	1
1.2. TortoiseMerge's Geschichte	1
2. Grundlegende Konzepte	3
2.1. Anzeigen und Zusammenführen von Unterschieden	3
2.2. Konflikte bearbeiten	3
2.3. Patches Anwenden	4
3. TortoiseMerge benutzen	5
3.1. Ansichten	5
3.1.1. Anzeigen / Zusammenführen	5
3.1.2. Patches Anwenden	7
3.2. Schaltflächen und andere Steuerelemente	8
3.3. Symbole für den Zeilenstatus	9
3.4. Zusammenführen / Konflikte bearbeiten	9
3.5. Dateien öffnen	10
3.5.1. Anzeigen / Zusammenführen	11
3.5.2. Patches Anwenden	11
3.6. Einstellungen	11
3.6.1. Einstellungen Allgemein	11
3.6.2. Einstellungen Farben	12
A. Tastaturkürzel	14
A.1. Tastaturkürzel	14
B. TortoiseMerge automatisieren	16
B.1. TortoiseMerge Kommandozeilenbefehle	16
Glossar	18

Abbildungsverzeichnis

1.1. Dateikonflikt	1
2.1. Dateikonflikt	4
3.1. Vergleich übereinander in einem Bereich	5
3.2. Vergleich nebeneinander mit zwei Bereichen	6
3.3. Zusammenführen mit drei Bereichen	7
3.4. Patchliste	8
3.5. Der Öffnen-Dialog	10
3.6. Die allgemeinen Einstellungen	11
3.7. Die Farbeinstellungen	12

Tabellenverzeichnis

B.1. Liste der Kommandozeilenbefehle 16

Vorwort



Tortoise Merge

Wann immer Sie an einem Projekt arbeiten, sei es ein Open Source Projekt wo mehrere Leute den Sourcecode einsehen, oder sei es ein kommerzielles Projekt, in dem Sie eine Komponente mit Sourcecode verkaufen, wird es immer Leute und Kunden geben, die Fehler finden oder Ihr Projekt verbessern wollen. Sie tun dies indem sie Ihnen so genannte Patchdateien senden. Sie müssen dann diese Patchdateien durchsehen und entscheiden ob Sie die Änderungen in Ihr Projekt übernehmen oder nicht.

TortoiseMerge hilft Ihnen bei diesen beiden Aufgaben:

- Patchdateien überprüfen
- Patchdateien anwenden

TortoiseMerge hilft Ihnen ebenfalls beim Auflösen von Konflikten, die bei der Arbeit mit einem Versionskontrollsystem auftreten können, wenn andere gleichzeitig an denselben Dateien arbeiten wie Sie.

1. TortoiseMerge ist frei!

TortoiseMerge ist freie Software. Sie brauchen nichts dafür zu bezahlen. Sie können das Programm benutzen wie Sie wollen. TortoiseMerge wird unter der GP Lizenz (GPL) entwickelt.

TortoiseMerge ist ein Open Source Projekt. Das bedeutet, dass sie Zugang zum Quellcode des Programms haben. Sie können den Quellcode unter folgender URL <http://code.google.com/p/tortoisesvn/source/browse/> [<https://sourceforge.net/p/tortoisesvn/code/HEAD/tree/>] ansehen. Die aktuellste Version (an der wir gerade arbeiten) ist immer im Ordner `/trunk/` zu finden. Die bereits veröffentlichten Versionen finden sich im Ordner `/tags/`.

Spenden!

Obwohl TortoiseSVN und TortoiseMerge für Sie als Anwender kostenlos sind, können Sie uns auf vielfältige Weise helfen indem Sie uns Patches schicken und bei der Entwicklung direkt mithelfen. Sie können uns natürlich bei den endlosen Stunden die wir für das Projekt vor dem Bildschirm verbringen, aufheitern.

Während wir an TortoiseMerge und TortoiseSVN arbeiten hören wir gerne Musik. Und weil wir viele viele Stunden an TortoiseSVN arbeiten, benötigen wir immer wieder *neue* Musik. Aus diesem Grund haben wir im Internet Seiten eingerichtet, von denen aus Sie uns Musik-CD's und auch mal eine DVD schenken können: <https://tortoisesvn.net/donate.html>. Bitte werfen Sie auch einen Blick auf die Liste der Personen, die das Projekt z.B. durch Übersetzungen unterstützt haben.

2. Danksagung

Stefan Küng

für die harte Arbeit an TortoiseMerge

Lübbe Onken

für die wundervollen Icons, das Logo, die unermüdliche Fehlersuche und die Dokumentation

Das Tigris Style Projekt

für einige der Stile, die in dieser Dokumentation wiederverwendet werden.

Unsere Helfer

für die Patches, Fehlermeldungen und dafür, dass sie anderen helfen, indem Sie Fragen auf unserer Mailingliste beantworten.

Unsere Spender
für viele Stunden voll toller Musik, die sie uns geschenkt haben.



Kapitel 1. Einleitung

1.1. Übersicht

TortoiseMerge ist ein freies Open Source Programm. Sie können damit Unterschiede in Textdateien zeigen, diese Unterschiede zusammenführen und auch Standard-Diff Dateien, sogenannte *Patchdateien* auf Ihre Arbeitskopie anwenden.

Wenn Sie mit Textdateien, z.B. Quelltexten ihrer bevorzugten Programmiersprache oder HTML/XML Dateien ihrer Dokumentation oder Webseite arbeiten, werden Sie öfters in eine Situation kommen in der Sie verschiedene Versionen von solchen Dateien vergleichen möchten. Manchmal erhalten Sie eine veränderte Datei von jemand anderem, manchmal möchten Sie einfach nachsehen was Sie an einer Datei schon alles geändert haben.

Wenn Sie mit einem Versionskontrollsystem arbeiten (z.B. Subversion) dann werden Sie manchmal einen *Konflikt* erhalten, wenn Sie ihre Arbeitskopie aktualisieren. Dies passiert wenn jemand anders dieselben Stellen derselben Datei geändert hat wie Sie. Dann müssen Sie diesen Konflikt von Hand wieder auflösen. Und TortoiseMerge hilft Ihnen dabei.

Das folgende Diagramm zeigt die Beziehung zwischen den verschiedenen Dateien bei einem Konflikt:

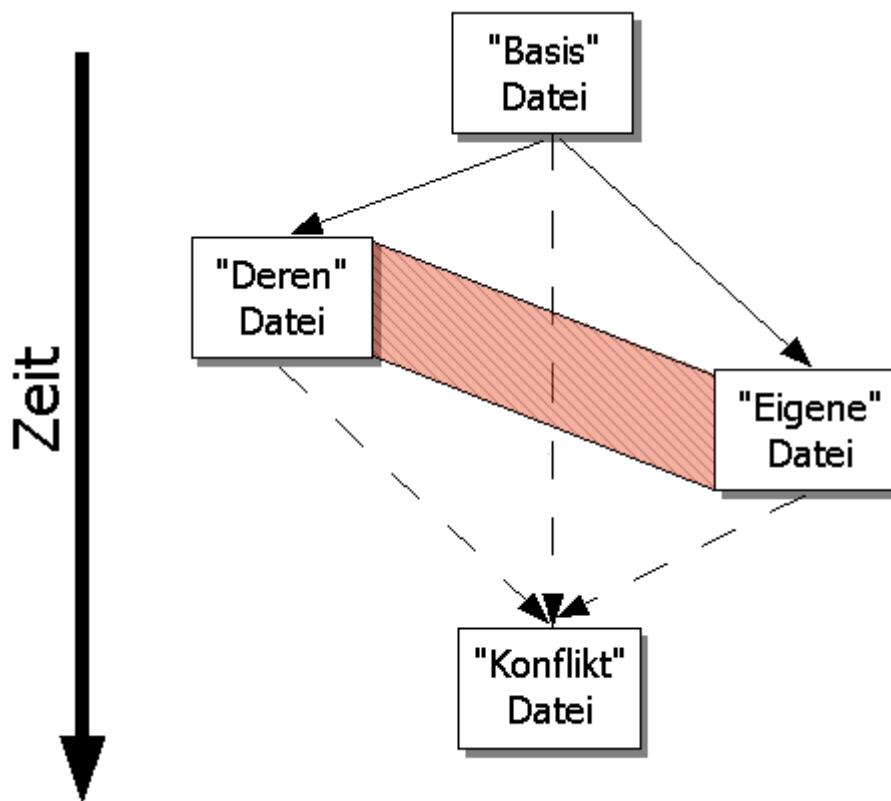


Abbildung 1.1. Dateikonflikt

1.2. TortoiseMerge's Geschichte

Während der Arbeit an *TortoiseSVN* [<https://tortoisesvn.net>] war es jedes Mal sehr aufwändig, wenn jemand uns Patches mit Fehlerkorrekturen oder Erweiterungen zugeschickt hat, diese Patches auch einzubinden. Häufig waren die Patches veraltet und konnten mit dem Kommandozeilenprogramm <https://unxutils.sourceforge.net/> nicht appliziert werden, da wir an den Dateien arbeiteten und diese manchmal bereits geändert waren.

Das war auch der Hauptgrund, weshalb TortoiseSVN keinen Befehl `TortoiseSVN → Erstelle Patch` implementiert hatte: wir wollten anstatt der Patchdateien lieber die ganzen Dateien von den Anwendern haben, weil es für uns einfacher war Änderungen auf diese Weise zum Projekt hinzuzufügen.

Also haben wir viel Zeit damit verbracht, das Internet nach einem guten grafischen Tool abzusuchen welches Patchdateien zu Projekten hinzufügen konnte. Aber alles was wir fanden war, dass ein solches Programm nicht existierte. Um eine lange Geschichte zu einem Ende zu bringen: wir haben dann beschlossen, dass wenn niemand anders ein solches Tool anbietet, wir eben selbst eines schreiben müssen. So entstand TortoiseMerge.

Da TortoiseSVN schon die Subversion Bibliothek benutzt welche gute Vergleichsfunktionen enthält, war es nur natürlich, dieselbe Funktionsbibliothek anstelle von GNU Diff für TortoiseMerge zu nutzen.

Mit Subversion 1.7 wurde der `svn patch` Befehl eingeführt, um einen Patch auf eine Arbeitskopie anzuwenden. TortoiseMerge verwendet nun die selben Funktionen wie Subversion, um Unterschiede im Verhalten zu vermeiden.

Kapitel 2. Grundlegende Konzepte

TortoiseMerge hat verschiedene Verwendungszwecke:

- Anzeigen von Unterschieden zwischen zwei Dateien, Zusammenführen von gewünschten und Entfernen unerwünschter Änderungen in einer der beiden Dateien.
- Bearbeiten von Konflikten zwischen lokalen Änderungen und dem Subversion Projektarchiv in Folge einer Aktualisierung.
- Patches anwenden und überprüfen.

2.1. Anzeigen und Zusammenführen von Unterschieden

In diesem Modus vergleichen Sie zwei Dateien. Die links angezeigte Datei stellt das Original dar (manchmal auch als *Deren* bezeichnet) und die rechte Datei die geänderte (manchmal auch als *Meine* bezeichnet).

Sie können folgende einfache, zeilenbasierte Änderungen an der Datei im rechten Bereich vornehmen:

- Geänderten Text auf das Original im linken Bereich zurücksetzen.
- Eine Kombination aus beiden Blöcken, entweder *Deren* vor *Meinem* oder *Meinen* vor *Deren* übernehmen.

Sie können die Datei im rechten Bereich wie in einem Texteditor bearbeiten. Solche Zeilen werden durch einen Stift gekennzeichnet. Beachten Sie bitte, dass Sie wenn Sie Zeilen- oder Blockoperationen vornehmen, diese zuerst - *vor manuellen Änderungen* - durchführen, da es für TortoiseMerge sonst unmöglich wird, die Beziehungen zur den Originaldateien aufrecht zu erhalten.

2.2. Konflikte bearbeiten

Dies wird manchmal als Dreiwegevergleich bezeichnet und wird in drei Bereichen angezeigt. Tatsächlich sind aber vier Dateien involviert. Die nicht angezeigte Datei ist die gemeinsame Basis, der letzte Vorgänger der beiden, sich in Konflikt befindenden Dateien. Die Beziehungen zwischen den drei Dateien stellen sich wie folgt dar:

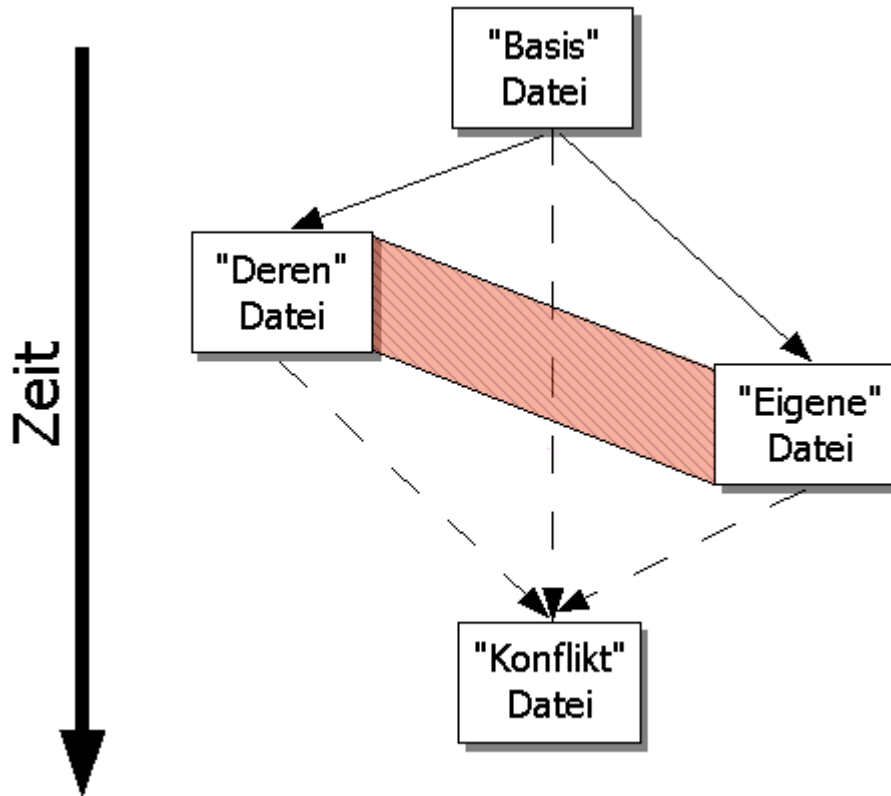


Abbildung 2.1. Dateikonflikt

Die *Basis* Datei entspricht der ältesten Version einer Datei, von welcher aus *Sie* und *die Anderen* Änderungen vorgenommen haben. *Meine* entspricht dabei der Basisdatei mit allen Änderungen welche *Sie* selbst vorgenommen haben und wird im rechten Bereich angezeigt. *Deren* entspricht der Basisdatei mit allen Änderungen welche *die Anderen* gemacht haben und wird im linken Bereich angezeigt. Der untere Bereich enthält die Ausgabedatei. In dieser werden die Konflikte aufgelöst.

Im Dreiwegevergleich, können Sie Blöcke aus *Meiner* oder *Deren* Datei oder aus beiden verwenden. Nur gehen in diesem Fall die Änderungen in die Zieldatei, die im unteren Bereich angezeigt wird.

2.3. Patches Anwenden

Ein Patch ist eine Datei im Subversion Standard-Diff Format, die die Informationen über Änderungen an einem Satz von Dateien enthält. Der Patch kann von einem anderen Entwickler stammen, so dass Sie seine Änderungen anwenden, überprüfen und gegebenenfalls in das Projektarchiv übertragen können. Die Patchdatei kann auch intern durch TortoiseSVN beim Vergleich zweier Ordner generiert werden. Dies geschieht zum Beispiel wenn Sie eine Revision im Projektarchiv mit der Arbeitskopie oder auch zwei Revisionen im Projektarchiv oder zwei verschiedene Pfade im Projektarchiv miteinander vergleichen.

In jedem Fall wird TortoiseMerge ein kleines Fenster mit den vom Patch betroffenen Dateien anzeigen. Durch einen Doppelklick auf eine der Dateien wird die Ausgangsrevision geholt und die Änderungen werden angewendet. Im linken Bereich wird die Originaldatei und im rechten die veränderte Datei angezeigt.

Sie können die Datei im rechten Bereich genau so wie im Vergleichsmodus bearbeiten.

Kapitel 3. TortoiseMerge benutzen

3.1. Ansichten

TortoiseSVN besitzt drei verschiedene Ansichten: mit einem, zwei oder drei Bereichen. Die Ein-/Zweifenstersicht dient zum Betrachten von Änderungen und die Dreifenstersicht zum Auflösen von Konflikten.

3.1.1. Anzeigen / Zusammenführen

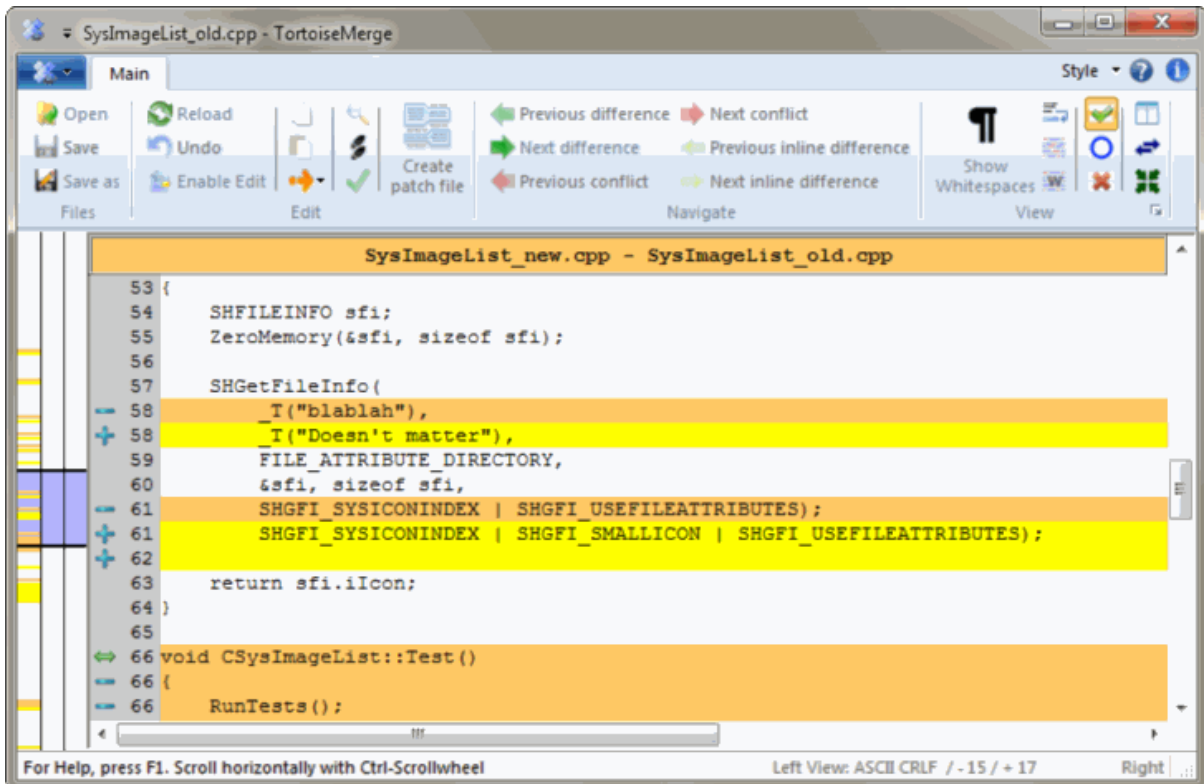


Abbildung 3.1. Vergleich übereinander in einem Bereich

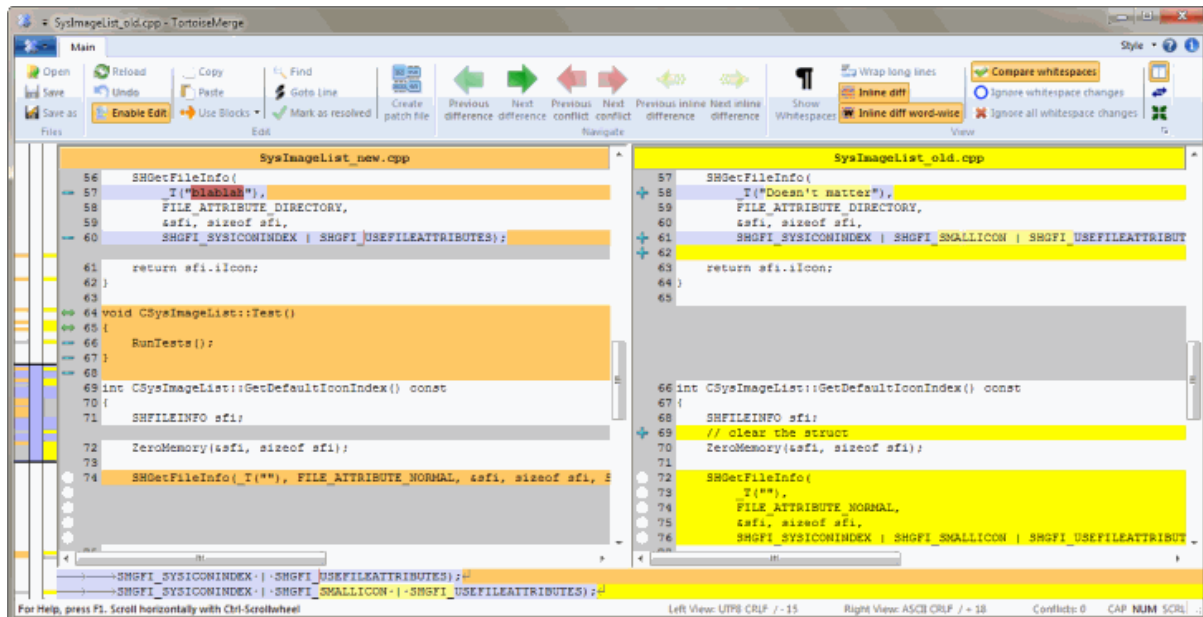


Abbildung 3.2. Vergleich nebeneinander mit zwei Bereichen

Die Zweifensicht bietet einige Möglichkeiten, die in der Einfensicht nicht zur Verfügung stehen:

- Unterschiede innerhalb einer Zeile werden in verschiedenen Farben angezeigt. Hinzugefügte Zeichen werden in einer helleren Farbe dargestellt, aber sie können selbstverständlich die verwendeten Farben einstellen. Gelöschte Abschnitte werden durch eine dunkelbraune senkrechte Linie angezeigt.
- Die Reorganisation von Quellcode führt häufig zu größeren Änderungen an Leerzeichen (Leerzeichen, Tabulatoren, Leerzeilen) ohne tatsächliche Änderungen am Programmcode. Zum Beispiel spalten Sie eine sehr lange Zeile in mehrere kurze Zeilen auf oder lassen einen automatischen Quellcodeformatierer über den Programmcode laufen.

Solche Änderungen werden durch einen weißen Kreis am linken Rand der Bereiche dargestellt. Wenn Sie einen Kreis sehen, wissen Sie sofort, dass keine echte Codeänderung vorliegt und Sie den Block nicht genauer untersuchen müssen.

- Manuelles Bearbeiten der rechten Datei ist in der Zweifensicht möglich. Änderungen werden durch ein Stiftsymbol gekennzeichnet. Manuelles Bearbeiten der linken Datei ist nur möglich, wenn die Schaltfläche Bearbeiten aktivieren bei aktiver linker Seite gedrückt wurde.

Für den Fall dass Sie drei Dateien zusammenführen möchten, zeigt TortoiseMerge Ihnen diese in drei Bereichen an. Die gleiche Ansicht wird auch für das Auflösen von Konflikten verwendet.

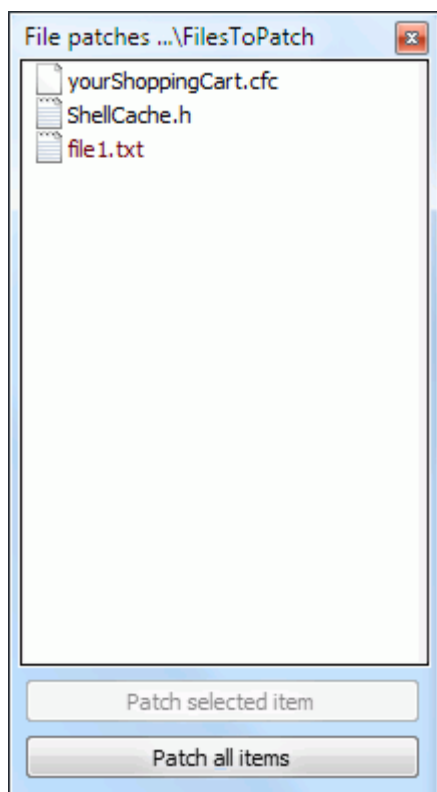


Abbildung 3.4. Patchliste

Wenn der Dateiname schwarz ist, kann der Patch ohne Probleme übernommen werden, weil die Zieldatei dem in der Patchdatei angegebenen Stand entspricht. Ist der Dateiname hingegen rot, dann kann der Patch nicht direkt übernommen werden, da die Zieldatei schon anderweitig geändert wurde.

Das Patchdateifenster hat ein Kontextmenü, das Ihnen eine Vorschau der Änderungen an der aktuellen Datei ermöglicht (Anwenden ohne zu speichern), den Patch anzuwenden und zu speichern sowie den Patch auf *alle* Dateien in der Liste anzuwenden und die Änderungen zu speichern. Ein Doppelklick startet die Vorschau.

3.2. Schaltflächen und andere Steuerelemente

Abhängig von Ihren Einstellungen, sehen Sie entweder eine Werkzeugleiste oder ein Band mit vielen Schaltflächen. Die Schaltflächen besitzen alle einen Hinweistext, der ihre Funktion erläutert, sobald Sie mit der Maus darüber fahren.

Am linken Rand ist eine Navigationsleiste. Diese liefert einen schnellen Überblick, wo sich Änderungen in der Datei befinden. Die Leiste hat drei Spalten. Die linke Spalte entspricht dem linken Bereich, die rechte Spalte dem rechten Bereich, die mittlere Spalte, falls angezeigt, dem unteren Bereich. In der Einfenstersicht wird nur die linke Spalte benutzt. Die Navigationsleiste kann auch als Bildlaufleiste benutzt werden, wobei sich dann jeweils alle drei Bereiche synchron verschieben.

Wenn Sie auf ein Wort doppelklicken wird jedes Vorkommen des Wortes im gesamten Dokument, sowohl in den Fenstern als auch in der Navigationsleiste hervorgehoben. Doppelklicken Sie erneut auf das Wort, um die Markierung zu entfernen.

Wenn Sie am linken Rand klicken oder wenn Sie innerhalb einer Zeile dreifach klicken, wird die gesamte Zeile ausgewählt.

Unterhalb des Fensters befindet sich die Statusleiste. Diese zeigt die Anzahl der in *Deren* und *Meine* hinzugefügten oder gelöschten Zeilen sowie die Anzahl der verbleibenden Konflikte.

Die Statusleiste enthält außerdem Kombinationslisten, die festlegen, wie die Dateien behandelt werden sollen:

Kodierung

Die Kodierung legt fest, wie die Zeichen in den Ansichten geladen, gespeichert und angezeigt werden. Die häufigste Zeichenkodierung in Englisch ist ASCII, aber sie können auch UTF8, UTF16LE, UTF16BE, UTF32LE und UTF32BE sowohl mit als auch ohne BOM (Byte Order Mark) einstellen.

Zeilenumbrüche

Unter Windows ist wird üblicherweise CRLF als Zeilenumbruch verwendet, aber Sie können die von Ihnen gewünschte Variante auswählen. Bitte beachten Sie, dass wenn Sie die Zeilenumbrüche ändern, *alle* Zeilenumbrüche in der gesamten Datei entsprechend geändert werden, auch wenn sie beim Laden unterschiedlich waren.

Tabulator

Die Option oberhalb der Kombinationsliste legt fest, ob Tabulatoren oder Leerzeichen eingefügt werden, wenn Sie die Tabulatortaste drücken. Wenn die Option *Intelligenter Tabulator* aktiviert ist, wird über einen Algorithmus festgestellt, welche Variante die bessere von beiden ist.

Die Tabulatorgröße legt fest, wie viele Leerzeichen eingefügt werden, wenn die Tabulatortaste gedrückt wird, beziehungsweise wie weit das nächste Wort bei einem Tabulatorzeichen eingerückt wird.

3.3. Symbole für den Zeilenstatus

In geänderten Zeilen wird die Art der Änderung durch ein Symbol angezeigt.



Eine Zeile wurde hinzugefügt.



Eine Zeile wurde entfernt.



Eine Änderung wurde durch das Zurücksetzen einer Zeile auf ihren Ursprungszustand zurückgenommen.



Diese Zeile enthält nur Änderungen in Leerzeichen. Falls mehrere aufeinanderfolgende Zeilen markiert sind, wurde eventuell ein Absatz neu umgebrochen.



Eine Zeile wurde manuell in TortoiseSVN geändert.



Diese Zeile enthält einen Konflikt.



Diese Zeile enthält einen Konflikt, der aufgrund der Vergleichsoptionen für Zeilenende oder Leerzeichen nicht sichtbar ist.



Diese Zeile wurde an eine andere Stelle verschoben.

3.4. Zusammenführen / Konflikte bearbeiten

TortoiseMerge *zeigt* Ihnen nicht nur die Unterschiede zwischen Dateien an, sondern lässt Sie diese Änderungen auch übernehmen und Konflikte lösen.

Wenn Sie in der Zweifensicht sind, dann können nur Sie die Datei im rechten Bereich (*Meiner Version*) ändern. Um Unterschiede aus der linken Datei (*Derer*) in die rechte zu übernehmen, machen Sie einen Rechtsklick auf die unterschiedliche Zeile und wählen Sie den gewünschten Befehl aus dem Kontextmenü aus.

Manchmal benötigen Sie beide Textblöcke. Das Kontextmenü bietet Ihnen die entsprechenden Funktionen Kontextmenü → Verwende beide Blöcke (diesen zuerst) und Kontextmenü → Verwende beide Blöcke (diesen zuletzt) an.

Sie können die Ausgabedatei wie in einem Texteditor bearbeiten. Solche Zeilen werden durch einen Stift gekennzeichnet. Beachten Sie bitte, dass Sie wenn Sie Zeilen- oder Blockoperationen vornehmen, diese zuerst - *vor manuellen Änderungen* - durchführen, da es für TortoiseMerge sonst unmöglich wird, die Beziehungen zur den Originaldateien aufrecht zu erhalten.

Wenn Sie einen Dreiwegevergleich durchführen (auch *Zusammenführen*- oder *Konfliktsicht* genannt), können Sie nur die Ergebnisdatei im unteren Bereich bearbeiten. Wie auch in der Zweifensicht können Sie durch Rechtsklicken auf Zeilen im Konflikt ein Kontextmenü anzeigen lassen und dann auswählen, wie Sie den Konflikt auflösen möchten. Sie können damit den rechten oder linken Block auswählen, aber auch beide Blöcke entweder „Meiner vor Deren“ oder „Deren vor Meinem“ in die Zieldatei übernehmen.

Manchmal wird in Subversion ein Dateikonflikt angezeigt, es sind jedoch keine Unterschiede sichtbar, wenn Sie die Datei in TortoiseMerge betrachten. Das kann an ihren Einstellungen für den Vergleich von Leerzeichen oder Zeilenumbrüchen liegen. Falls Sie beschlossen haben, solche Änderungen zu ignorieren, werden die entsprechenden Zeilen durch das *Konflikt Ignoriert* Symbol gekennzeichnet. Um den Konflikt aufzulösen müssen Sie trotzdem eine der beiden Varianten auswählen.



Wichtig

Beachten Sie, dass wenn TortoiseMerge nochmals mit den selben Dateien aufgerufen wird, alle Änderungen der Ergebnisdatei *verworfen werden*, egal ob durch TortoiseMerge oder manuelles Bearbeiten und die Ergebnisdatei im Ursprungszustand wieder hergestellt wird.

3.5. Dateien öffnen

Wenn Sie TortoiseMerge ohne Kommandozeilenbefehle starten, dann müssen Sie die Dateien mit dem **Datei → Öffnen** von Hand öffnen.

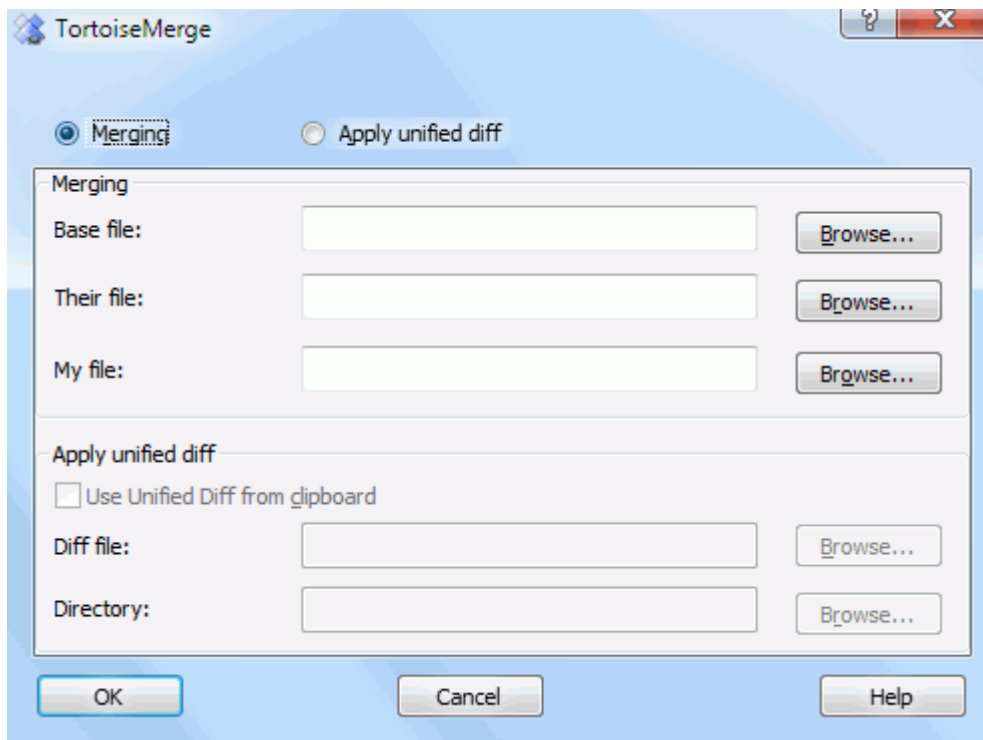


Abbildung 3.5. Der Öffnen-Dialog

Das erste was Sie entscheiden müssen ist ob Sie Dateien vergleichen / zusammenführen möchten oder ob Sie eine Patchdatei anwenden möchten. Je nachdem was Sie wählen, werden die entsprechenden Schaltflächen und Eingabefelder aktiviert.

3.5.1. Anzeigen / Zusammenführen

Wenn Sie Dateien vergleichen / zusammenführen möchten, müssen Sie mindestens zwei der drei möglichen Pfade für *Basis*, *Meiner* und *Deren* angeben. Wenn Sie nur zwei Dateien auswählen, wird TortoiseMerge ihnen die Unterschiede zwischen diesen Dateien entweder in einem Bereich übereinander oder in zwei Bereichen nebeneinander anzeigen.

Wenn Sie drei Dateien zusammenführen, zeigt TortoiseMerge Ihnen diese in einer Dreifensicht. Diese Ansicht wird im Allgemeinen zum Auflösen von Konflikten verwendet. Die Ausgabedatei hat in diesem Fall keinen Namen und Sie müssen das Ergebnis mittels Datei → Speichern unter... sichern.

3.5.2. Patches Anwenden

Wenn Sie eine Patchdatei anwenden möchten, müssen Sie sowohl die Patchdatei selbst, als auch den Zielordner angeben.

3.6. Einstellungen

3.6.1. Einstellungen Allgemein

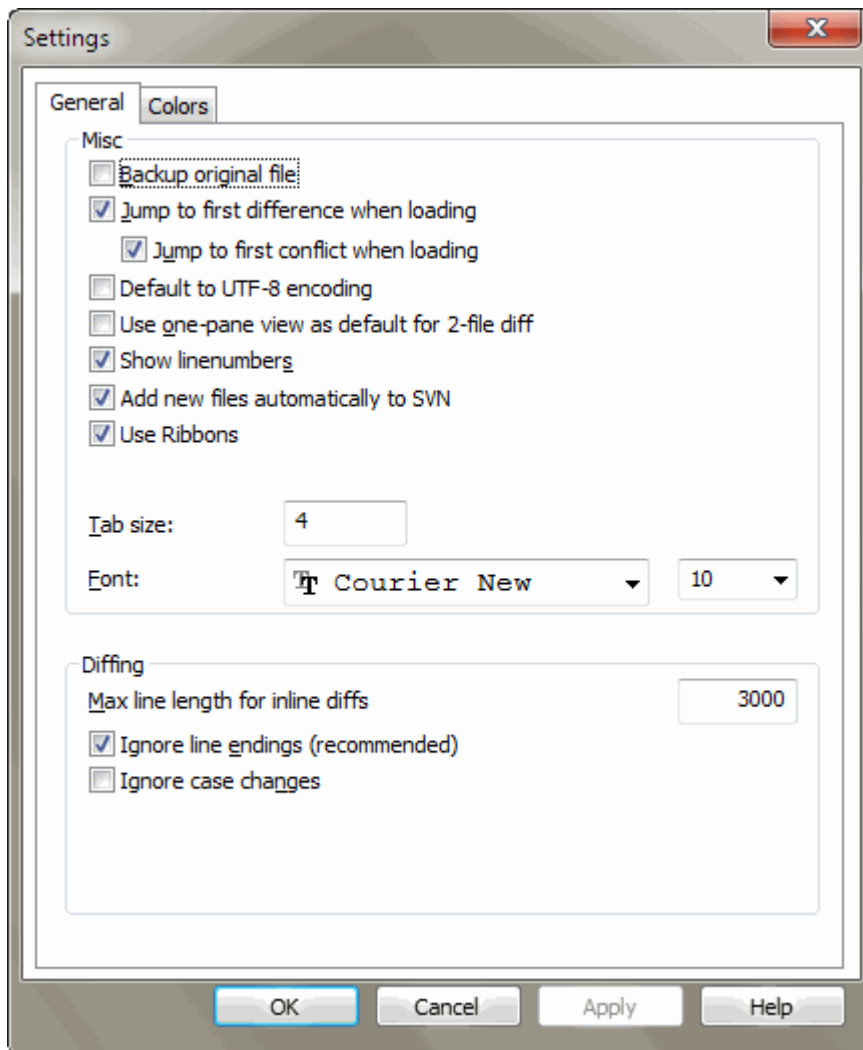


Abbildung 3.6. Die allgemeinen Einstellungen

Die meisten Optionen auf dieser Seite sind selbsterklärend. Ein paar benötigen jedoch ergänzende Informationen.

Erstelle Sicherheitskopie benennt die Originaldatei in der Arbeitskopie in `Dateiname.bak` um, bevor die Änderungen gespeichert werden.

Wenn **Standardmäßig UTF-8-Codierung** festgelegt ist, werden ANSI-Dateien als UTF-8 codiert geladen und, wenn bearbeitet, als solche gespeichert.

Max Zeilenlänge für Inline diffs TortoiseMerge kann langsam werden, wenn Inline Diffs für sehr lange Zeilen angezeigt werden sollen. Deshalb werden nur Zeilen, die kürzer als 3000 Zeichen sind mit Inline Diffs angezeigt. Sie können diesen Wert hier ändern.

Ignoriere Zeilenende verbirgt Änderungen, die nur aus unterschiedlichen Zeilenumbrüchen bestehen.

Ignoriere Groß- / Kleinschreibung verbirgt Groß- / Kleinbuchstaben Änderungen. Dies ist besonders nützlich bei Anwendungen wie Visual Basic, die die Schreibweise von Variablen ohne Warnung ändern.

3.6.2. Einstellungen Farben

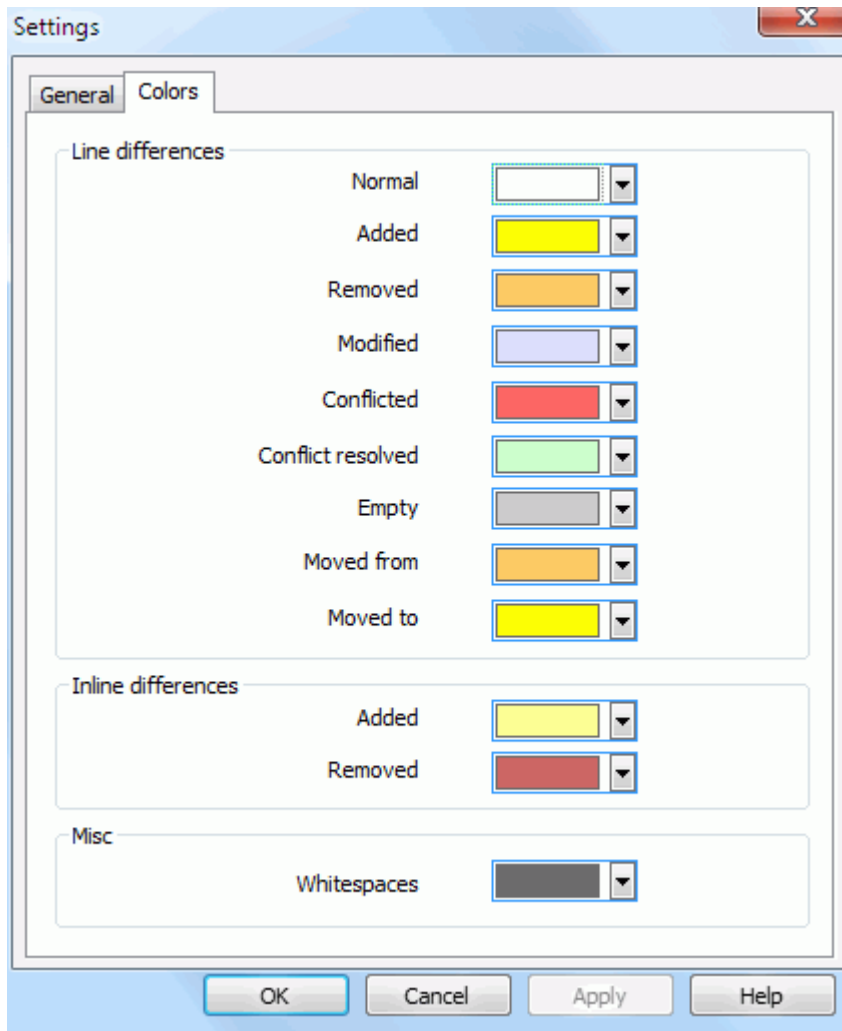


Abbildung 3.7. Die Farbeinstellungen

Auf dieser Seite können Sie die Hintergrundfarben zum Hervorheben der verschiedenen Änderungen einstellen.

Normal

Alle nicht veränderten Zeilen oder Zeilen deren Änderungen verborgen wurden.

Hinzugefügt

Zeilen, die hinzugefügt wurden.

Entfernt

Zeilen, die gelöscht wurden.

Verändert

Bei Zeilen mit geringem Änderungsumfang werden die Unterschiede innerhalb der Zeile farbig hervorgehoben. Diese Farbe wird für *unveränderte* Teile der Zeile verwendet. Modifizierte Teile werden durch die unten beschriebenen Farben für hinzugefügt/gelöscht gekennzeichnet. Wenn **Änderungen in Zeile farbig hervorheben** nicht aktiv ist, wird diese Farbe nicht verwendet und geänderte Zeilen werden als Ersetzungen dargestellt.

Konflikt

Dieselbe Zeile wurde in beiden Dateien geändert.

Konflikt aufgelöst

Falls dieselbe Zeile in beiden Dateien geändert wurde und Sie mittlerweile eine der beiden Varianten ausgewählt haben.

Leer

Zeilen wurden im gegenüberliegenden Bereich hinzugefügt und keine entsprechende Zeile existiert in diesem Bereich.

Hinzugefügter Text

Wenn Differenzen innerhalb der Zeile angezeigt werden, wird hinzugefügter Text mit dieser Farbe markiert.

Gelöschter Text

Wenn Differenzen innerhalb der Zeile angezeigt werden, wird gelöschter Text mit dieser Farbe markiert.

Diverse Leerzeichen

Zeichen zur Hervorhebung von Leerraum werden in einer anderen Farbe als der normale Text angezeigt.

Anhang A. Tastaturkürzel

Liste der Tastaturkürzel und Befehle.

A.1. Tastaturkürzel

Strg+Q, Strg+W, Escape

Das Programm beenden

Strg+C

Den markierten Text in die Zwischenablage kopieren

Strg+X, Umsch+Entf

Den markierten Text in die Zwischenablage ausschneiden

Strg+V, Umsch+Einf

Den markierten Text aus der Zwischenablage einfügen

Strg+Z, Alt+Rück

Die letzten Änderungen rückgängig machen

Strg+F

Den Dialog zum Suchen und Ersetzen von Text öffnen

Strg+O

Dateien zum Vergleichen/Zusammenführen öffnen

Strg+S

Änderungen speichern

Strg+Umsch+S

Speichern unter...

F7

Zum nächsten Unterschied gehen

Umsch+F7

Zum vorherigen Unterschied gehen

F8

Zum nächsten Konflikt gehen

Umsch+F8

Zum vorherigen Konflikt gehen

Strg+D

Zwischen Einfeldersicht und Zweifeldersicht wechseln

Strg+R

Dateien neu laden und alle Änderungen rückgängig machen

Strg+T

Platzhalter für nicht druckbare Zeichen anzeigen/verstecken

Strg+L

Unveränderte Abschnitte anzeigen/verstecken

Strg+P

Zeilen umbrechen/nicht umbrechen

Strg+G

Gehe zu Zeile

Strg+A

Gesamten Text auswählen

Strg+P

Zeilen umbrechen/nicht umbrechen

Strg+U

Ansichten umschalten

Strg+Mausrad

Anzeige links/rechts rollen

Strg+Tab

Zwischen linker/rechter/unterer Ansicht wechseln

Anhang B. TortoiseMerge automatisieren

TortoiseMerge kann auch über die Kommandozeile gestartet werden. Auf diese Weise können die zu vergleichenden Dateien direkt ohne den Öffnen Dialog angegeben werden. Dies ist vor allem dann nützlich, wenn Sie TortoiseMerge aus einer anderen Anwendung heraus aufrufen wollen.

B.1. TortoiseMerge Kommandozeilenbefehle

Die meisten Befehle benötigen zusätzliche Informationen, wie einen Pfad oder einen anderen Parameter. Zur Übergabe des Parameters hängen Sie einfach einen ':' an den Befehl an und schreiben den Parameter dahinter. Zum Beispiel:

```
/base:"C:\Ordner\Meine Basisdatei.txt"
```

Befehl	Beschreibung
/?	Zeigt einen Dialog mit allen möglichen Kommandozeilenbefehlen an.
/help	Das Gleiche wie ?.
/base	Definiert die <i>Basisdatei</i> für einen Dreiwegevergleich. Diese ist der gemeinsame Vorgänger der zu vergleichenden Dateien, auch wenn sie nicht angezeigt wird. Beim Zweiwegevergleich wird die Datei im linken Bereich angezeigt.
/basename	Der Anzeigename der <i>Basisdatei</i> . Er wird anstatt des Dateinamens in der Titelzeile oder beim Dreiwegevergleich in einem Hinweistext angezeigt.
/basereflectedname	Der Name für die Editor-Konfigurationsschablonen.
/theirs	Definiert <i>Deren</i> Datei einen Dreiwegevergleich, die im linken Bereich angezeigt wird.
/theirsname	Der Anzeigename von <i>Deren</i> Datei. Er wird anstatt des Dateinamens in der Titelzeile angezeigt.
/theirsreflectedname	Der Name für die Editor-Konfigurationsschablonen.
/mine	Definiert <i>Meine</i> Datei, die sowohl beim Dreiwege- als auch im Zweiwegevergleich im rechten Bereich dargestellt wird.
/minename	Der Anzeigename von <i>Meiner</i> Datei. Er wird anstatt des Dateinamens in der Titelzeile angezeigt.
/minereflectedname	Der Name für die Editor-Konfigurationsschablonen.
/merged	Legt die <i>zusammengeführte</i> Ergebnisdatei fest, die in Vergleichen verwendet wird. Dies ist der Dateipfad, in dem das Ergebnis des Zusammenführens, bzw. des Konfliktauflösens gespeichert wird. Wenn dieser Parameter für einen Dreiwegevergleich nicht angegeben wurde, fragt TortoiseSVN den Anwender, wo die Ergebnisse gespeichert werden sollen. Wenn dieser Parameter für einen Zweiwegevergleich nicht angegeben wurde, wird TortoiseSVN automatisch die im rechten angezeigte Datei als Ziel verwenden.
/mergedname	Der Anzeigename der <i>Zieldatei</i> . Er wird anstatt des Dateinamens in der Titelzeile angezeigt.
/mergedreflectedname	Der Name für die Editor-Konfigurationsschablonen.
/patchpath	Der Pfad in dem ein Patch angewendet werden soll. Wenn Sie diesen Parameter nicht angeben, versucht TortoiseMerge selbst, einen zu den Pfaden in der Patchdatei passenden Basispfad herauszufinden, aber das kann <i>sehr lange</i> dauern.
/saverequired	Diese Option zwingt TortoiseMerge dazu, vor dem Beenden danach zu fragen, ob die Datei gespeichert werden soll, selbst wenn der Anwender keine Änderungen vorgenommen hat.

Befehl	Beschreibung
/saverequiredonconflicts	Diese Option zwingt TortoiseMerge dazu, vor dem Beenden bei Konflikten danach zu fragen, ob die Datei gespeichert werden soll, selbst wenn der Anwender keine Änderungen vorgenommen hat.
/patchoriginal	Der Anzeigename der zu patchenden Originaldatei.
/patchpatched	Der Anzeigename der Zieldatei.
/diff	Der Pfad zu der Patch- / Diffdatei, die auf ein Verzeichnis angewendet werden soll.
/oneway	Zwingt TortoiseMerge mit der Einfeldersicht anstatt der in den Optionen eingestellten Sicht zu starten.
/reversedpatch	Vertauscht die Dateien im rechten und linken Bereich.
/createunifieddiff	Erzeugt eine Standard-Diff Datei (Patchdatei) aus den beiden in /origfile:"pfad_der_originaldatei" und /modifiedfile:"pfad_der_geänderten_datei" übergebenen Dateien. Das Ziel wird in /outfile:"pfad_der_ergebnisdatei" spezifiziert. Wenn /outfile leer ist, wird ein „Datei Speichern“ Dialog angezeigt, in dem der Anwender die Ausgabedatei angeben kann. Hinweis: Wenn /createunifieddiff gesetzt ist, werden alle anderen Parameter ignoriert.
/line	Gibt die Zeilennummern an, in die nach dem Laden der Dateien gesprungen werden soll.
/readonly	Verhindert, dass die Datei bearbeitet wird. Das heißt, die Bearbeitungsfunktionen von TortoiseMerge sind deaktiviert.
/nosvnresolve	Hindert TortoiseSVN daran, nachzufragen, ob eine Datei nach dem Speichern in SVN als aufgelöst markiert werden soll.

Tabelle B.1. Liste der Kommandozeilenbefehle

Sie können aus Kompatibilitätsgründen mit anderen Vergleichsprogrammen auch einfache Dateinamen auf der Befehlszeile übergeben. In der Vereinfachten Form sieht der Aufruf folgendermaßen aus

```
TortoiseMerge BasisDateiPfad MeinDateiPfad [ DerenDateiPfad ]
```

Bei zwei Dateien werden diese miteinander verglichen. Bei drei Dateien wird die erste als BASE Datei verwendet und die beiden anderen in einem Dreivegevergleich mit ihr verglichen.

Glossar

Aktualisieren	Dieser Subversionbefehl holt die neuesten Änderungen aus dem Projektarchiv in Ihre Arbeitskopie und führt dabei die Änderungen von anderen mit Ihren Änderungen zusammen.
Änderungen rückgängig	Subversion hält eine lokale „Basis“-Kopie von jeder Datei in dem Zustand, in dem sie zuletzt aktualisiert wurde in der Arbeitskopie. Wenn Sie Änderungen gemacht haben und diese rückgängig machen wollen, können Sie den Befehl „Rückgängig“ verwenden, um zurück zur Basisversion der Datei zu kommen.
Annotieren	Dieser Befehl steht nur für Textdateien zur Verfügung und er annotiert jede Zeile mit der Revision und dem letzten Autor. In unserer GUI Anwendung namens TortoiseBlame wird zusätzlich die Logmeldung angezeigt, wenn Sie mit der Maus über die Revisionsnummer fahren.
Arbeitskopie	Dies ist Ihr lokaler „Sandkasten“, der Bereich, in dem Sie an versionierten Dateien arbeiten, und er liegt normalerweise auf Ihrer lokalen Festplatte. Sie erzeugen eine Arbeitskopie, indem Sie einen Ordner aus einem Projektarchiv „Auschecken“ und Sie führen Ihre Änderungen mit einem „Übertragen“ in das Projektarchiv zurück.
Auschecken	Ein Subversionbefehl, der eine lokale Arbeitskopie in einem leeren Verzeichnis erstellt, indem versionierte Dateien aus einem Projektarchiv heruntergeladen werden.
BASE Revision	Die aktuelle Basisrevision einer Datei oder eines Ordners in Ihrer <i>Arbeitskopie</i> . Dies ist die Revision in der sich die Datei oder der Ordners befand als zuletzt Auschecken, Aktualisieren oder Übertragen aufgerufen wurde. Die BASE Revision entspricht normalerweise nicht der HEAD Revision.
Bereinigen	Ein Zitat aus dem Subversion Buch: „Bereinigt die Arbeitskopie rekursiv, entfernt dabei Sperren und setzt unvollendete Operationen fort. Wenn Sie jemals einen <i>Arbeitskopie gesperrt</i> Fehler bekommen, führen Sie diesen Befehl aus, um veraltete Sperren zu entfernen und Ihre Arbeitskopie wieder in einen nutzbaren Zustand zu versetzen.“ Beachten Sie, dass in diesem Zusammenhang <i>Sperren</i> lokale Dateisystemsperrern bezeichnet und keine Sperren im Projektarchiv.
Eigenschaft	Zusätzlich zur Versionierung von Dateien und Ordnern erlaubt Subversion die Versionierung von Metadaten für Ordner und Dateien, auch als „Eigenschaften“ bezeichnet. Jede Eigenschaft hat einen Namen und einen Wert, ähnlich wie ein Registrierungsschlüssel. Subversion selbst hat einige spezielle Eigenschaften definiert, welche intern benutzt werden, wie zum Beispiel <code>svn:eol-style</code> . TortoiseSVN nutzt ebenfalls spezielle Eigenschaften, so zum Beispiel <code>tsvn:logminsize</code> . Sie können eigene Eigenschaften mit jedem Namen und Wert erstellen den sie möchten.
Export	Dieser Befehl erzeugt eine Kopie eines versionierten Ordners. Wie eine Arbeitskopie jedoch ohne die lokalen <code>.svn</code> Verzeichnisse.
FSFS	Ein proprietäres Subversion-Dateisystem für Projektarchive. Kann über Netzwerkfreigaben genutzt werden. Seit Version 1.2 Standard für neue Projektarchive.
GPO	Gruppenrichtlinienobjekt.
HEAD Revision	Die neueste Version einer Datei oder eines Ordners im <i>Projektarchiv</i> .

Hinzufügen	Ein Subversion Befehl, der zum Hinzufügen von Dateien oder Ordnern zu einem Projektarchiv benutzt wird. Die neuen Objekte werden beim Übertragen zum Projektarchiv hinzugefügt.
Historie	Zeigt die Revisionshistorie einer Datei oder eines Ordners. Auch als „Log“ bekannt.
Importieren	Ein Subversionbefehl mit dem eine vollständige Ordnerhierarchie mit einer einzigen Revision in ein Projektarchiv importiert wird.
Konflikt	Wenn Änderungen vom Projektarchiv mit den lokalen Änderungen zusammengeführt werden sollen, kann es vorkommen dass diese Änderungen in den selben Zeilen der Datei auftreten. In diesem Fall kann Subversion nicht selbst entscheiden welche Version zu verwenden ist und die Datei wird somit als „in Konflikt“ markiert. Sie müssen die Datei von Hand editieren und den Konflikt auflösen, bevor Sie weitere Änderungen übertragen können.
Konflikt lösen	Wenn sich Dateien in einer Arbeitskopie nach einem Zusammenführen in einem Konflikt befinden, müssen diese Konflikte von einer Person mittels eines Editors (oder vielleicht TortoiseMerge) aufgelöst werden. Dieser Prozess wird als „Konflikt lösen“ bezeichnet. Wenn die Konflikte aufgelöst sind, können diese Dateien als aufgelöst markiert werden, was eine Übertragung erlaubt.
Kopieren	In einem Subversion Projektarchiv können Sie eine Kopie einer einzelnen Datei oder eines ganzen Baumes erstellen. Diese sind als „billige Kopien“ in Form eines Verweises auf das Original implementiert. Somit benötigt selbst die Kopie eines Dateibaumes fast keinen Platz. Eine Kopie behält die Historie ihrer Vorgänger, so dass auch Logmeldungen über die Kopie hinaus verfolgt werden können.
Log	Zeigt das Änderungsprotokoll einer Datei oder eines Ordners. Auch „Historie“ genannt.
Löschen	Wenn Sie ein versioniertes Objekt löschen (und dann Übertragen), existiert die Datei oder Ordner nach der übertragenen Revision nicht mehr im Projektarchiv. Aber selbstverständlich existiert diese noch in früheren Revisionen und es kann immer noch darauf zugegriffen werden. Falls notwendig, können Sie das gelöschte Objekt wieder in die Arbeitskopie kopieren und diese so mit der kompletten Historie „wieder herstellen“.
Patch	Wenn eine Arbeitskopie nur Änderungen an Textdateien aufweist, ist es möglich mit dem Diff-Befehl von Subversion eine Datei zu erstellen, welche diese Änderungen im Standard-Diff Format enthält. Eine Datei mit solchem Inhalt wird üblicherweise als „Patch“ bezeichnet und kann zum Beispiel jemand anderem per Mail geschickt werden, der diese Änderungen dann in seine Arbeitskopie übernehmen kann. Jemand ohne Übertragungsrechte zum Projektarchiv kann so Änderungen vornehmen, diese Änderungen an eine Person mit Übertragungsrechten schicken, welche dann diese Änderungen übertragen kann. Oder man kann eine solche Patchdatei anderen zuerst zur Diskussion schicken, bevor man die Änderungen überträgt.
Projektarchiv	Ein Projektarchiv ist ein zentraler Platz in dem Daten gespeichert und verwaltet werden. Das Projektarchiv kann sowohl auf einem Server liegen und über das Netzwerk angesprochen werden, als auch auf der lokalen Platte liegen.
Revision	Jedes Mal, wenn Sie Änderungen übertragen, wird eine neue „Revision“ im Projektarchiv erstellt. Jede Revision stellt den Status des Projektarchivs zu einem bestimmten Zeitpunkt in der Vergangenheit dar. Wenn Sie in der Zeit

	<p>zurückgehen möchten, können Sie den Zustand des Projektarchivs, wie es zur Revision N war, ansehen.</p> <p>Anders ausgedrückt bezieht sich eine Revision auf eine Menge von Änderungen, die, als diese Revision erstellt wurde, vorgenommen wurden.</p>
Revisionseigenschaft (revprop)	<p>So wie Dateien Eigenschaften haben können, hat auch jede Revision im Projektarchiv Eigenschaften. Einige spezielle Revisionseigenschaften werden automatisch hinzugefügt, wenn die Revision erstellt wird. Diese sind: <code>svn:date</code>, <code>svn:author</code> <code>svn:log</code> welche das Übertragungsdatum, die Person welche die Übertragung gemacht hat und die Logmeldung darstellen. Diese Eigenschaften können verändert werden, aber sie sind nicht versioniert, was bedeutet dass jede Änderung permanent ist und nicht rückgängig gemacht werden kann.</p>
Sperre	<p>Wenn Sie eine Datei sperren, wird diese im Projektarchiv als nicht übertragbar markiert, mit Ausnahme der Arbeitskopie von der aus die Sperre gesetzt wurde.</p>
SVN	<p>Eine häufig verwendete Abkürzung für Subversion.</p> <p>Der Name des spezifischen Protokolls, das von „svnserve“ genutzt wird.</p>
Übertragen	<p>Dieser Subversionbefehl wird verwendet, um die Änderungen in Ihrer lokalen Arbeitskopie zurück in das Projektarchiv zu übertragen.</p>
Umplatzieren	<p>Wenn das Projektarchiv verschoben wird, vielleicht weil Sie es in einen anderen Ordner auf dem Server verschoben haben oder der Domänenname des Servers sich ändert, dann müssen Sie die Arbeitskopie „umplatzieren“, so dass die Projektarchiv-URL zum neuen Ort zeigt.</p> <p>Hinweis: Sie sollten diesen Befehl nur benutzen wenn die Arbeitskopie zur selben Stelle im Projektarchiv selbst zeigt und nur das Projektarchiv selbst verschoben wurde. In allen anderen Situationen müssen Sie den Befehl „Wechseln zu“ verwenden.</p>
Vergleich	<p>Abkürzung für „Zeige Unterschiede“. Sehr nützlich, wenn Sie genau sehen wollen, welche Änderungen Sie vorgenommen haben.</p>
Verzweigen	<p>Ein Begriff der häufig in Versionskontrollsystemen verwendet wird, um zu beschreiben, wenn sich die Entwicklung an einem bestimmten Punkt verzweigt und verschiedenen Wegen folgt. Sie können von der Hauptentwicklungslinie abzweigen, um neue Funktionen zu implementieren ohne den Hauptzweig instabil zu machen. Genauso können Sie eine stabile, freigegebene Version abzweigen, an der Sie nur noch Fehler beseitigen, während die Weiterentwicklung am instabilen Hauptzweig erfolgt. In Subversion sind Zweige als „billige Kopien“ implementiert.</p>
Wechseln	<p>So wie „Aktualisieren zu Revision“ den Zeitpunkt der Arbeitskopie ändert, um auf einen anderen Punkt in der Vergangenheit zu zeigen, ändert „Wechseln zu“ die Position der Arbeitskopie, so dass sie an einen anderen Ort im Projektarchiv zeigt. Dies ist vor allem nützlich, wenn Sie mit verschiedenen Zweigen arbeiten, in denen nur wenige Dateien unterschiedlich sind. Sie können ihre Arbeitskopie zwischen den Zweigen hin und her wechseln wobei jeweils nur die Unterschiede übertragen werden.</p>
Zusammenführen	<p>Der Prozess mit welchem Änderungen vom Projektarchiv zur Arbeitskopie hinzugefügt werden, ohne lokale Änderungen zu beeinträchtigen. Manchmal können diese Änderungen nicht automatisch hinzugefügt werden und die Arbeitskopie befindet sich dann in einem Konflikt.</p>

Das Zusammenführen passiert automatisch, wenn Sie ihre Arbeitskopie aktualisieren. Sie können auch mit dem TortoiseSVN Befehl „Zusammenführen“ spezifische Änderungen von einem Zweig in der Arbeitskopie zusammenführen.