

TortoiseMerge

Windows用の差分・マージツール

Version 1.13

Stefan Küng
Lübbe Onken
Simon Large

TortoiseMerge: Windows用の差分・マージツール: Version 1.13

: Stefan Küng, Lübbe Onken, 、 Simon Large

翻訳: 倉澤 望 (鍋太郎) (nabetaro @ caldron. jp), 藤本 理弘 (fujimoto@internet.ne. jp), 阿部 恒彦

発行日 2019/09/30 20:14:36 (r28662)

目次

序章	vi
1. TortoiseMergeはフリー!	vi
2. 謝辞	vi
1. はじめに	1
1.1. 概要	1
1.2. TortoiseMergeの歴史	1
2. バージョン管理の基本概念	3
2.1. 差分の表示やマージ	3
2.2. 競合の編集	3
2.3. パッチの適用	4
3. TortoiseMergeの使い方	5
3.1. 表示モード	5
3.1.1. 表示・マージ	5
3.1.2. パッチの適用	7
3.2. ボタンやその他のコントロール	8
3.3. 行状態アイコン	9
3.4. 競合のマージと編集	9
3.5. ファイルを開く	10
3.5.1. 表示・マージ	11
3.5.2. パッチの適用	11
3.6. 設定	11
3.6.1. メイン設定ページ	11
3.6.2. 色設定ページ	12
A. キーボードのショートカット	14
A.1. キーボードのショートカット	14
B. TortoiseMergeの自動化	16
B.1. TortoiseMergeのコマンドラインスイッチ	16
用語集	18

図の一覧

1. 1. ファイルの競合	1
2. 1. ファイルの競合	4
3. 1. 1画面表示	5
3. 2. 2画面表示	6
3. 3. 3画面表示	7
3. 4. パッチを適用するファイル一覧	8
3. 5. 開くダイアログ	10
3. 6. 設定の一般ページ	11
3. 7. 設定の色ページ	12

表の一覧

B. 1. 有効なコマンドラインオプションの一覧	16
--------------------------------	----

序章



Tortoise Merge

複数の眼がソースコードを検査してくれるオープンソースプロジェクトや、ソースコードをコンポーネントと共に販売する商用プロジェクトのどちらでも、プロジェクトで作業する際には、バグを発見したり、プロジェクトの改良をしたい人、顧客がいることでしょう。彼らはパッチを送ってくれるので、そのパッチを検査し、適用するかどうか判断しなければなりません。

TortoiseMerge は次のどちらの作業でも利用できます。

- ・ パッチのレビュー
- ・ パッチの適用

また TortoiseMerge は、ソース管理システムやその他で、同じファイルに対して作業したことにより発生する、競合したファイルを整理する助けにもなります。

1. TortoiseMergeはフリー!

TortoiseMerge はフリーです。お金を払う必要はありませんし、どんな用途にも使用できます。 GNU Public License (GPL) のもとで開発されています。

TortoiseMerge はオープンソースプロジェクトです。つまり、あなたはこのプログラムのソースコードに対して完全な読み取りアクセス権を持っています。ソースコードは、<https://sourceforge.net/p/tortoisesvn/code/HEAD/tree/> から閲覧することができます。開発中の最新版は /trunk/ 以下にあり、リリースされた版は /tags/ 以下にあります。

寄付をお願いします

TortoiseSVNとTortoiseMergeはフリーですが、パッチを送ったり、開発の中で重要な役割を担ったりして、開発者をサポートすることができます。また、コンピューターの前で延々と時間を過ごす私たちを応援することができます。

TortoiseSVNで作業している間、私たちは音楽を聴くのが大好きです。我々はプロジェクトに多くの時間を費やしているため、多くの音楽が必要です。そこで、私たちの好きな音楽のCDやDVDのウィッシュリストを、<http://tortoisesvn.net/donate.html> [<https://tortoisesvn.net/donate.html>] で公開しています。また、パッチや翻訳を提供してプロジェクトに貢献している人のリストもご覧ください。

2. 謝辞

Stefan Küng

TortoiseMerge の大変な作業に

Lübbe Onken

きれいなアイコン、ロゴ、バグつぶし、文書整備に

The Tigris Style project

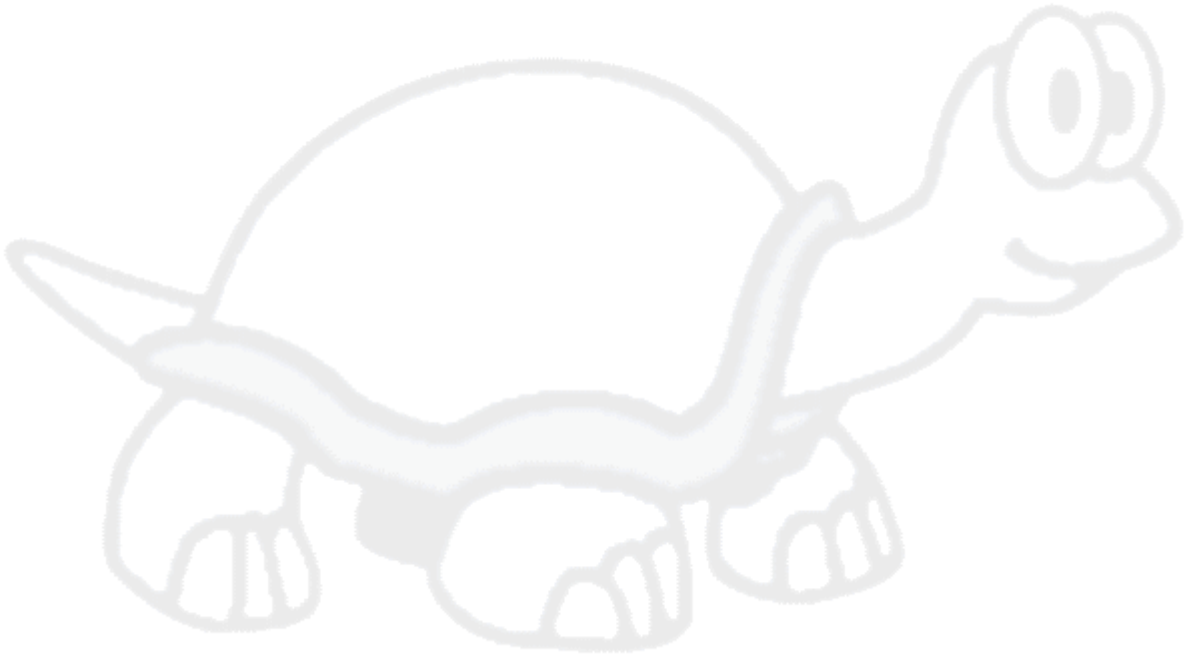
本書で再利用されているスタイルに

われらが協力者達

パッチや、バグレポート、新しいアイデア、メーリングリストでの質疑応答などの助け合いに

われらが寄進者達

送ってくれた音楽で楽しんだ時間に



第1章 はじめに

1.1. 概要

TortoiseMerge はフリーでオープンソースのアプリケーションです。テキストファイルの差分を表示したり、変更点をマージしたり、パッチと呼ばれる Unified 差分ファイルを検査したり適用したりすることができます。

任意のプログラム言語のソースファイルや、ドキュメントや Web サイトの html/xml ファイルといったテキストファイルで作業していると、そのファイルの異なるバージョンと比較したくなる状況があるでしょう。他の誰かが行ったバージョンとの差分を得たり、自分がどのような変更を行ったか見るといったことです。

Subversion のようなバージョン管理システムを使用して作業を行っているとき、作業コピーを更新するときに競合が発生することがあります。これは、作業コピーに変更を加えた部分と同じ部分に対して、他の誰かが別の変更を加えたことによって発生します。競合は手作業で解決しなければなりません。TortoiseMerge はこの作業を助けてくれます。

競合時のファイルの関係を、以下の図に示します。

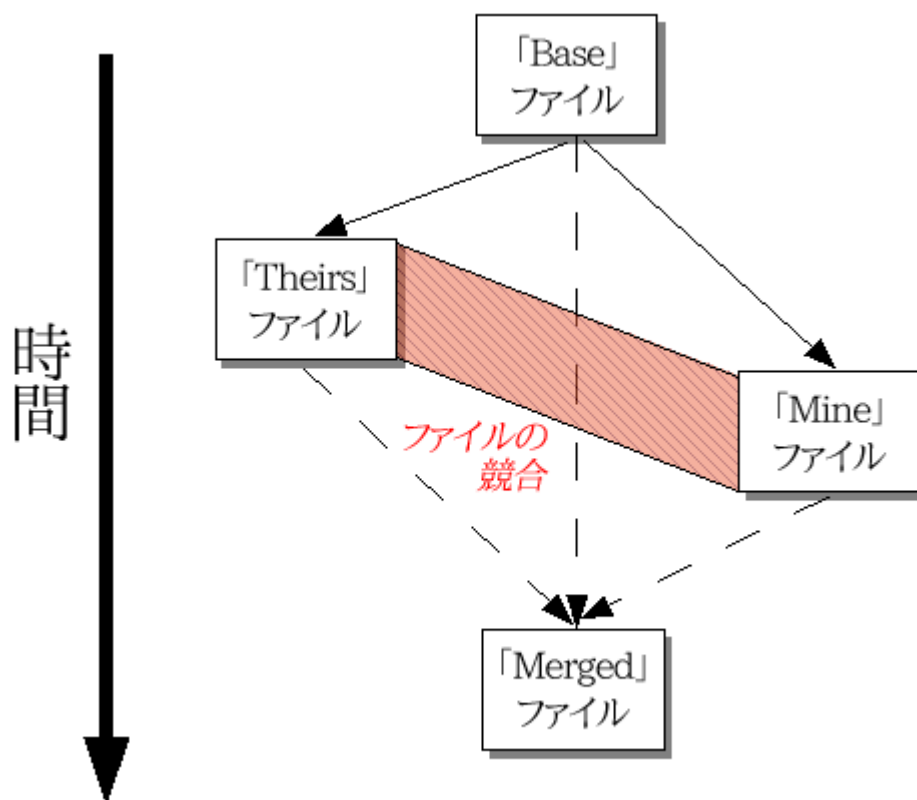


図1.1 ファイルの競合

1.2. TortoiseMergeの歴史

TortoiseSVN [<https://tortoisesvn.net>] の作業をしているとき、プロジェクトの拡張やバグの修正のためのパッチを送ってもらったときにはいつも、そのパッチを適用するのに大量の作業を行わなければなりません。また多くの場合、既にファイルを更新してしまっていたために、そのパッチは時代遅れになっており、コマンドラインパッチツール <http://unxutils.sourceforge.net/> [<https://unxutils.sourceforge.net/>] では適用できませんでした。

また、TortoiseSVN がもともと TortoiseSVN → パッチの作成 を実装していなかった主な理由でもあるのですが、パッチの適用が簡単だったので、私たちはユーザーがファイル全体を送ってくれるのを歓迎していました。

長時間を費やしてパッチファイルを適用することができる GUI ツールをインターネットで探しましたが、そのようなツールは存在しないということが分かっただけでした。その後は省略しますが、そのようなツールを提供してくれる人がいないのであれば、自分たちで作らなければならないと思い立ちました。そうして TortoiseMerge が生まれました。

TortoiseSVN は既にすばらしい差分エンジンを内蔵した Subversion のライブラリを使用していたので、GNU diff の代わりにこのライブラリを使用するのは自然なことでした。

Subversion 1.7 でパッチファイルを適用する `svn patch` コマンドが追加されました。TortoiseMerge は Subversion と動作を一致させるため、Subversion のコマンドを利用します。

第2章 バージョン管理の基本概念

TortoiseMerge は次のような様々な用途に使用できます。

- ・ 2つのファイルの差分を表示し、変更点をマージしたり、不要な変更を削除したりすることができます。
- ・ ローカルな変更と、更新で取得した Subversion のリポジトリにある変更との競合の編集を行うことができます。
- ・ パッチを適用し、結果をレビューします。

2.1. 差分の表示やマージ

このモードでは、2つのファイルを比較できます。左画面のファイルはオリジナルファイル (Theirs とも呼ばれる) として扱われ、右画面のファイルは変更後のファイル (Mine とも呼ばれる) として扱います。

右画面にあるファイルに対しては、単なる行単位の変更に加え、次のような操作を行うことができます。

- ・ 変更された行を、左画面に表示されているテキストに戻す。
- ・ Mine の前に Theirs を配置したり、Theirs の前に Mine を配置したりして、両方のブロックを結合する。

また、右画面のファイルはテキストエディターのように編集することもできます。その行は鉛筆アイコンで印がつけられます。なお、前述の行単位やブロック単位の変更を行うつもりなら、編集する前に行っておく方がよいでしょう。編集してしまうと、TortoiseMerge が オリジナルファイルとの関連を追跡できなくなります。

2.2. 競合の編集

これは3方向マージと呼ばれ、3画面表示になります。しかし、実際には4つのファイルが関係しています。現在競合している2つのファイルの共通の祖先である、共通のベースファイルは表示されません。3つのファイルの関係は、以下の図で説明します。

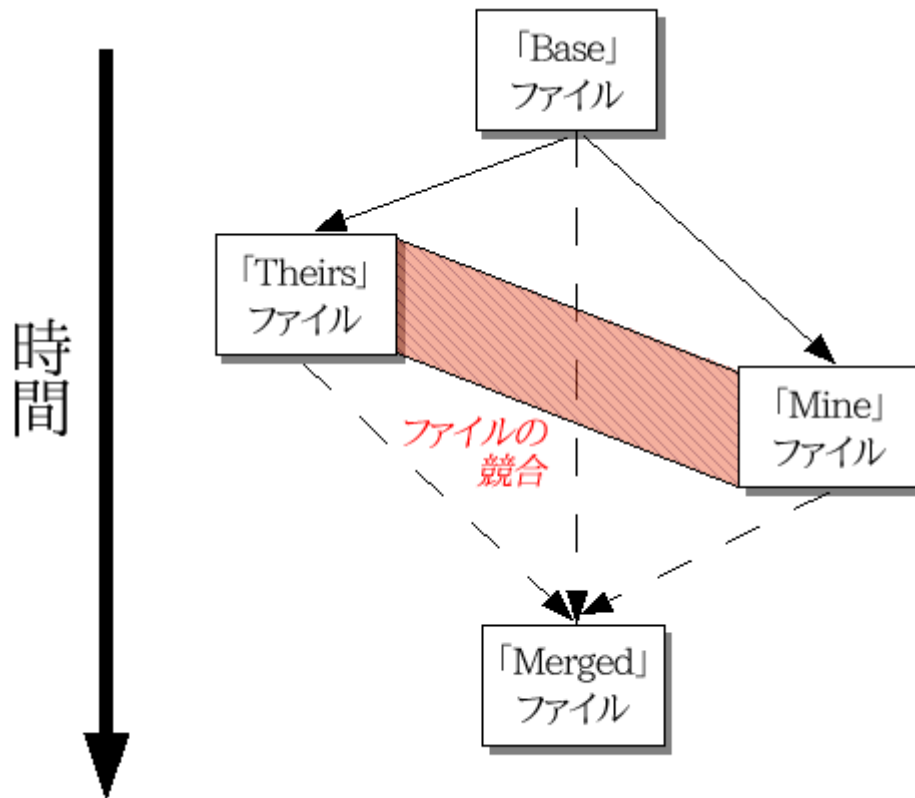


図2.1 ファイルの競合

ベースファイル (Base) は、自分と他人がそのファイルに対して変更を始めた、最も古いバージョンを表します。Mine は Base ファイルに対して自分が変更した結果を表します。また、Theirs は他人が変更した結果を表します。左画面には、ベースファイルに対する Theirs の変更が表示されます。また、右画面には、ベースファイルに対する Mine の変更を表示します。下画面には、競合を解決した結果を出力するファイルが表示されます。

競合の解決画面では、Mine や Theirs もしくは両方から、採用するブロックを選択できます。この場合、変更結果は下の画面に表示されます。

2.3. パッチの適用

パッチファイルは Subversion の Unified 差分ファイルで、変更を適用するファイルの一覧が含まれています。パッチは、あなたが変更点を確認してからコミットできるように、他の開発者によって作成されたかもしれません。また、TortoiseSVN が2つのフォルダーを比較する際に、内部的に作成したかもしれません。これは、リポジトリのリビジョンと作業コピーのフォルダーを比較したり、リポジトリ内の2つのリビジョンを比較したり、2つの異なるリポジトリパスを比較したりすると発生します。

いずれの場合も、パッチに含まれるファイルの一覧が、小さいウィンドウで表示されます。その中のファイルをダブルクリックすると、関連するファイルが取得され、変更が適用されます。左画面にはパッチ適用前の、右画面にはパッチ適用後のファイルの内容が表示されます。

比較・編集モードと同様に、右画面のファイルを編集することができます。

第3章 TortoiseMergeの使い方

3.1. 表示モード

TortoiseMerge は、1画面表示、2画面表示、3画面表示の3つの表示モードを持っています。1・2画面表示は変更を表示するために、3画面表示は競合を解決するために使用します。

3.1.1. 表示・マージ

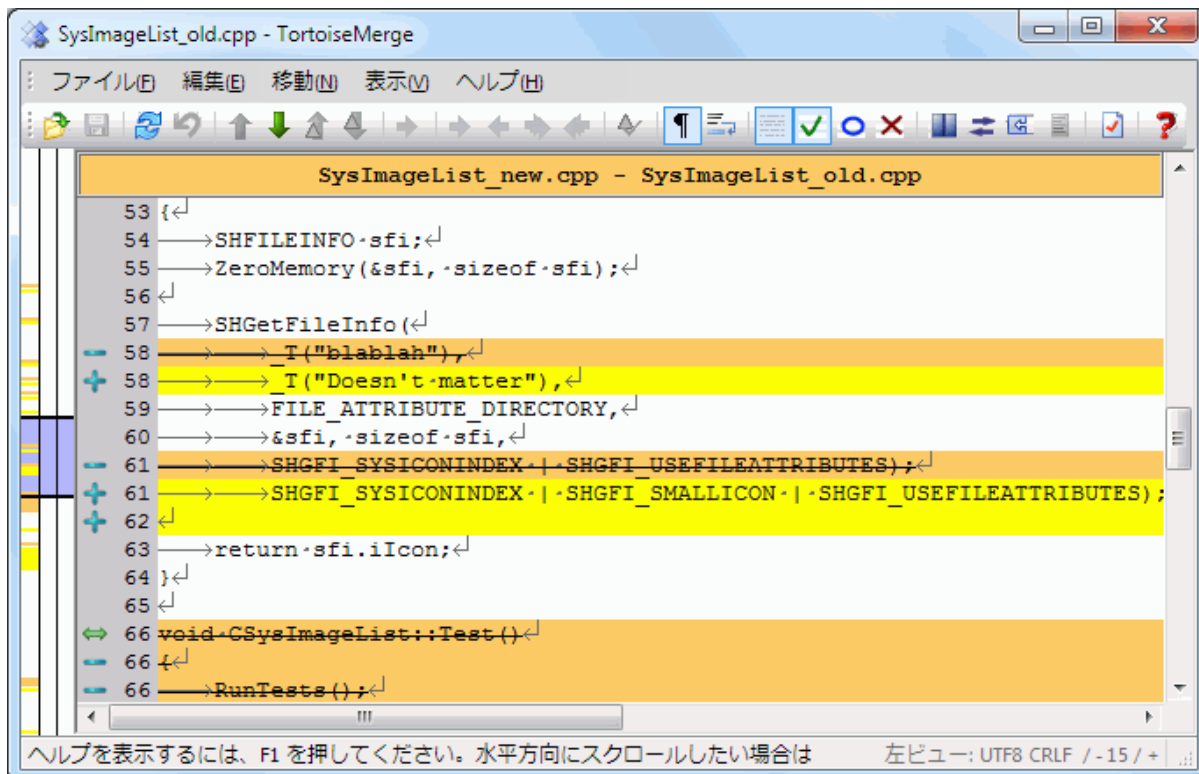


図3.1 1画面表示

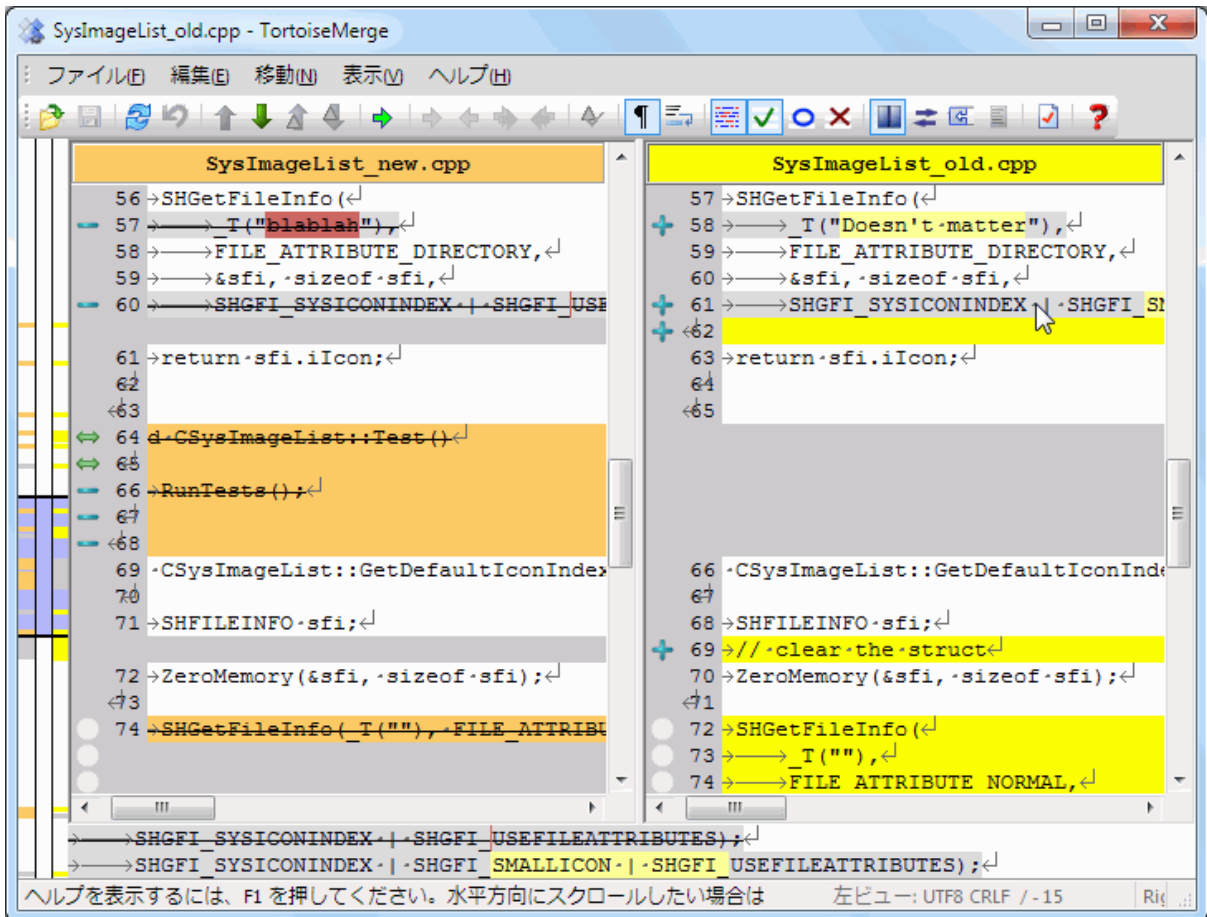


図3.2 2画面表示

2画面表示には、1画面表示にはない次のような特徴があります。

- ・ 変更のあった行内の変更点が、違う色で表示されます。文字列が追加された部分は明るい色になりますが、もちろんここで使用する色は設定できます。削除された部分は暗い茶色表示され、文字列に取り消し線が入ります。どのように動作するかは、上のスクリーンショットで確認してください。
- ・ コードを再編成すると、ホワイトスペース（空白、タブ、改行）を大量に変更しても、コードには実質的に変更がないことがあります。例えば、非常に長い行を数行に分割したり、短い複数の行をひとつにまとめたりした場合です。

このような変更は、画面の左端に白い円が表示されます。このような白い円がある場合には、実質的にはその部分のコードに変更がなく、変更箇所のチェックを行う必要がないことがすぐにわかります。

- ・ 2画面表示では、右画面のファイルを手作業で編集できます。変更した場所には鉛筆マークが表示されます。左画面がアクティブな時に**編集を有効化**を押した場合のみ、左画面も手作業で編集できます。

3つのファイルを比較・マージする場合、TortoiseMerge は3画面表示で差分を表示します。ファイルの競合を解決する必要があるときにも便利です。

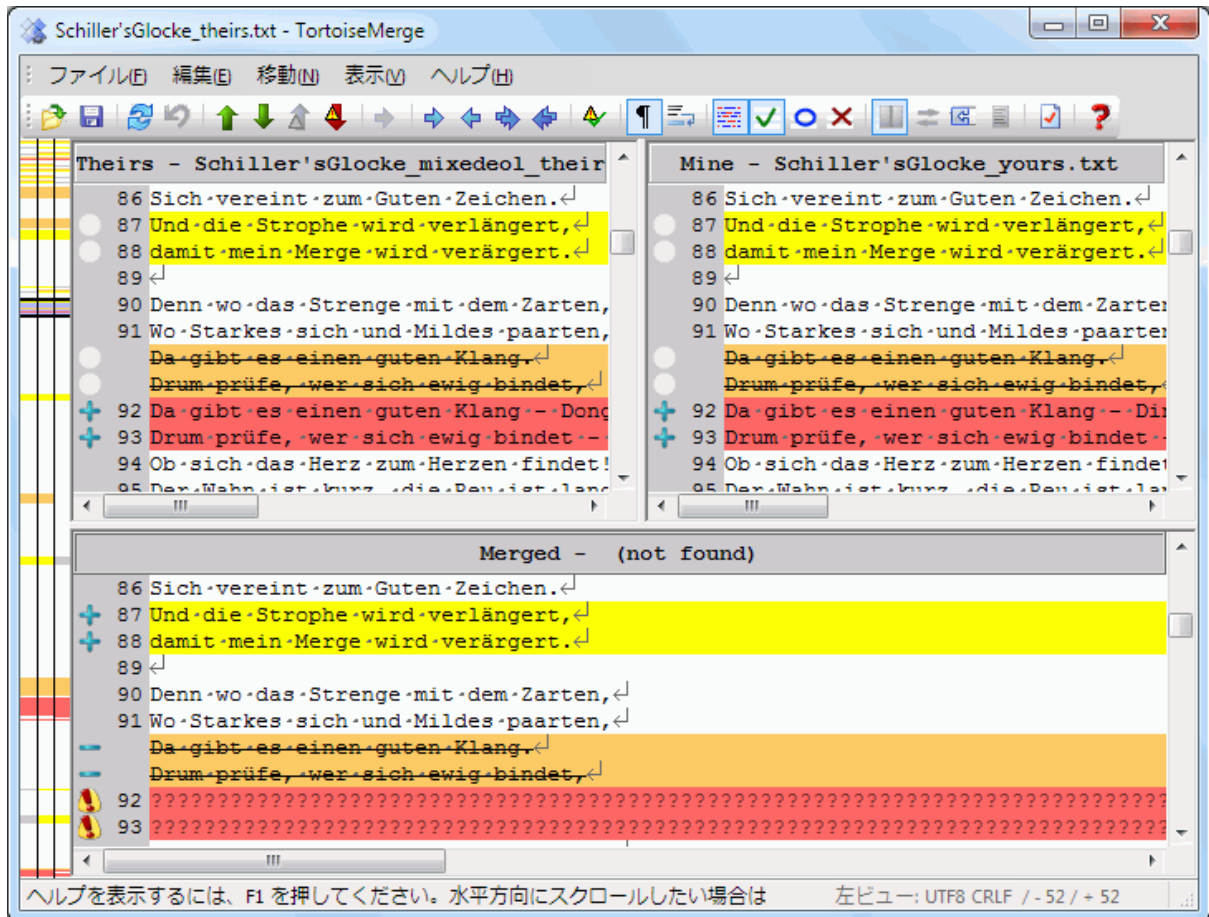


図3.3 3画面表示

左画面には Their ファイルと Base ファイルの差異が表示され、右画面には Mine ファイルと Base ファイルの差異が表示されます。下画面には、Base、Mine、Theirs をマージした結果が発生する競合と共に表示されます。

画面のタイトル部分にマウスを置くと、どちらの画面でも差分を取っているファイル名がツールチップに表示されます。

左画面と右画面は、アクティブな時に編集を有効化を押した場合のみ、手作業で編集できます。

3.1.2. パッチの適用

TortoiseMerge がパッチファイルを解釈すると、そのパッチファイルで変更されるファイルを表示する、小さなウィンドウが表示されます。

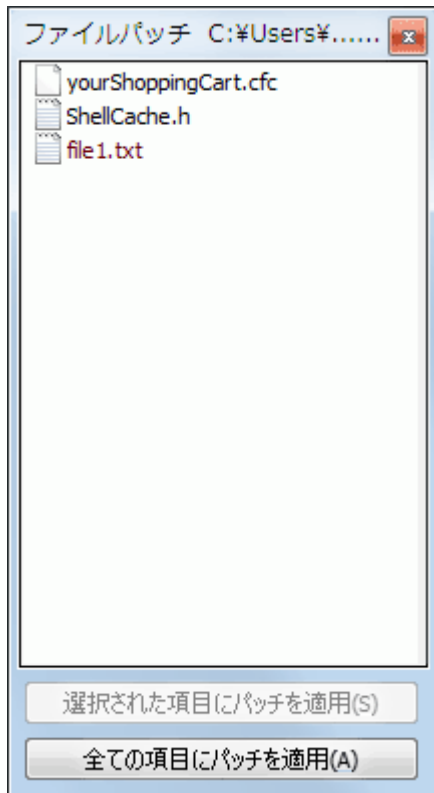


図3.4 パッチを適用するファイル一覧

ファイル名が黒で表示されている場合、問題なくパッチを当てることができます。これはパッチがそのファイルに対して古くなっているわけではないことを表します。しかし、ファイル名が赤で表示されている場合、既にファイルが変更されているため、直接パッチを当てることができません。

パッチファイルウィンドウにはコンテキストメニューがあり、現在のファイルにパッチを適用した結果をプレビュー（保存せず適用）したり、選択したファイルのみ適用して保存したり、一覧にあるファイルすべてに適用・保存を行うといった操作ができます。ダブルクリックするとプレビューします。

3.2. ボタンやその他のコントロール

設定によっては、多くのボタンが表示されたツールバーやリボンが表示されます。各ボタン上でマウスポインターを停止すると、機能を説明するツールチップが表示されます。

ウィンドウの左側にはロケータバーがあります。これによって、ファイル中のどこに変更があるのかを概観することができます。ロケータバーは3列あります。左の列は左画面を、右の列は右画面を、中央の列は（表示されている場合）下画面を表しています。1画面表示時は左の列のみ表示されます。ロケータバーは、各画面を同時にスクロールさせるスクロールバーとしても使用できます。

単語をダブルクリックすると、文書中の単語の出現位置が、メイン画面とロケータバーの両方で強調表示されます。強調表示を解除するには、もう一度単語をダブルクリックします。

左マージンをクリックしたり、行をトリプルクリックしたりすると、行全体を選択することができます。

ウィンドウの下部にあるのはステータスバーです。ここには Theirs と Mine の間で追加された行数、削除された行数、未解決の衝突の数が表示されます。

ステータスバーにはどのようにファイルを取り扱うかを指定するコンボボックスがあります：

エンコーディング

エンコーディングでビューの文字列をどのように読込・保存・表示するかを設定します。英語では普通 ASCII です(つまり OS のローカルエンコーディング)。UTF8、UTF16LE、UTF16BE、UTF32LE、UTF32BE (それぞれバイトオーダーマークありなし)に変更することもできます。

改行コード

Windows では一般に改行コードには CRLF が使われますが、任意の改行コードを指定することもできます。改行コードを変更すると、たとえ読み込んだファイル内の改行コードが揃っていても、ファイル内のすべての改行コードが変更されることに注意してください。

タブ

コンボボックスの一番上の項目は TAB キーを押した際にタブとスペースのどちらを挿入するかを指定します。スマートタブ文字 を有効にすると、タブとスペースのどちらが適切かを自動的に判定します。

タブ幅はタブキーを押した際にスペースをいくつ挿入するか、もしくはタブがあった際にタブの次の単語をどれだけインデントするかを指定します。

3.3. 行状態アイコン

変更された行は、どのような変更が行われたかがアイコンで表示されます。



追加された行です。



削除された行です。



変更を取りやめ、元の内容に戻された行です。



この行は空白しか変更されていません。マークが連続している箇所では、段落の折り返し位置が変更されて、単語が隣の行に移動しているかもしれません。



TortoiseMerge をテキストエディターとして使用して、手作業で編集された行です。



競合のある行です。



競合がありますが、空白や改行の設定により効果が見えない行です。



他の場所との間で行単位の移動が検出された行です。

3.4. 競合のマージと編集

TortoiseMerge はファイルの差分を 表示 するだけでなく、競合を解決したり変更を適用したりすることができます。

2画面表示の場合、右画面 (Mine) のみ編集できます。左のファイル (Theirs) に変更を加えるには、変更する行で右クリックし、**コンテキストメニュー** → 「theirs」のテキストブロックを使用を選択してください。そうすれば、左のファイルからの変更が右のファイルに加えられます。

時には、両方のテキストブロックが必要になる場合があります。その場合は、**コンテキストメニュー** → **両方のテキストブロックを使用 (こちらを前にする)** や **コンテキストメニュー** → **両方のテキストブロックを使用 (こちらを後にする)** を使用してください。

また、テキストエディターのように出力ファイルを編集することもできます。その行は鉛筆アイコンで印がつけられます。なお、前述の行/ブロックベースの変更を行うつもりなら、編集する前に行っておいた方がよいでしょう。編集してしまうと、TortoiseMerge が オリジナルファイルとの関連を追跡できなくなります。

3画面表示（ マージビュー とも呼ばれる）の場合、下画面にあるファイル（マージ後）のみ編集できます。2画面表示と同様に、競合した行で 右クリック し、 **コンテキストメニュー** → 「theirs」のテキストブロックを使用 と **コンテキストメニュー** → 「mine」のテキストブロックを使用 のどちらかを選択してください。さらに両方のブロックを使用したい場合は、 **コンテキストメニュー** → 「theirs」の前に「mine」のテキストブロックを使用 と **コンテキストメニュー** → 「mine」の前に「theirs」のテキストブロックを使用 のどちらかを選択してください。選択したコマンドの結果が、マージ後ファイルに反映されます。

時には、TortoiseMerge では競合していないのに、Subversion で競合が発生したと印がつけられることがあります。これは、選択した空白の扱いによるものかもしれません。行末や空白の変更を無視するようにした場合、その行は 競合無視 アイコンを用いて印がつけられます。競合を解決するために、どのバージョンを採用するか選択する必要があります。



重要

同じファイルで再度 TortoiseMerge を使用すると、TortoiseMerge で変更したか手作業で編集したかにかかわらず、作業コピーに行った変更が 取り消され、競合の編集を行い始めた状態のファイルになるので注意してください。

3.5. ファイルを開く

コマンドラインスイッチを指定せずに TortoiseMerge を起動した場合は、 **ファイル** → **開く** を使用して手作業でファイルを開いてください。

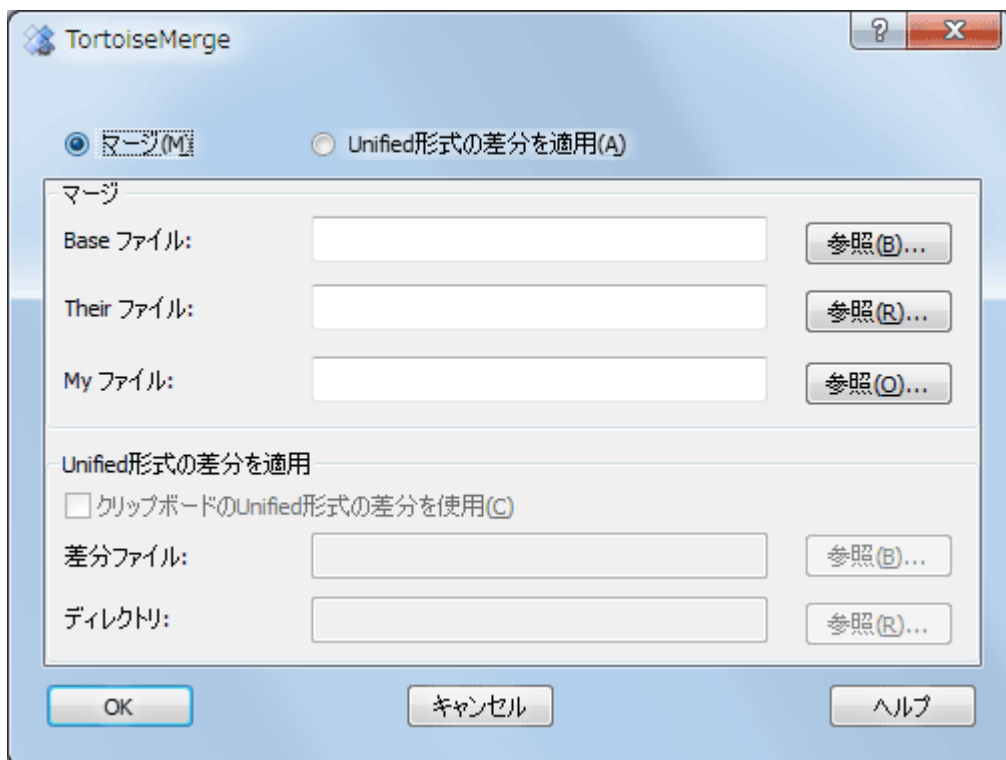


図3.5 開くダイアログ

まず始めにあなたがすることは、ファイルの比較・マージをしたいのか、パッチを適用したいのかを決めることです。その選択により、該当するエディットボックスやブラウザボタンが有効になります。

3.5.1. 表示・マージ

ファイルの比較・マージを行う際には、Base, Mine, Theirs の3つのうち、少なくとも2つのパスを設定しなければなりません。2つのファイルだけを指定すると、その2つのファイルの差分が2画面か1画面のどちらかで表示されます。

3つのファイルをマージする場合、3画面表示で差分が表示されます。この表示は通常、ファイルの競合を解決する必要があるときに使用されます。この場合、出力ファイルには名前が付かないので、結果を保存するには **ファイル→名前を付けて保存...** を使う必要があります。

3.5.2. パッチの適用

パッチファイルを適用する場合、パッチファイルそのもののパスと、パッチファイルを適用するフォルダーのパスの両方を、設定しなければなりません。

3.6. 設定

3.6.1. メイン設定ページ

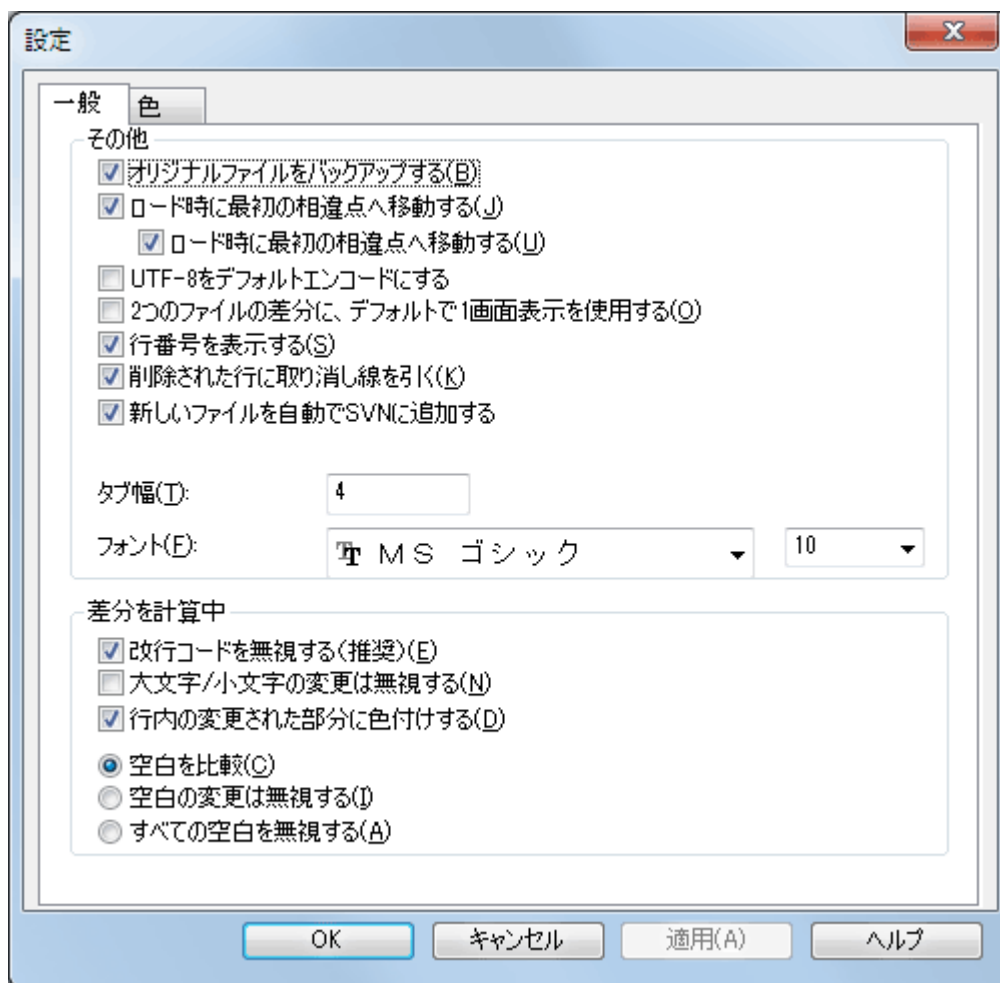


図3.6 設定の一般ページ

ほとんどのオプションは読めばわかりますが、いくつか要点を示してしておきます。

オリジナルファイルをバックアップする 変更したバージョンを保存する前に、作業コピーにあるファイルが filename.bak に名前が変更されます。

UTF-8エンコーディングをデフォルトにする をセットすると、ANSI のファイルでも UTF-8 エンコードとして開くようになり、編集すると UTF-8 エンコードとして保存されるようになります。

行内差分を取る行の最大長 TortoiseMerge で非常に長い行の行内差分を表示すると、遅くなってしまいます。そのため、3000文字未満の行のみ行内差分が表示されるようになっています。ここでその長さを変えることができます。

改行コードを無視する 改行コードの違いがないかのように違いを隠します。

大文字/小文字の変更は無視する テキストファイル中、大文字/小文字しか変わらない変更を隠します。Visual Basic のような、警告なしで変数の大文字/小文字を変更してしまうようなアプリでは便利です。

3.6.2. 色設定ページ

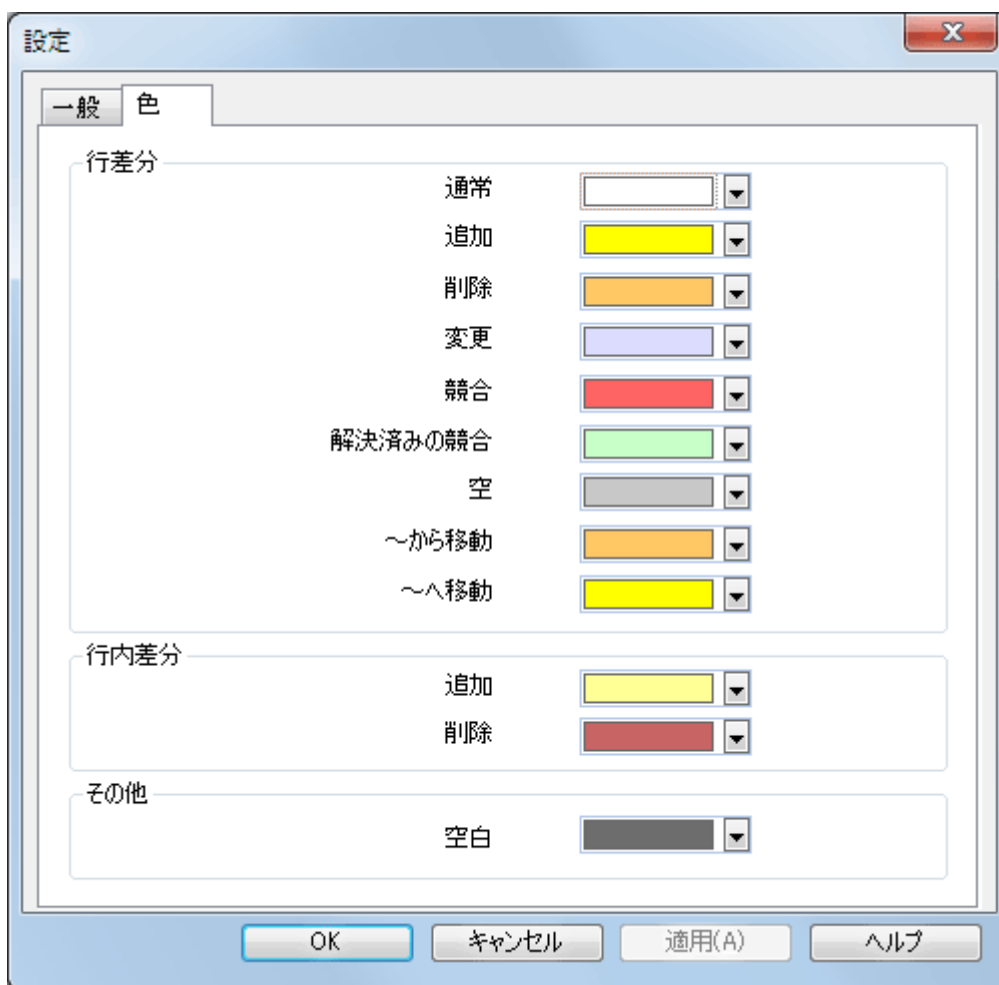


図3.7 設定の色ページ

このダイアログで、変更がある行がハイライト表示される背景色を選択できます。

通常

変更されていないか、変更が隠されているすべての行。

追加

追加された行。

削除

削除された行。

変更

比較的变化が少なく、行内差分の形で表示される行。この色は行内の 変更されていない 箇所に使用されます。変更された箇所は、後述する行内の追加・削除色で表示されます。 **行内の変更された部分に色付けする** が無効の場合、この色は使用されず、変更された行は常に置換された行として表示されます。

競合

両方のファイルで同じ行が変更されていた箇所。

解決済みの競合

両方のファイルで同じ行が変更され、どちらを使用すべきか既に選択した箇所。

空

反対の画面には行が追加されているが、こちらの画面には対応する行が存在しない箇所。

行内追加テキスト

行差分で表示時、追加されたテキストがこの色で表示されます。

行内削除テキスト

行差分で表示時、削除されたテキストがこの色で表示されます。

その他の空白

空白を表す文字を、通常のテキストとは別の色で表示します。

付録A キーボードのショートカット

キーボードのショートカットとコマンドの一覧です。

A.1. キーボードのショートカット

Ctrl+Q、Ctrl+W、Escape
プログラムを終了する

Ctrl+C
選択したテキストをクリップボードへコピーする

Ctrl+X、Shift+Del
選択したテキストを切り取ってクリップボードに入れる

Ctrl+V、Shift+Insert
クリップボードから選択したテキストを貼り付ける

Ctrl+Z、Alt+Backspace
最後の変更を元に戻す

Ctrl+F
文字列の検索や置換のために検索ダイアログを開く

Ctrl+O
差分・マージ用にファイルを開く

Ctrl+S
変更を保存

Ctrl+Shift+S
名前を付けて保存 ...

F7
次の相違点へ移動

Shift+F7
前の相違点へ移動

F8
次の競合点へ移動

Shift+F8
前の競合点へ移動

Ctrl+D
1画面表示と2画面表示を切り替え

Ctrl+R
ファイルを再読込し、すべての変更の取り消し

Ctrl+T
空白の表示・非表示を切り替え

Ctrl+L
未変更箇所を折りたたむかどうかを切り替え

Ctrl+P
行の折り返しの有無を切り替え

Ctrl+G

指定した行に移動

Ctrl+A

文字列全体を選択

Ctrl+P

行の折り返しの有無を切り替え

Ctrl+U

画面表示の切り替え

Ctrl+マウスホイール

左右の表示をスクロールする

Ctrl+Tab

左画面／右画面／下画面の切り替え

付録B TortoiseMergeの自動化

TortoiseMerge は、コマンドラインパラメーターを指定すると、ファイルを選択する開くダイアログを表示せずに起動できます。TortoiseMerge を他のアプリケーションから使用する際に便利です。

B.1. TortoiseMergeのコマンドラインスイッチ

ほとんどのスイッチには、パスやその他の文字列などの追加情報が必要です。この場合、スイッチの後に「:」に続けて文字列やパスを指定してください。例えば、次のように指定します。

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

コマンド	説明
/?	主要なコマンドラインスイッチをダイアログで表示します。
/help	? と同様です。
/base	3方向差分で使用する base ファイルを指定します。画面には表示されませんが、これが共通の祖先になります。2方向差分ならば左ファイルになります。
/basename	base ファイルの名前です。ファイルパスの代わりに、画面のタイトルに表示されます。3方向差分では、画面のタイトル部のツールチップに表示されます。
/basereflectedname	使用する editorconfig テンプレートの名前です。
/theirs	3方向差分で使用される theirs ファイルを指定します。左画面に表示されます。
/theirsname	theirs ファイルの名前です。ファイルパスの代わりに、画面のタイトルに表示されます。
/theirsreflectedname	使用する editorconfig テンプレートの名前です。
/mine	3方向差分で使用される mine ファイルを指定します。右画面に表示されます。2方向差分では右ファイルになります。
/minename	mine ファイルの名前です。ファイルパスの代わりに、画面のタイトルに表示されます。
/minereflectedname	使用する editorconfig テンプレートの名前です。
/merged	マージ済みファイルの出力先を指定します。これはマージしたりや競合を解決したりした後のファイルが保存されるファイルパスです。 3方向差分でこれが指定されなかった場合、TortoiseMergeは結果をどこに保存するかをユーザーに尋ねます。 2方向差分でこれが指定されなかった場合、TortoiseMergeは自動的に右ビューに指定されたファイルを保存先として使用します。
/mergedname	merged ファイルの名前です。ファイルパスの代わりに、画面のタイトルに表示されます。
/mergedreflectedname	使用する editorconfig テンプレートの名前です。
/patchpath	適用するパッチファイルのパスです。このパスを設定しない場合、TortoiseMerge はパッチファイルのあるパスと一致するパスから探そうとしますが、これには 非常に長い時間 がかかります。
/saverequired	これを指定すると、TortoiseMerge はファイルを編集しているかどうかにかかわらず、ファイルの上書き時に確認を行います。

コマンド	説明
/saverequiredonconflict	これを指定すると、TortoiseMerge はファイルを編集しているかどうかにかかわらず、衝突が見つかった際にはファイルの上書き時に確認を行います。
/patchoriginal	パッチを当てるオリジナルファイルの名前です。画面のタイトルに使われます。
/patchpatched	パッチを当てたファイルの名前です。画面のタイトルに使われます。
/diff	ディレクトリに適用するパッチ・差分ファイルへのパスです。
/oneway	ユーザーの設定に関わらず、TortoiseMerge を強制的に1画面表示で起動します。
/reversedpatch	差分を取るために指定した2ファイルの左右の画面を入れ替えます。
/createunifieddiff	/origfile:"<変更前ファイルへのパス>" と /modifiedfile:"<変更後ファイルへのパス>" で指定したファイルから、Unified 差分ファイル (パッチファイル) を作成します。ターゲットパスは /outfile:"<結果のパッチファイルへのパス>" で指定します。 /outfile を指定しない場合、パッチファイルを保存する場所を指定するためにファイルの保存ダイアログが表示されます。 /createunifieddiff を指定した場合、それ以外のパラメーターはすべて無視されることに注意してください。
/line	ファイルを読み込んだ後にジャンプする行番号を指定します。
/readonly	ファイルが編集されるのを防止します。つまり、TortoiseMerge の編集機能が無効になります。
/nosvnresolve	TortoiseMerge でファイルを保存した後、SVN 上で解決済みにするかどうかを尋ねることを抑制します。

表B.1 有効なコマンドラインオプションの一覧

他の差分プログラムとの互換のため、コマンドラインにはファイル名だけを渡すこともできるようになっています。略式のコマンドラインは、

```
TortoiseMerge <baseファイルパス> <mineファイルパス> [ <theirsファイルパス> ]
```

という書式です。2つのファイルを与えると、互いに比較します。3つのファイルを与えると、最初のファイルをベースファイルとして、他の2つと比較する3方向差分になります。

用語集

BASE リビジョン	作業コピー にあるファイルやフォルダーの現在のベースリビジョンで、最後にチェックアウト、更新、コミットを実行したときの、ファイルやフォルダーのリビジョンです。BASE リビジョンは、通常は HEAD リビジョンと同じではありません。
FSFS	Subversion が持つリポジトリ用のファイルシステムバックエンドです。ネットワークフォルダー上で使用することができます。1.2以降のリポジトリのデフォルトです。
GPO	グループポリシーオブジェクト。
HEAD リビジョン	リポジトリにあるファイルやフォルダーの最新リビジョンです。
SVN	Subversion のよく使われる省略表現。 「svnserve」リポジトリサーバーで使われる、Subversion カスタムプロトコルの名前。
インポート	単一のリビジョンで、フォルダー階層の内容をリポジトリにインポートする Subversion のコマンドです。
エクスポート	このコマンドは、作業コピーのようにバージョン管理下のフォルダーをコピーしますが、.svn は作成しません。
クリーンアップ	Subversion book から引用します。「作業コピーの再帰的なクリーンアップ（ロックの解除、未完操作の回復）を行います。作業コピーがロックされています というエラーが出続ける場合、このコマンドを実行し、古くなったロックを削除し、作業コピーをまた使えるようにします。」ここで言う ロック とは、ファイルシステムのロックを指しており、リポジトリのロックではないことに注意してください。
コピー	Subversion リポジトリでは、単一ファイルやツリー全体のコピーを作成できます。これは、領域を消費しないように、オリジナルへのリンクに少し似ている「簡易コピー」で実装されています。コピーの作成ではコピー内に履歴を保存します。そのためコピーされる前についても追跡できます。
コミット	ローカルの作業コピーの変更点をリポジトリに渡し、リポジトリのリビジョンを新しく作成するのに使用する Subversion のコマンドです。
チェックアウト	空のディレクトリに、リポジトリからバージョン管理下のファイルをダウンロードし、ローカルの作業コピーを作成する Subversion のコマンドです。
パッチ	作業コピーの変更がテキストファイルのみである場合、Subversion の Diff コマンドを使用すると、変更内容を Unified 差分ファイルとして単一ファイルに出力することができます。この種のファイルはよく「パッチ」と呼ばれており、他の誰か（やメーリングリスト）にメールで送ったり、他の作業コピーに適用したりすることができます。コミットのアクセス権を持たない人は、変更を作成し、パッチファイルをコミット権限を持つ人に送ることができます。また、変更に自信がない場合、他の人にパッチを送ってレビューしてもらうこともできます。
ブランチ	バージョン管理システムでよく使用される用語で、ある時点で開発が2つの独立したパスに分岐したことをと表します。メインの開発

ラインからブランチを作成すれば、メインラインを不安定にせずに新機能の開発を行うことができます。また、今後バグフィックスのみを行うための安定版リリースのブランチを作成すれば、新機能の開発は不安定なトランクで行うことができます。Subversion のブランチは、「簡易コピー」として実装されています。

プロパティ	ディレクトリやファイルのバージョン管理をするのに加えて、Subversion はバージョン管理されたメタデータを追加できます。これは、バージョン管理下のディレクトリやファイルごとに「プロパティ」として参照されます。プロパティには、レジストリキーと同じように、それぞれ名前と値があります。Subversion には、 <code>svn:eol-style</code> のような内部で使用する特殊なプロパティがいくつかあります。TortoiseSVN にも <code>tsvn:logminsize</code> のような特殊なプロパティがあります。任意の名前と値を持つプロパティの追加もできます。
マージ	リポジトリに追加された変更を、ローカルで行った変更を壊さないように、作業コピーに追加するプロセスです。時には自動的に調整できず、作業コピーが競合と呼ばれる状態になります。 作業コピーを更新する際に、自動的にマージが行われます。また、TortoiseSVN のマージコマンドを使用して、別のブランチにある変更を指定してマージすることもできます。
リビジョン	変更セットをコミットするたびに、新しい「リビジョン」がリポジトリに作成されます。各リビジョンは、履歴の決まった時点のリポジトリツリーの状態を表します。過去にさかのぼる場合は、リビジョン N のような形でリポジトリを調べられます。 言い換えるとリビジョンは、リビジョンが作成された時に行われた変更を示しています。
リビジョンプロパティ (revprop)	ファイルがプロパティを持てるように、リポジトリの各リビジョンもプロパティを持つことができます。リビジョンが作成されるたびに、 <code>svn:date</code> <code>svn:author</code> <code>svn:log</code> といった特殊なリビジョンプロパティが自動的に作成され、それぞれコミット日時、コミットした人、ログメッセージを表しています。これらのプロパティは編集できますが、バージョン管理できません。そのため変更は永続的で元に戻せません。
リポジトリ	データを集中して格納し保守する場所。複数のデータベースやファイルをネットワーク上に分散して置くこともでき、ネットワークに出ずに直接アクセスできる場所に置くこともできます。
ログ	ファイルやフォルダーのリビジョンの歴史を表します。「履歴」とも呼びます。
ロック	バージョン管理下の項目のロックを取得すると、その作業コピーを除いて、ロックが解除されるまでリポジトリにコミット不可の印が付きます。
作業コピー	ローカルの「サンドボックス」で、バージョン管理ファイルに対して作業を行う場所です。また通常ローカルのハードディスクに記録されています。リポジトリからの「チェックアウト」で作業コピーを作成し、「コミット」で変更点をリポジトリに反映します。
再配置	サーバー上の異なるディレクトリに移動したり、ドメイン名が変更されたりして、リポジトリが移動した場合、作業コピーを「再配置」すれば、作業コピーのリポジトリのURLが新しい場所を指すようになります。

【注意】 このコマンドは、作業コピーが指している同じリポジトリの同じ場所を指し、そのリポジトリが移動されてしまったときのみで使用してください。その他の場合には、代わりに「切り替え」コマンドを使用する必要があります。

切り替え	ちょうど「特定リビジョンへ更新」が履歴上の別の位置へ、作業コピーの時間ウィンドウを変更するように、「切り替え」はリポジトリの別の位置へ、作業コピーの場所ウィンドウを変更します。トランクとブランチの違いが少ない時、それぞれの作業する際に役に立ちます。その2つの間で作業コピーを切り替えると、差異のあるファイルのみが転送されます。
削除	バージョン管理下のファイルを削除（してコミット）すると、リポジトリ内のそのコミットを行ったバージョン以降、その項目はもう存在しなくなります。しかしもちろん、それ以前のリポジトリのリビジョンには、まだ存在していますから、まだそれにアクセスできます。必要なら削除した項目をコピーし、履歴を含め完全に「復活」できます。
変更の取り消し	作業コピーを最後に更新したとき、Subversion はそれぞれのファイルの「当時の」コピーをローカルに保持しています。変更を行った後で、その変更を取り消したい場合は、「変更の取り消し」コマンドを使用して当時のコピーに戻せます。
履歴	ファイルやフォルダーのリビジョンの歴史を表します。「ログ」とも呼びます。
差分	「差分の表示」の略。どのような変更が行われたのか正確に見たいときに便利です。
更新	リポジトリから作業コピーへ最新の変更点を取得するコマンド。作業コピーの変更点に、他の人が行った変更をマージします。
注釈履歴	このコマンドはテキストファイル専用で、それぞれの行が最後に変更された時の、リポジトリのリビジョンと作者を注釈表示します。私たちの GUI の実装は TortoiseBlame と呼ばれ、リビジョン番号の上にマウスを置くと、コミット日時やログメッセージも表示します。
競合	リポジトリの変更がローカルにマージされる際、時には同じ行に変更がある場合があります。この場合、Subversion はどちらを使用するか自動的に決定できません。これを競合と呼びます。それ以降の変更をコミットする前には、ファイルを手作業で編集し競合を解決しなければなりません。
競合の解決	作業コピーのファイルが、マージ後に競合状態になったままになった場合、人間がエディター（または TortoiseMerge）で競合を整理しなければなりません。このプロセスは「競合の解決」と言われています。競合したファイルを解決済みにすると、コミットできるようになります。
追加	ファイルやディレクトリをリポジトリに追加する Subversion コマンドです。新しい項目は、コミットした際にリポジトリに追加されます。