

TortoiseMerge

diff/merge ツール for Windows

Version 1.6.16

Stefan Küng
Lübbe Onken
Simon Large

TortoiseMerge: diff/merge ツール for Windows: Version 1.6.16

: Stefan Küng, Lübbe Onken, 、 Simon Large
翻訳: 倉澤 望 (鍋太郎) (nabetaro @ caldron. jp)

発行 2011/01/21 21:21:17 (r20750)

目次

序	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	2
2. 基本コンセプト	3
2.1. 差分の表示・マージ	3
2.2. 競合の編集	3
2.3. パッチを適用	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. パッチを適用	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	10
3.4. 競合のマージ・編集	10
3.5. ファイルを開く	11
3.5.1. 表示・マージ	11
3.5.2. パッチを適用	12
3.6. 設定	12
3.6.1. メイン設定ページ	12
3.6.2. 色設定ページ	13
A. キーボードショートカット	15
A.1. キーボードショートカット	15
B. TortoiseMerge の自動化	16
B.1. TortoiseMerge コマンドラインスイッチ	16
用語集	18

目次

1.1.	ファイルの競合	1
2.1.	ファイルの競合	4
3.1.	1 ペイン表示	5
3.2.	2 ペイン表示	6
3.3.	3 ペイン画面	7
3.4.	パッチをあてるファイル一覧	8
3.5.	ツールバー	8
3.6.	開くダイアログ	11
3.7.	設定の一般ページ	12
3.8.	色の設定ページ	13

表目次

B. 1. 有効なコマンドライン一覧	16
--------------------------	----

序



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 は以下のどちらのタスクでも支援します。

- ・ 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。GP ライセンス (GPL) のもと開発されています。

TortoiseMerge はオープンソースプロジェクトです。これは、あなたがこのプログラムのソースコードに完全に読み取りアクセスできることを意味しています。このリンク <http://code.google.com/p/tortoisesvn/source/browse/> から、ソースを閲覧できます。ユーザ名とパスワードを要求されますので、ユーザ名を `guest`、パスワードを空のままにしてください。(私たちが現在作業している) 最新のバージョンは `/trunk/` に、既にリリースされたバージョンは `/tags/` に配置しています。

寄付!

TortoiseSVN と TortoiseMerge はフリーですが、パッチを送ったり、開発体制内で積極的に役割をこなしたりして、開発者をサポートできます。また、コンピュータの前で過ごす私たちを限りなく応援できます。

私たちは TortoiseSVN の作業中、音楽を聴くのが大好きです。また長時間このプロジェクトに関わっているために、音楽がたくさん必要です。そこで、私たちの好きな CD や DVD の欲しいものリストを用意しました。<http://tortoisesvn.tigris.org/donate.html> パッチや翻訳を送ってプロジェクトに貢献している人の一覧もご覧ください。

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

われらが貢献者達

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・オープンソースアプリケーションです。テキストファイルの差分を表示、変更点のマージ、パッチ とよく呼ばれる unified diff ファイルの検査・適用を行います。

お好み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。

バージョン管理システム (例: Subversion) を利用して作業を行っているとき、作業コピーの更新時に競合することがあります。これは作業を行ったファイルのある部分に対して、他の誰かが同じ部分を変更したことによって起こります。この競合は手動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。

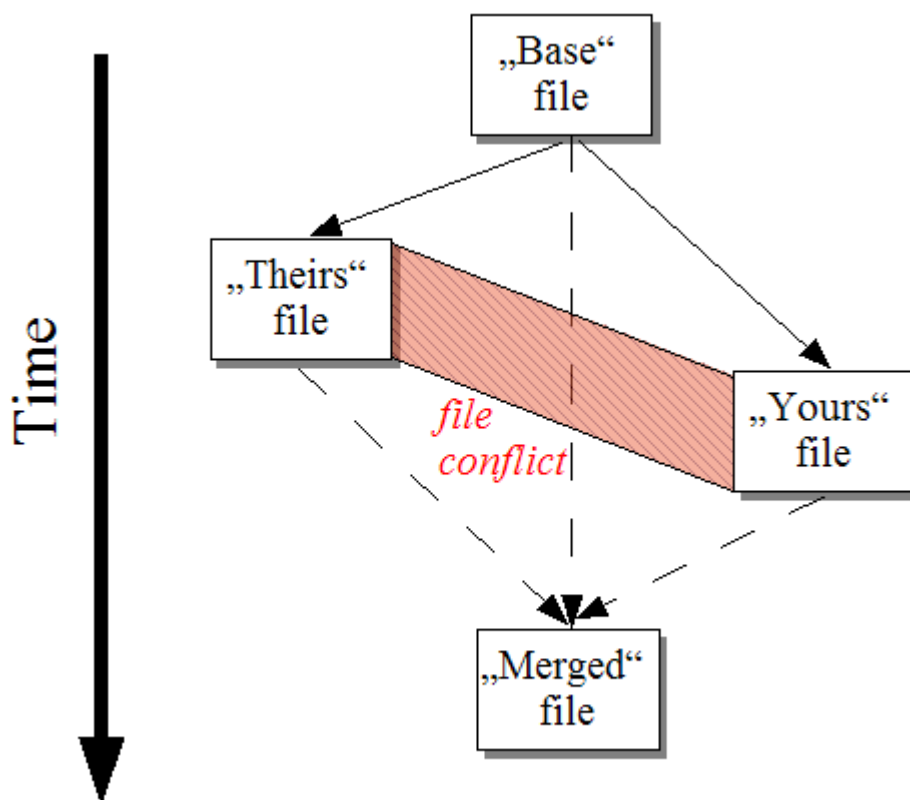


図 1.1. ファイルの競合

古くなってしまったパッチファイルを適用しようとしたときにも、競合は発生します。この場合 TortoiseMerge ではなく、現在のバージョンに対するパッチファイルがないか人に訊いたり、パッチファイルを手で適用する必要があるでしょう。

ですが、TortoiseMerge を使うとこういことがもう必要ありません。少なくとも Subversion のようなバージョン管理システムを使用する限りにおいては、TortoiseMerge はパッチを作成したりバージョン・バージョンを解析し、ソース管理シス

テムからその特定のバージョンを自動的に取得します。その後パッチを適用し、自分の行った変更と他の人の行った変更の両方を確認できるようにします。これにより、変更点を保持するかどうか決めることができます。

1.2. TortoiseMerge の歴史

TortoiseSVN [<http://tortoisesvn.tigris.org>] の作業をしていると、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うことになっていました。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/> 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なのですが、パッチの適用が簡単だったので、私たちはユーザがファイル全体を送ってくれるのを歓迎していました。

パッチファイルを適用する GUI ツールをインターネットで探すのに長時間費やしましたが、そのようなツールがありませんでした。はしょって言うと、そんなツールがないのなら自分たちで作ってしまおうということです。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 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。

第2章 基本コンセプト

TortoiseMerge は以下の異なる用途に使用できます。

- ・ 2 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えます。
- ・ パッチファイルの適用ができます。これが TortoiseMerge を作成した理由だったりします :)

2.1. 差分の表示・マージ

このモードでは、2 つのファイルを比較できます。左ペインのファイルをオリジナルファイル（時に *Theirs* として参照）として扱い、右ペインのファイル（*Mine* として参照）を変更したファイルとして扱います。

右ペインにあるファイルに対して、以下のことを含む、単純な行ベースの変更を行うことができます。

- ・ 更新行を、左ペインに表示したテキストに戻します。
- ・ 両方のブロックを結合したり、*Mine* の前に *Theirs* や、*Theirs* の前に *Mine* といったことができます。

また、右ペインでテキストエディタのように編集もできます。その行は鉛筆アイコンで印がつけられます。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しなくなります。

2.2. 競合の編集

これは 3 ペインで表示する、3 方向マージで行います。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 ファイルの共通の祖先である、共通ベースファイルは表示しません。3 ファイルの関係は、以下の図で説明します。

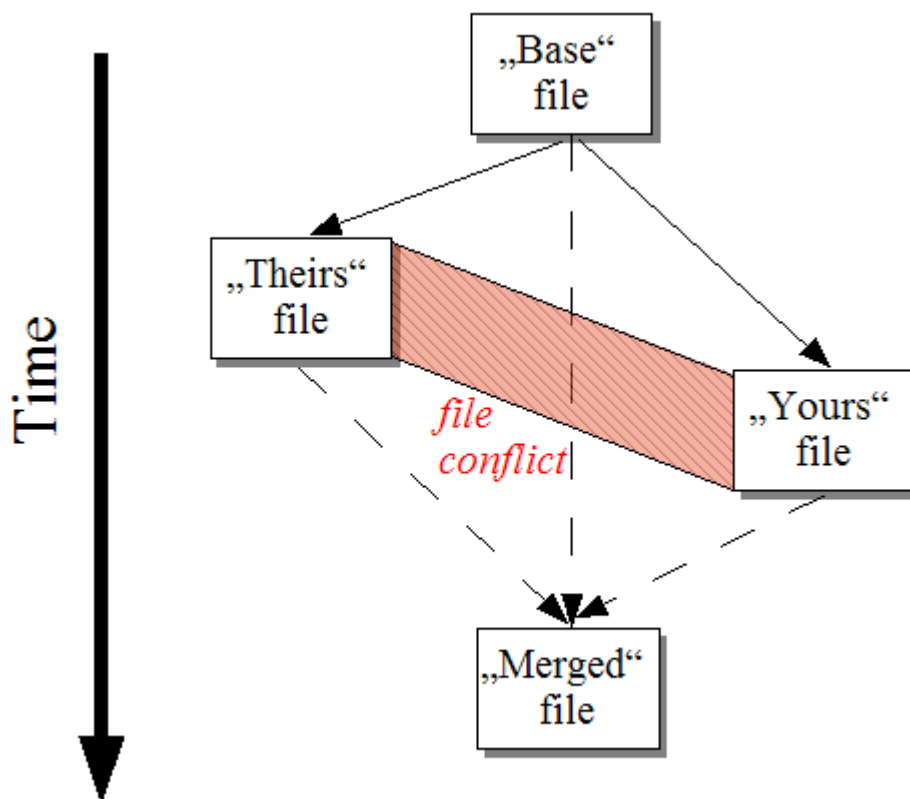


図 2.1. ファイルの競合

ベースファイルは You と They が変更を始めた、ファイルの最も古いバージョンを表します。Mine はベースファイルに対して you が行ったすべての変更があるファイルを表します。また、Theirs は 他人 が行ったすべての変更があるファイルを表します。左ペインはベースファイルに対する Theirs の変更を表示します。また、右ペインはベースファイルに対する Mine の変更を表示します。下ペインには、競合を解決しようとしている出力ファイルです。

競合解決ビューでは、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。しかし、この場合、変更は下部ペインに表示します。

2.3. パッチを適用

パッチファイルは、Subversion の unified diff 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、リポジトリリビジョンと作業コピーフォルダを比較したり、2 つの異なるリポジトリ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、TortoiseMerge はパッチに含まれるファイルを表示する、小さなウィンドウを表示します。その中のファイルをダブルクリックすると、関連ファイルを取得し、変更を適用します。左ペインにはオリジナルファイルの内容を表示し、右ペインにはパッチを適用した結果を表示します。

比較・編集モードで、右ペインに対して編集を行えます。

第3章 TortoiseMerge の使い方

3.1. 表示モード

TortoiseMerge は、以下の 3 つのビューモードを持っています。1 ペインビュー、2 ペインビュー、3 ペインビューです。1 ペイン・2 ペインビューは変更の確認に、3 ペインビューは、競合の解決に使用します。

3.1.1. 表示・マージ

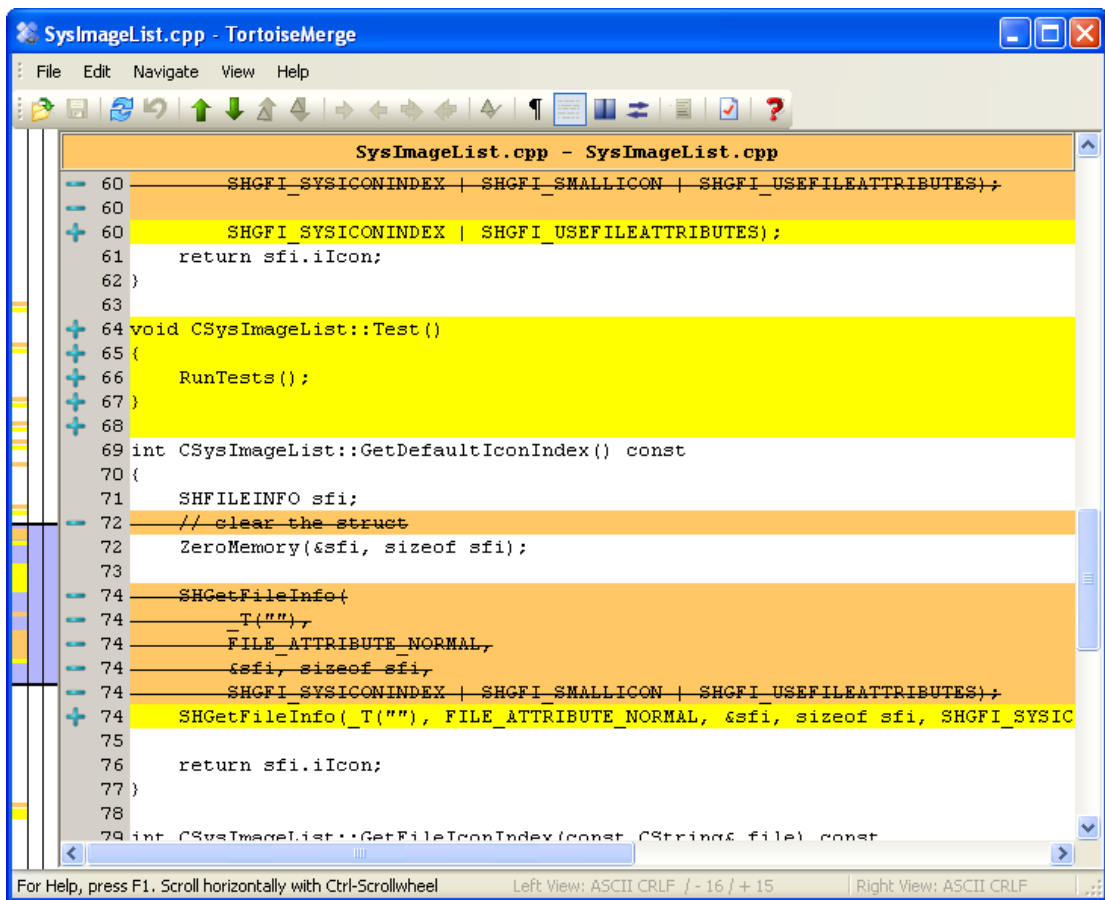


図 3.1. 1 ペイン表示

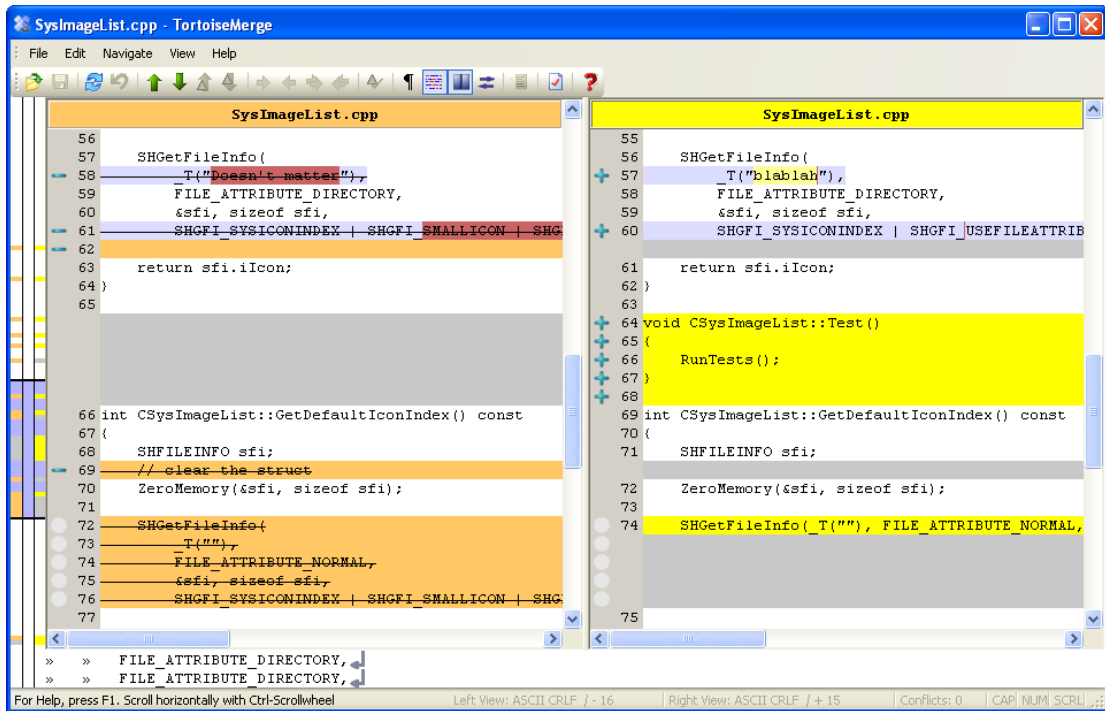


図 3.2. 2 ペイン表示

2 ペイン表示には、1 ペイン表示にはない以下の特徴があります。

- ・ 変更があった行の変更点を、色を変えて表示します。文字列が追加された部分は明るい色になります（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で文字列に縦線が入ります。どのように動作するかは、上記のスクリーンショットを確認してください。
- ・ コードの再編成をすると、ホワイトスペース（空白、タブ、改行）に大量に変更があっても、実際のコードには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、といったことです。

そういった変更は、表示の左側に白円で表します。そういった白円がある場合には、その実際のコードには変更がなく、変更箇所のチェックを行う必要がないことがすぐわかります。

- ・ 2 ペイン表示では、右ペインにあるファイルの手編集が可能です。そのような変更は鉛筆マークが付きます。

3 ファイルを比較・マージする場合、TortoiseMerge は 3 ペインで差分を表示します。ファイルの競合を解消する必要があるときにも便利です。

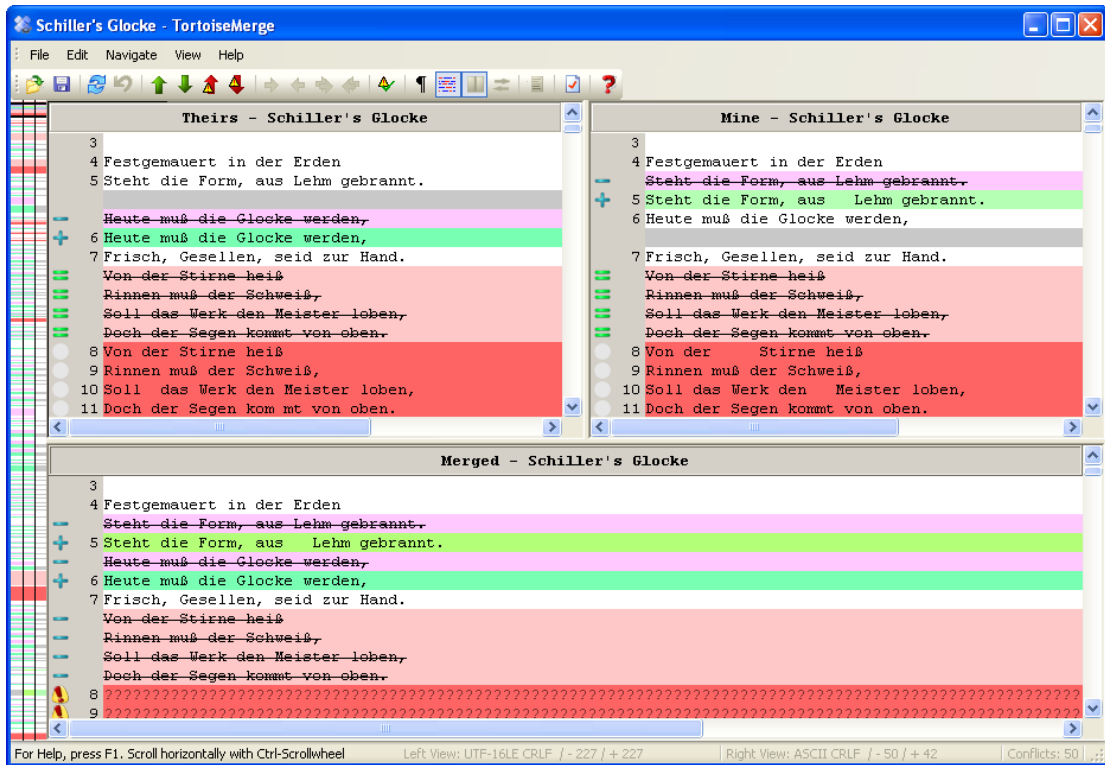


図 3.3. 3 ペイン画面

左ペインには Their ファイルと Base ファイルの差異を表示し、右ペインには Mine ファイルと Base ファイルの差異を表示します。下部ペインには、競合した Base, Mine, Theirs をマージした結果を表示します。

ペインタイトルにマウスを持っていくと、どちらのペインでも差分をとっているファイル名をツールチップに表示します。

3.1.2. パッチを適用

TortoiseMerge がパッチファイルをパースすると、そのパッチファイルで変更するファイルを表示する、小さなウィンドウを表示します。

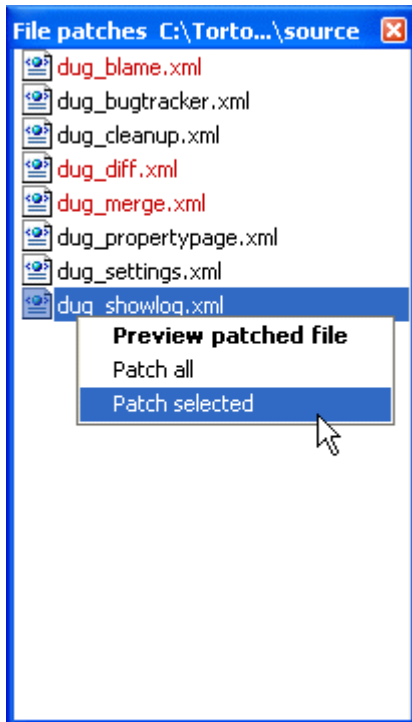


図 3.4. パッチをあてるファイル一覧

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られます。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、既にファイルを変更してしまっているため、直接パッチを当てることができません。

でも、ご心配なく。Subversion で作業していれば、古いバージョンのファイルを取り出し、競合を解消できます。実際、TortoiseMerge は自動で古いバージョンのファイルを取り出し、Base (古いバージョンのファイル)、Mine (現在作業コピーにあるファイル)、Theirs (古いバージョンのファイルにパッチが当たったファイル) の差分を表示します。以上から手動で競合を解消できます。

パッチファイルウィンドウは、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、コンテキストメニューがあります。ダブルクリックするとプレビューします。

TortoiseMerge がパッチを直接適用できるか、まず古いバージョンを取得してから適用するかによって、2 ペイン表示 (直接適用) か 3 ペイン表示 (古いバージョンの取得) で表示します。

3.2. ボタンやその他のコントロール



図 3.5. ツールバー

メニューバーの下にツールバーがあります。ツールに共通な機能に簡単にアクセスできるようにします。

- ・ **開く** 「ファイルを開く」ダイアログにアクセスします。他に何が?
- ・ **保存** 行った変更を保存します。

- ・ **再読み込み** TortoiseMerge で行った変更を取り消し、ファイルをディスクから再度読み込みます。
- ・ **元に戻す** TortoiseMerge で行った変更を元に戻す、マルチレベルアンドゥを実装しています。
- ・ **前/次の相違点へ** 次のファイルの差異に直接ジャンプします。ファイルの未変更点を素通りしてスクロールをしなくても良くなります。
2 ペイン差分モードの場合、画面下部の詳細バーに、マウスの場所に関係なく先頭行にある差分を表示します。
- ・ **前/次の競合へ** 競合しており解消する必要のある、次のファイルの差異に直接ジャンプします。
- ・ **Theirs を使用/Mine を使用** 左右のペインのどちらを使用するか選択します。変更と競合の編集に関する詳細は、次節をご覧ください。
- ・ **Mine の前に theirs/theirs の前に Mine** は、競合したファイルで変更を結合する際に使用します。変更と競合の編集に関する詳細は、次節をご覧ください。
- ・ **解消マークをつける** 競合をすべて解消して変更を保存したら、エクスプローラに戻って解消マークをつけなくても、TortoiseMerge で解消マークをつけられます。
- ・ **空白を表示** 空白に変更があることが分かるように、空白やタブ文字をシンボルで表示するオプションを切り替えます。
- ・ **行内差分 (単語単位)** は、行内差分の表示を切り換えます。単語単位モードでは、文字が変更されていると、その単語全体を強調表示します。文字単位モードでは、変更のあった各単語ごとに強調表示します。
デフォルトは単語単位で、ほとんどの場合これで十分です。例えば、変数名を変更した場合、単語単位モードでは、その変数名全体を強調表示します。ところが、文字単位モードでは、文字の追加削除でごちゃごちゃし、混乱しやすくなります。しかし、きちんとした単語にならない場合もあります。例えば 16 進ファイルの場合、文字単位の表示の方が便利です。
- ・ **ビューの切り替え** 1 ペインビューと 2 ペインビューを選択します。3 ペインビューでは無効です。
- ・ **左と右のビューを入れ替え** 2 ペインビューで左右のウィンドウを入れ替えます。同じファイルの 2 つのリビジョン間で比較するのに対して、2 つの異なるツリーで比較を行っている場合、TortoiseSVN はどちらのファイルを 「オリジナル/ベース」 版と見なして左ペインに表示したらいいか分かりません。間違っていた場合、これで簡単に表示を切り替えられます。
- ・ **パッチリストの非表示/表示** TortoiseMerge にパッチリストがある場合、このボタンで非表示にしたり (ウィンドウ参照)、表示したり (ファイルリストの確認) できます。
- ・ **設定** TortoiseMerge 設定ダイアログを表示します。

ウィンドウの左側にあるのは位置指定バーです。これにより、ファイルにある変更点を素早く参照できます。位置指定バーは 3 列あります。左の列は左ペインを、右の列は右ペインを、中央の列は (表示されている場合) 下部ペインを参照します。1 ペインビューでは左の列のみ使用します。位置指定バーは、ウィンドウ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。

下部ウィンドウの下は、ステータスバーです。Theirs と Mine の追加・削除行数や、解消されていない競合数を表示します。また、使用している改行スタイルや、TortoiseMerge が仮定した文字のエンコード (ASCII, UTF-8, など)、UTF-8 のファイルのバイトオーダーマーク (BOM) の有無も表示します。ステータスバーの左側は、マウスでコントロールを指す際に、ヒントを表示するのに使用します。

3.3. 行状態アイコン

変更のある行は、どのような変更が行われたかを表すアイコンで、マーク付けがなされます。



追加した行です。



削除した行です。



オリジナルな内容に戻して、変更されていない行です。



この行には空白の変更しかありません。連続した行がマークされている場合、単語が隣接行に移されて、再度折り返しが行われた可能性があります。



追加した行です。



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。

3.4. 競合のマージ・編集

TortoiseMerge はファイルの差分を 表示 するだけでなく、競合を解消したり変更を適用したりできます。

2 ペインビューの場合、右ペイン (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利用** を選択してください。その後、左のファイルの変更を右のファイルに加えてください。

時には、実際に両方のテキストブロックが必要になります。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利用 (こちらを先にする)** や **コンテキストメニュー → 両方のテキストブロックを利用 (こちらを後にする)** を使用してください。

また、テキストエディタのように出力ファイルの編集もできます。その行は鉛筆アイコンで印がつけられます。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しおけなくなります。

3 ペインビュー (時に マージビュー と呼ばれます) の場合、下部ビューにあるファイル (Merged) のみ編集できます。2 ペインビューと同様に、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利用** と **コンテキストメニュー → 「mine」 のテキストブロックを利用** 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利用** 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利用** 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、Merged ファイルを変更します。

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。競合を解消するために、どのバージョンを採用するか選択する必要があります。

競合によっては、Theirs と Mine から選択するだけで解消するわけに行かず、手で編集しなければならないことがあります。TortoiseMerge は、まず簡単な変更を行ってしまってから保存し、手で編集して解消する難しい変更のみにできます。



重要項目

再度同じファイルで TortoiseMerge を使用すると、TortoiseMerge か手で編集かに関わらず、作業コピーに行った変更が 取り消され、競合編集を行い始めた状態のファイルになることに注意してください。

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 を起動すると、**ファイル** → **開く** を使用して、自分でファイルを開かなければなりません。

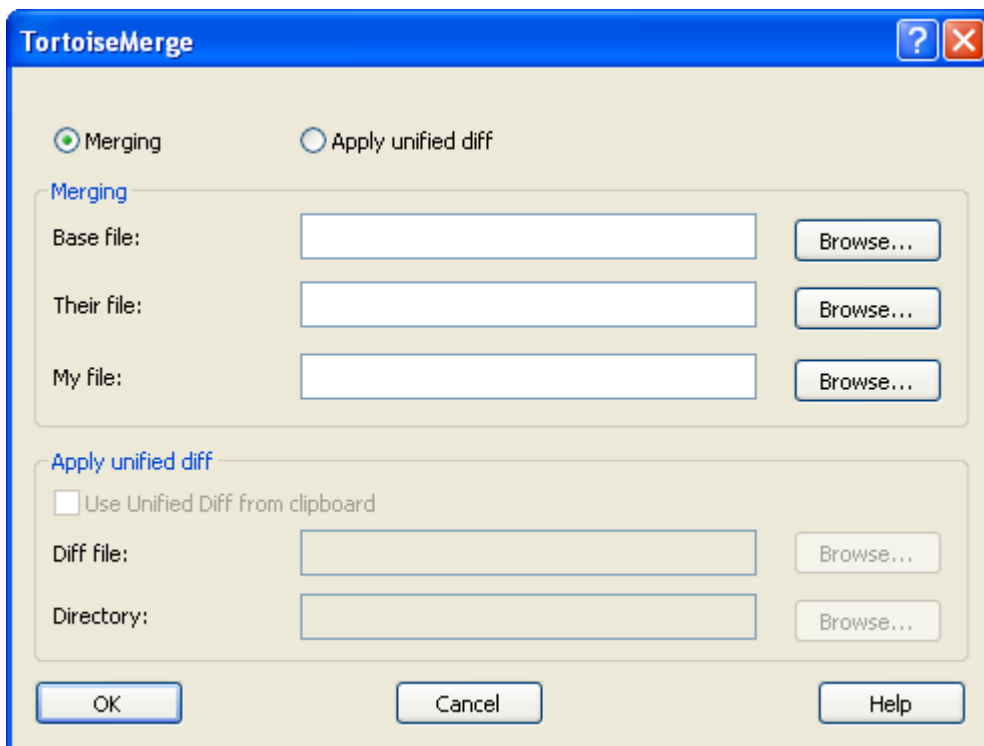


図 3.6. 開くダイアログ

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ズ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 つパスを設定しなければなりません。2 つだけファイルを指定すると、TortoiseMerge は、その 2 つのファイルの差分を 2 ペインか 1 ペインのどちらかで表示します。

3 ファイルを比較・マージする場合、TortoiseMerge は 3 ペインビューで差分を表示します。このビューはファイルの競合を解消する必要があるときにも便利です。この場合、出力

ファイルには名前が付かず、結果の保存に **File → 名前を付けて保存...** を使う必要があります。

3.5.2. パッチを適用

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のパスの両方を、設定しなければなりません。



注意

現在、Unified Diff Format しかサポートしていません。さらに、Subversion の作業コピーで作成されたファイルのみです。CVS で作成されたパッチファイルのような他の形式は、現在サポートしていません!

3.6. 設定

3.6.1. メイン設定ページ

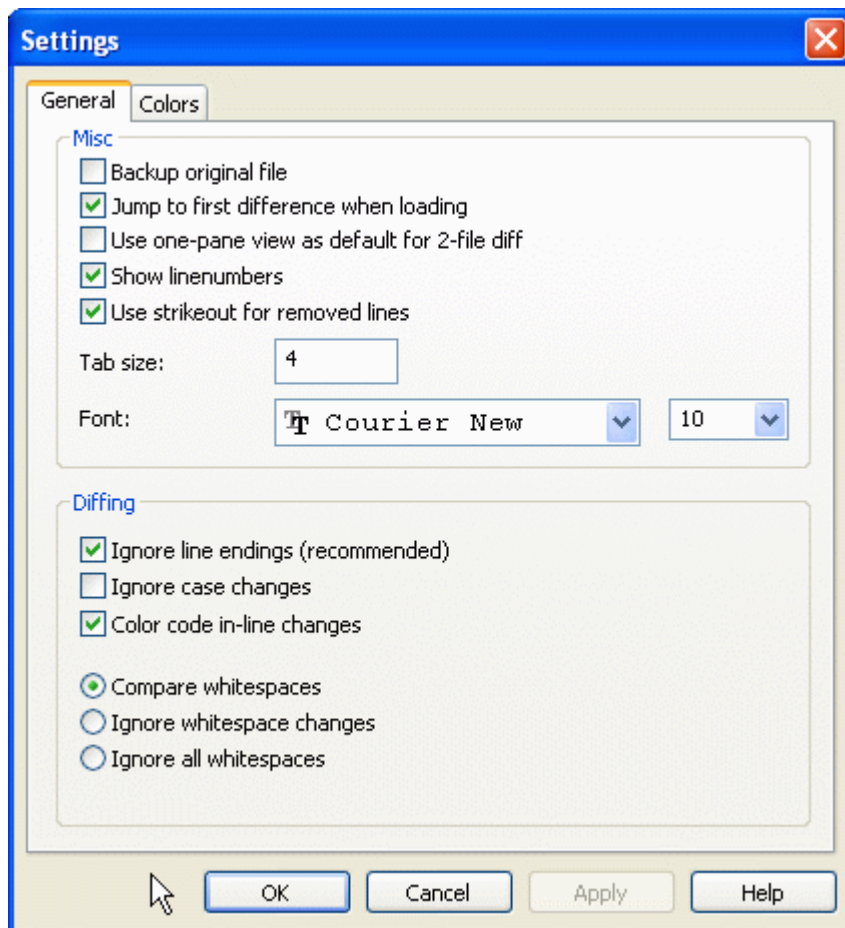


図 3.7. 設定の一般ページ

ほとんどのオプションは自己説明的ですが、いくつか要点を明らかにしておきます。

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を filename.bak に名前を変更します。

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 テキスト中で、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。

変更された行内の色コード メイン表示ペインの行内の変更を、差分バーと同様に色分けします。大きく変更された行は、この方法で表示すると分かりづらいです。この場合、行を置き換えたように（古い行の削除と新しい行の追加）表示します。

空白を比較 行の追加・削除と同様（Subversion の扱いと同様）に、インデントや行内の空白の変更を示します。

空白の変更は無視する 空白の種類や量しか変わらない変更を隠します。例：インデントの変更や、タブとスペースの変更など。とはいえ何もないところへの空白の追加や、空白の完全削除は表示します。

空白を無視する 空白のみの差異を隠します。

3.6.2. 色設定ページ

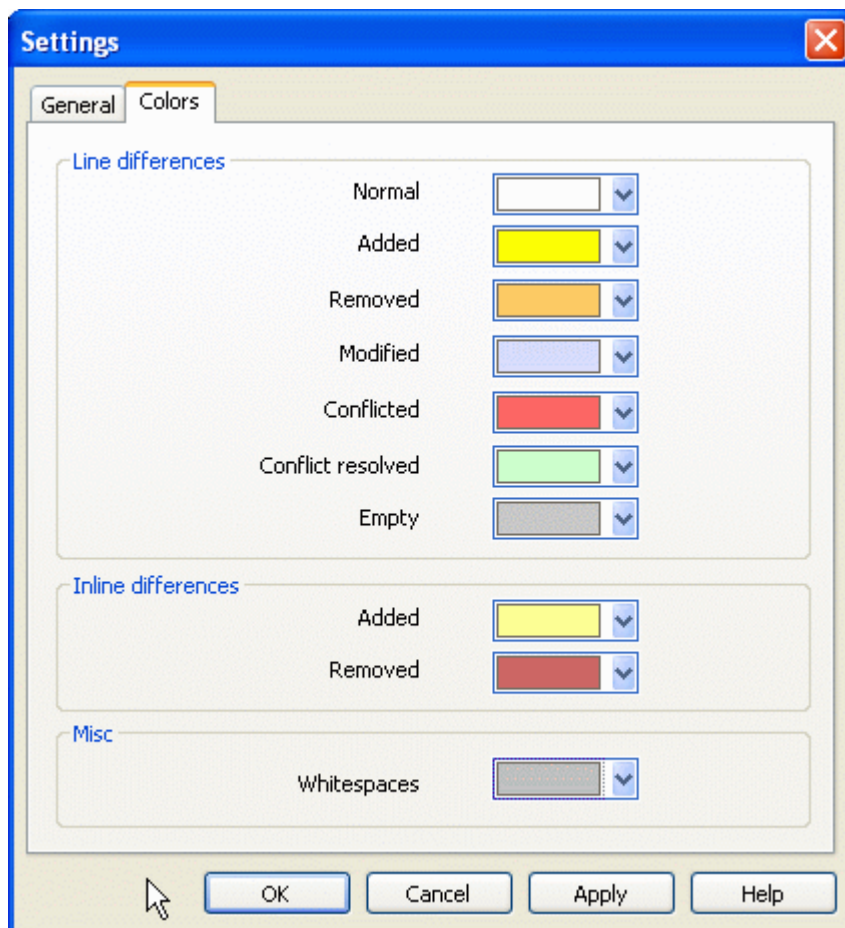


図 3.8. 色の設定ページ

このダイアログで変更がある行をハイライトする背景色を選択できます。

通常

全行が変更されていないか、変更が隠されている。

追加

追加された行。

削除

削除された行。

変更

比較的变化が少ない行は、行差分の形で表示します。この色は行の中の 未変更 箇所に使用します。変更箇所は、以下で説明する行追加色や行削除色で表します。**変更された行内の色コード** が無効の場合、この色を使用せず、変更行を必ず置換で表します。

競合

両方のファイルで同じ行が変更されていた。

解消した競合

両方のファイルで同じ行が変更されており、どの版を使用すべきか選択した部分です。

空

反対のペインにある行が、このペインには存在しない部分

行内追加テキスト

行差分で表示時、追加テキストをこの色で表します。

行内削除テキスト

行差分で表示時、削除テキストをこの色で表します。

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. キーボードショートカット

キーボードショートカットとコマンドの一覧

A.1. キーボードショートカット

Ctrl-Q

プログラムの終了

Ctrl-C

選択したテキストをクリップボードへコピーします

Ctrl-F

文字列検索用に検索ダイアログを開く

Ctrl-O

diff/merge 用にファイルを開く

Ctrl-S

変更の保存

Ctrl-Shift-S

名前を付けて保存...

F7

次の相違へ移動

Shift-F7

前の相違へ移動

F8

次の競合へ移動する

Shift-F8

前の競合へ移動する

Ctrl-D

1 ペイン diff と 2 ペイン diff を切り替える

Ctrl-R

ファイルを再読込し、すべての変更の取り消す

Ctrl-T

空白の表示・非表示を切り替える

Ctrl-マウスホイール

左右の表示をスクロールします。

付録 B. TortoiseMerge の自動化

TortoiseMerge は、ファイルを選択する **開く** ダイアログを出さずに、コマンドラインパラメータを指定して起動できます。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 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といった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・パスを指定してください。例:

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

コマンド	説明
<code>/?</code>	非常に重要なコマンドラインスイッチをダイアログに表示します。
<code>/help</code>	? と同様です。
<code>/base</code>	3 方向 diff で使用する base ファイルを指定します。独立した画面にはできませんが、これが共通の祖先になります。2 方向 diff なら左に表示します。
<code>/basename</code>	base ファイルの名前です。ファイルのパスの代わりにタイトルに表示します。3 方向 diff では、ビュータイトルのツールチップに表示します。
<code>/theirs</code>	3 方向 diff で使用する theirs ファイルを指定します。左ペインに表示します。
<code>/theirsname</code>	theirs ファイルの名前です。ファイルパスの代わりにビュータイトルに表示します。
<code>/mine</code>	3 方向 diff で使用する mine ファイルを指定します。右ペインに表示します。2 方向 diff では右のファイルです。
<code>/minename</code>	mine ファイルの名前です。ファイルパスの代わりにビュータイトルに表示します。
<code>/merged</code>	3 方向 diff で結果として得られる merged ファイルを指定します。マージや競合解消した結果を保存するファイルパスです。設定しない場合、TortoiseMerge はどこに保存するか訊いてきます。
<code>/mergedname</code>	merged ファイルの名前です。ファイルパスの代わりにビュータイトルに表示します。
<code>/patchpath</code>	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、非常に長い時間がかかります。
<code>/patchoriginal</code>	パッチを当てるオリジナルファイルの名前です。ビュータイトルに使われます。
<code>/patchpatched</code>	パッチを当てたファイルの名前です。ビュータイトルに使われません。
<code>/diff</code>	パッチ・diff ファイルをディレクトリに適用するパスです。
<code>/oneway</code>	TortoiseMerge を、ユーザが指定したビューではなく、強制的に 1 方向ビューで起動します。
<code>/reversedpatch</code>	差分を取るよう指定した 2 ファイルの左右のビューを入れ替えます。

コマンド	説明
/createunifieddiff	<p>/origfile:"path_to_original_file" と /modifiedfile:"path_to_modified_file" で指定したファイルから、unified diff 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:"path_to_resulting_patchfile" にセットします。/outfile を設定していない場合、ファイルの保存ダイアログを表示し、パッチファイルを保存する場所を指定できます。/createunifieddiff をセットした場合、他のパラメータをすべて無視することに注意してください。</p>

表 B.1. 有効なコマンドライン一覧

他の diff プログラムと互換があるように、コマンドラインに単にファイル名を与えることもできます。コマンドラインの単純な形は以下になります。

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

ファイルを 2 つ与えると、相互に比較します。3 ファイル与えると、最初の物を BASE ファイル、他の 2 つで比較する 3 方向 diff となります。

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ののリビジョンです。BASE リビジョンは、通常 HEAD リビジョンと同じではありません。
BDB	Berkeley DB。十分テストされているリポジトリ用データベースバックエンドですが、ネットワーク共有できません。1.2 以前のリポジトリのデフォルトです。
FSFS	リポジトリ用の Subversion が持つファイルシステムバックエンドです。ネットワーク共有が可能です。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の最新リビジョンです。
Revision	変更セットのコミット時、常に新しい「リビジョン」をリポジトリに作成します。各リビジョンは、履歴の決まった場所に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指し示しています。
SVN	Subversion のよく使われる省略表現。 「svnserve」リポジトリサーバで使われる、Subversion カスタムプロトコルの名前です。
インポート	ひとつのリビジョンで、フォルダ階層のエントリをリポジトリにインポートするコマンドです。
エクスポート	このコマンドは、作業コピーと同様にバージョン管理下のフォルダをコピーしますが、.svn は作成しません。
クリーンアップ	以下 Subversion book から引用します。「作業コピーを再帰的にクリーンアップ（ロックの削除、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	手元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 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、手元の作業コピーを作成する Subversion コマンドです。
パッチ	作業コピーにテキストファイルの変更のみある場合、Unified Diff 形式で変更内容の単一ファイルを作成するのに、Subversion の Diff コマンドを使用できます。この形のファイルは、よく「パッチ」と言われており、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できます。コミットアクセスできない人は、権限のあるコミッタが適用するように、変更をパッチファイルにして送ることができます。また、変更に自信がなければ、他の人に見てもらうようにパッチを送れます。
ブランチ	ある点で開発が2つの独立したパスに分岐したことをと表す、リビジョン管理システムの用語です。メインラインを不安定にせず、新機能の開発を行うように、メインの開発ラインからブランチを作成できます。また、今後バグフィックスしか行わない安定版リリースのブランチを作成し、新機能の開発は不安定なトランクで行えます。Subversion のブランチは、「簡易コピー」として実装されています。
マージ	リポジトリに追加された変更を、手元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 のマージツールを用いて、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。
リビジョン属性 (revprop)	ちょうどファイルが属性を持てるように、リポジトリの各リビジョンも属性を持てます。リビジョンが作成されるたびに、svn:date svn:author svn:log といった特殊な revprops が自動的に作成され、それぞれコミット日時、コミッタ、ログメッセージを表しています。これらの属性は編集できますが、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	データを格納し保守する中心部。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のリビジョンの歴史を表します。「履歴」とも表します。
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き、ロックが外れるまでリポジトリにコミット不可の印を付けます。
作業コピー	手元の「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。また通常手元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、リポジトリが移動する場合、作業コピー

	<p>を「再配置」して、リポジトリの URL を新しい場所に指し示してください。</p> <p>注意: このコマンドは、作業コピーが同じリポジトリ、同じ場所を指している、そのリポジトリが移動されてしまったときのみを使用してください。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。</p>
切り替え	<p>ちょうど「リビジョンへの更新」が履歴上の別のポイントへ、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ポイントへ、作業コピーの場所ウィンドウを変更します。違いが少ししかないトランクとブランチの双方に作業する際に、これが特に役に立ちます。その 2 つの間で作業コピーを切り替え、違いのあるファイルのみを転送します。</p>
削除	<p>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に、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。</p>
取り消し	<p>作業コピーを最後に更新したときから、Subversion はそれぞれのファイルの「当初の」コピーを手元に保持しています。変更を行い、その変更を取り消したい場合は、「取り消し」コマンドを使用して当初のコピーに戻せます。</p>
属性	<p>ディレクトリやファイルをバージョン管理下に置くのに加えて、Subversion はバージョン管理下の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・ファイルごとの「属性」として参照されます。属性には、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、svn:eol-style のような内部で使用する特殊な属性がいくつかあります。TortoiseSVN にも tsvn:logminsize のような特殊な属性があります。お好みの名前と値を持つ属性の追加もできます。</p>
履歴	<p>ファイルやフォルダのリビジョンの歴史を表します。「ログ」とも表します。</p>
差分	<p>「差分表示 (Show Differences)」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。</p>
更新	<p>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。</p>
注釈履歴	<p>このコマンドはテキストファイル専用で、すべての行に対して、最後に変更したリポジトリのリビジョンと、誰が変更したのかを注釈します。我々の GUI 実装では、TortoiseBlame を呼び出し、リビジョン番号の上にマウスを持っていくと、コミット日時やログメッセージも表示します。</p>
競合	<p>リポジトリの変更が手元にマージされる際、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、ファイルを手で修正し競合を解消しなければなりません。</p>

解消	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（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「競合の解消」と言われています。競合したファイルに解消マークを付けると、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。