

TortoiseMerge

**Uma ferramenta de diff/
unificação para Windows**

Versão 1.6.16

**Stefan Küng
Lübbe Onken
Simon Large**

TortoiseMerge: Uma ferramenta de diff/unificação para Windows: Versão 1.6.16

por Stefan Küng, Lübbe Onken, e Simon Large
tradutor-tradução: créditos de tradução

Publicado 2011/01/21 21:21:17 (r20750)

Índice

Preface	vi
1. TortoiseMerge é gratuito!	vi
2. Agradecimentos	vi
1. Introdução	1
1.1. Visão Geral	1
1.2. História do TortoiseMerge	2
2. Conceitos Básicos	3
2.1. Visualizando e Unificando as Diferenças	3
2.2. Editando Conflitos	3
2.3. Aplicando Correções	4
3. Usando TortoiseMerge	5
3.1. Visualizando Modos	5
3.1.1. Visualizando / Unificando	5
3.1.2. Aplicando Correções	7
3.2. Botões e Outros Controles	8
3.3. Ícones da Situação da Linha	10
3.4. Unificando / Editando Conflitos	10
3.5. Arquivos Abertos	11
3.5.1. Visualizando / Unificando	11
3.5.2. Aplicando Correções	12
3.6. Configurações	12
3.6.1. Páginas das Configurações Principais	12
3.6.2. Páginas de Configuração de Cores	13
A. atalhos do teclado	15
A.1. Atalhos do teclado	15
B. Automatizando TortoiseMerge	16
B.1. Opções da Linha de Comando do TortoiseMerge	16
Glossary	18

Lista de Figuras

1.1. Conflito do Arquivo	1
2.1. Conflito do Arquivo	4
3.1. Uma Painel de Visualização	5
3.2. Dois Painéis de Visualização	6
3.3. Três Painéis de Visualização	7
3.4. Lista de Arquivo de Correção	8
3.5. A Barra de Ferramenta	8
3.6. A Janela para Abrir Arquivos	11
3.7. A Principal Página de Configurações	12
3.8. A Página de Configuração de Cores	13

Lista de Tabelas

B.1. Lista de opções disponíveis na linha de comando	16
--	----

Preface



Tortoise Merge

Quando trabalhando em um projeto, mesmo em um projeto de código aberto onde várias pessoas revisam seu código, ou em um projeto comercial onde você vende um componente com seu código fonte, existirão pessoas e clientes que encontrarão problemas ou querem melhorias no seu projeto. Para fazer isso eles podem enviar as correções e você pode então revisar e decidir o que gostaria de aplicar.

TortoiseMerge ajuda você com estas tarefas:

- Revisando correções
- Aplicando correções

TortoiseMerge também ajuda você a resolver conflitos que podem ocorrer se você estiver trabalhando com um sistema de controle de versão e outras pessoas estão trabalhando nos mesmos arquivos que você.

1. TortoiseMerge é gratuito!

TortoiseMerge é gratuito. Você não precisa pagar nada por ele, você pode usar da maneira que quiser. Ele foi desenvolvido sobre a licença GP (GPL).

TortoiseMerge é um projeto de Código Aberto. Isto significa que você tem total acesso para ver o código deste programa. Você pode vê-lo no sítio <http://code.google.com/p/tortoisesvn/source/browse/>. Será necessário informar um usuário e uma senha. O usuário é `guest`, e a senha deve ser deixada em branco. A versão mais recente (a que está sendo trabalhada) está localizada em `/trunk/`, e as versões liberadas estão localizadas em `/tags/`.

Doe!

Even though TortoiseSVN and TortoiseMerge are free, you can support the developers by sending in patches and play an active role in the development. You can also help to cheer us up during the endless hours we spend in front of our computers.

While working on TortoiseSVN we love to listen to music. And since we spend many hours on the project we need a *lot* of music. Therefore we have set up some wish-lists with our favourite music CDs and DVDs: <http://tortoisesvn.tigris.org/donate.html> Please also have a look at the list of people who contributed to the project by sending in patches or translations.

2. Agradecimentos

Stefan Küng
pelo trabalho duro com o TortoiseMerge

Lübbe Onken
pelos belos ícones, logomarca, encontrar erros e cuidar da documentação

O Estilo de Projeto Tigris
por alguns modelos que reutilizados nesta documentação

Nossos contribuintes
pelas correções, reporte de problemas e novas idéias, e por ajudar aos outros respondendo perguntas na nossa lista de discussão.

Nossos colaboradores
pelas muitas horas de diversão com as músicas que nos enviaram



Capítulo 1. Introdução

1.1. Visão Geral

TortoiseMerge é uma aplicação gratuita/código aberto. Ele lhe permite ver as diferenças em arquivos textos, unificar estas alterações e até revisar e aplicar arquivos de diferenças unificados, também chamados *correções*.

Quando se trabalha com arquivos textos, por exemplo, código fonte em sua linguagem de programação favorita, ou arquivos html/xml de documentação ou do seu sítio, você muitas vezes precisará comparar diferentes versões destes arquivos. Algumas vezes você receberá uma versão diferente de alguém, outras vezes você apenas quer ver as mudanças que você tem feito.

Se você estiver trabalhando com um sistema de controle de versão (ex: Subversion) então você algumas vezes encontrará *conflitos* quando atualizar sua cópia de trabalho. Isto acontece se alguém faz uma alteração em partes do arquivo que você atualmente está trabalhando e alterou. Então você precisa resolver estes conflitos manualmente. É neste ponto que TortoiseMerge pode lhe ajudar.

O seguinte diagrama mostra a relação o relacionamento entre os arquivos envolvidos no conflito:

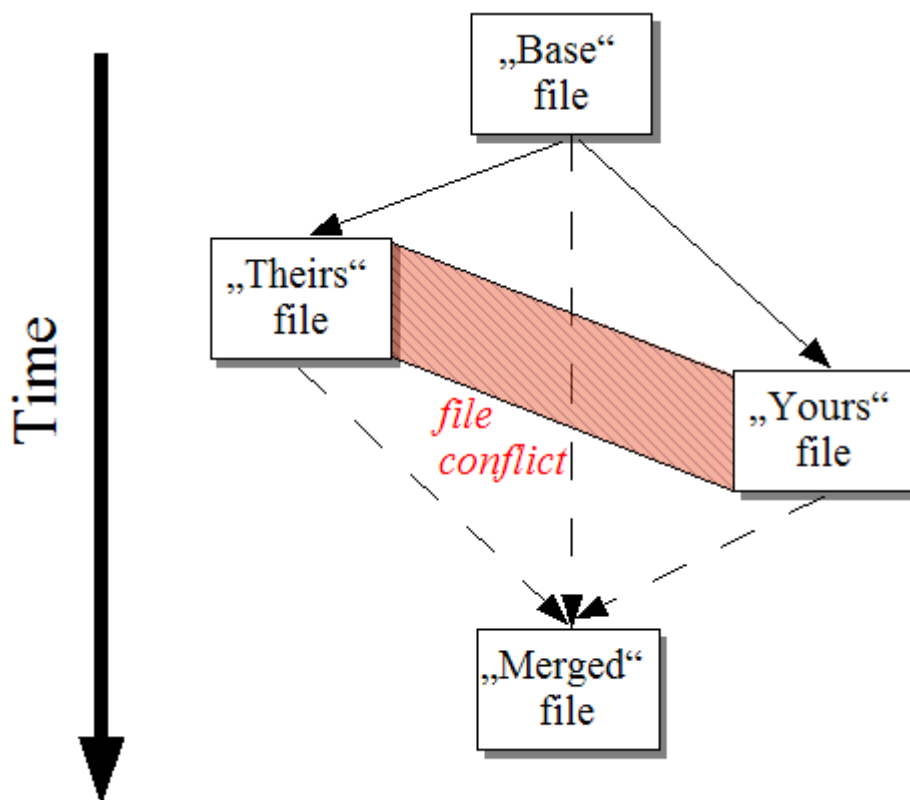


Figura 1.1. Conflito do Arquivo

Um conflito pode acontecer se você tentar aplicar uma correção desatualizada. Nestes casos, sem o TortoiseMerge, você poderia pedir a pessoa que lhe enviou a correção uma versão mais atualizada ou tentar aplicar a correção manualmente.

Mas agora com o TortoiseMerge isto não é mais necessário - pelo menos não se você estiver trabalhando com um sistema de controle de versão como o Subversion. TortoiseMerge analisa o arquivo de correção pela revisão/versão que a correção foi criado e automaticamente tenta aplicar a correção nesta versão

específica. Então ele aplica a correção e permite que você revise as suas mudanças e as mudanças da correção. Então você pode decidir quais alterações deseja manter.

1.2. História do TortoiseMerge

Durante o trabalho com o *TortoiseSVN* [<http://tortoisesvn.tigris.org>] descobrimos que sempre que os usuários nos enviavam melhorias para o projeto ou apenas correções, nós gastávamos muito tempo só aplicando essas correções e melhorias. E muitas vezes as correções eram enviadas de código desatualizados e que não poderiam ser aplicados usando a ferramenta de correção na linha de comando <http://unxutils.sourceforge.net/> porque nós estávamos trabalhando nos arquivos e algumas vezes já havíamos alterado os arquivos.

Esta foi também a principal razão pela qual TortoiseSVN originalmente não tem um TortoiseSVN → Criar Correção implementada: nós preferíamos que os usuários nos enviassem os arquivos inteiros pois assim ficava mais fácil aplicar a correção.

Então nós gastamos um tempo, procurando na internet por uma ferramenta visual que permitisse aplicar os arquivos de correção, mas descobrimos que tal ferramenta não existia. Então para encutar a história: Nós decidimos que se ninguém poderia prover tal ferramenta nós deveríamos fazer a nossa própria. Foi quando TortoiseMerge nasceu.

Já que TortoiseSVN já usava as bibliotecas do Subversion, da qual possuía uma boa ferramenta de diferenças construída, era natural usarmos tal biblioteca ao invés da biblioteca GNU.

Capítulo 2. Conceitos Básicos

TortoiseMerge tem uma porção de diferentes utilidades:

- Visualizando as diferenças entre dois arquivos e unificando as alterações em um único, ou removendo alterações indesejadas.
- Editando conflitos entre alterações locais e alterações no repositório do Subversion vindas de uma atualização.
- Aplicando arquivos de correções. Esta é a razão pelo qual TortoiseMerge foi criado :)

2.1. Visualizando e Unificando as Diferenças

Neste modo você está comparando dois arquivos. O arquivo no lado esquerdo é considerado o arquivo original (algumas vezes referenciado como *Deles*, e o arquivo do lado direito é o arquivo modificado (algumas vezes referenciado como *Meu*).

Você pode fazer apenas simples alterações na linha-base no arquivo do painel direito o qual inclui:

- Revertendo linhas alteradas para o mostrado no painel esquerdo
- Usando uma combinação de ambos os blocos, ou *Deles* antes *Meu* ou *Meu* antes *Deles*.

Você pode também editar o arquivo no painel direito como se você estivesse em um editor de texto. Assim as linhas são marcadas usando o ícone de um lápis. Por favor note que se você quer fazer qualquer alteração de linha/bloco-base descrita abaixo, é melhor estas alterações primeiro já que uma vez que começou a editar o arquivo se torna impossível para o TortoiseMerge rastrear o relacionamento com o arquivo original.

2.2. Editando Conflitos

Isto algumas vezes se refere a unificação de três arquivos, e neste caso mostra três painéis. Entretanto existem quatro arquivos envolvidos. O arquivo que não é mostrado é o arquivo base comum, o último ancestral comum de dois dos arquivos que estão em conflito. O relacionamento entre os três arquivos é explicado no diagrama que segue:

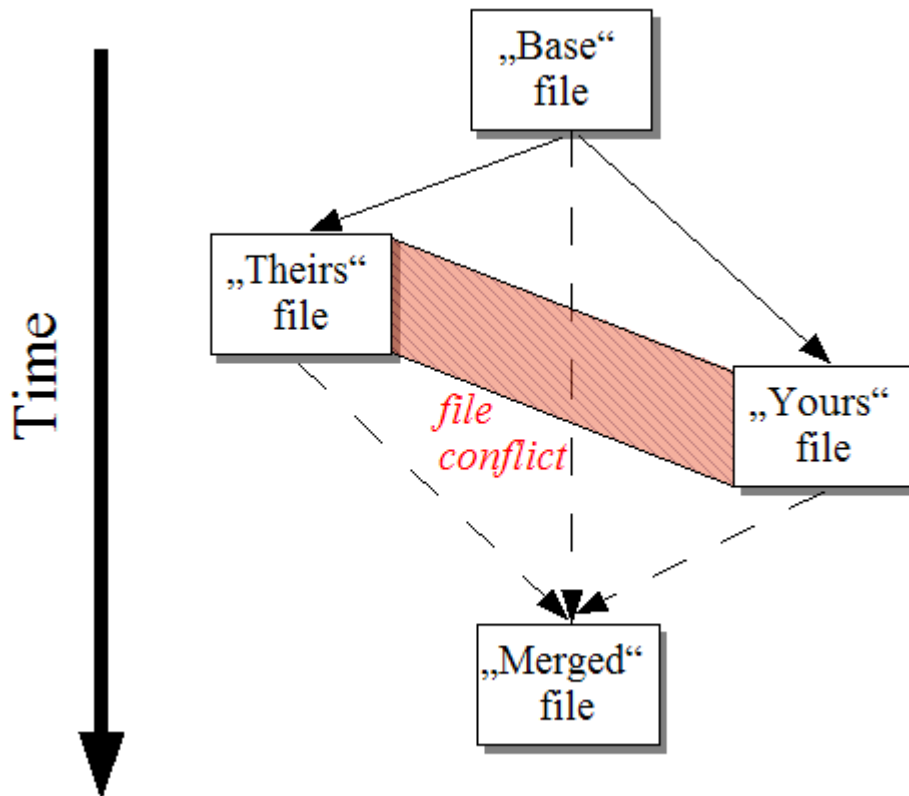


Figura 2.1. Conflito do Arquivo

O arquivo base representa a versão mais antiga do arquivo, de onde *Você* e *Eles* começaram as modificações. *Meu* representa o arquivo base com todas as mudanças que *você* fez, e *Deles* é o arquivo com todas as alterações que *alguém* fez no arquivo. O painel da esquerda mostra as modificações *Deles* relativo ao arquivo base, e o painel direito mostra as alterações no *Meu* arquivo em relação ao arquivo base. O painel inferior é o arquivo de saída onde você está tentando resolver os conflitos.

Na janela de resolução do conflito, você pode escolher usar blocos do *Meu* ou do *Deles* ou ambos. Mas neste caso, as alterações serão mostradas no painel inferior.

2.3. Aplicando Correções

Um arquivo de correção é um arquivo de diferenças unificadas do Subversion, o qual contém a informação necessária para aplicar as alterações nos arquivos. A correção pode ter sido enviada por outro desenvolvedor então você pode ver as alterações feitas e possivelmente submeter elas para o repositório. Ou o arquivo de correção pode ter sido gerado internamente pelo TortoiseSVN quando comparando dois diretórios. Isto acontece quando você compara uma revisão do repositório com a sua cópia de trabalho, ou se você compara duas revisões diferentes do repositório, ou se você compara dois caminhos diferentes no repositório.

Em cada caso TortoiseMerge vai mostrar uma pequena janela listando os arquivos incluídos na correção. Um duplo clique em um desses arquivos buscará o arquivo em questão e aplicará as mudanças. O painel da esquerda mostra o conteúdo do arquivo original e o painel da direita mostra o arquivo depois que as mudanças foram aplicadas.

Você pode editar o arquivo no painel direito exatamente como você faria em modo de comparação/edição.

Capítulo 3. Usando TortoiseMerge

3.1. Visualizando Modos

TortoiseMerge tem três principais modos de visualização: um painel, dois painéis e três painéis de visualização. Um/Dois painéis de visualização são usados para ver alterações e o terceiro painel é usado para resolver conflitos.

3.1.1. Visualizando / Unificando

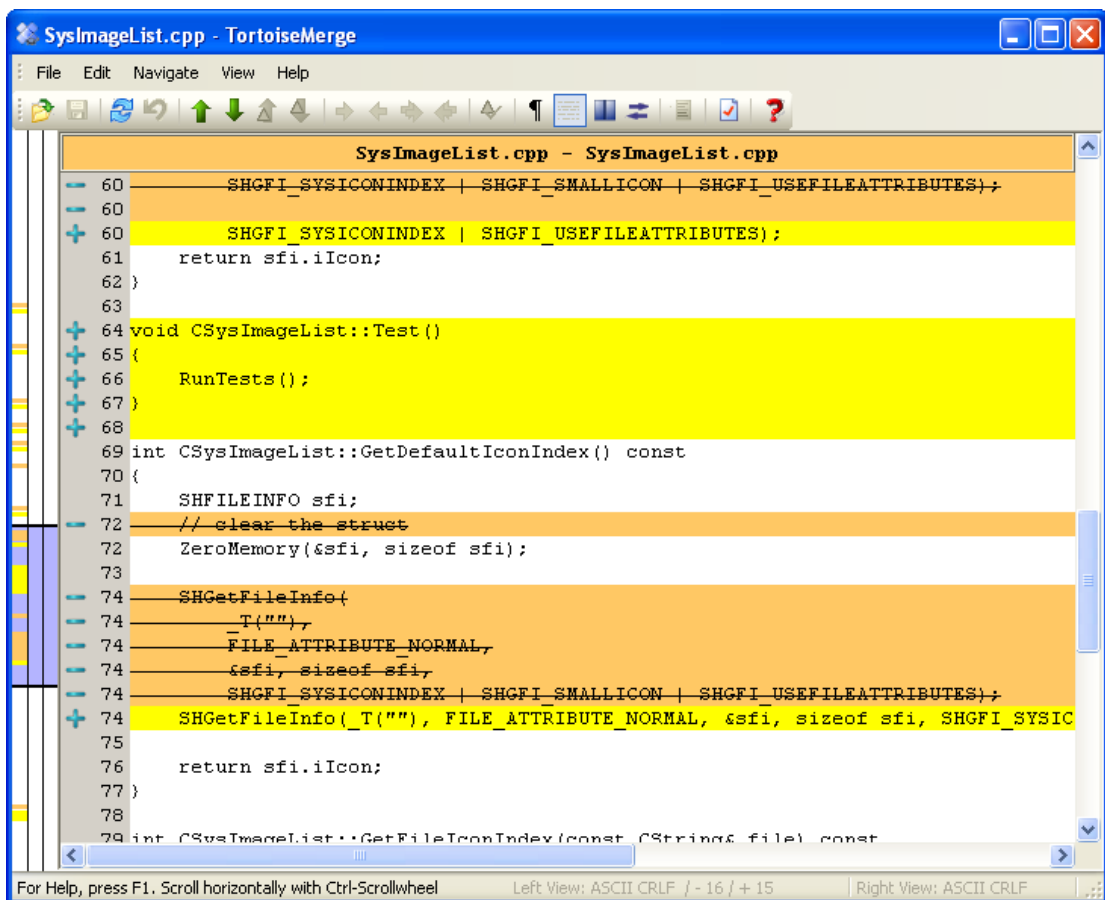


Figura 3.1. Uma Painel de Visualização

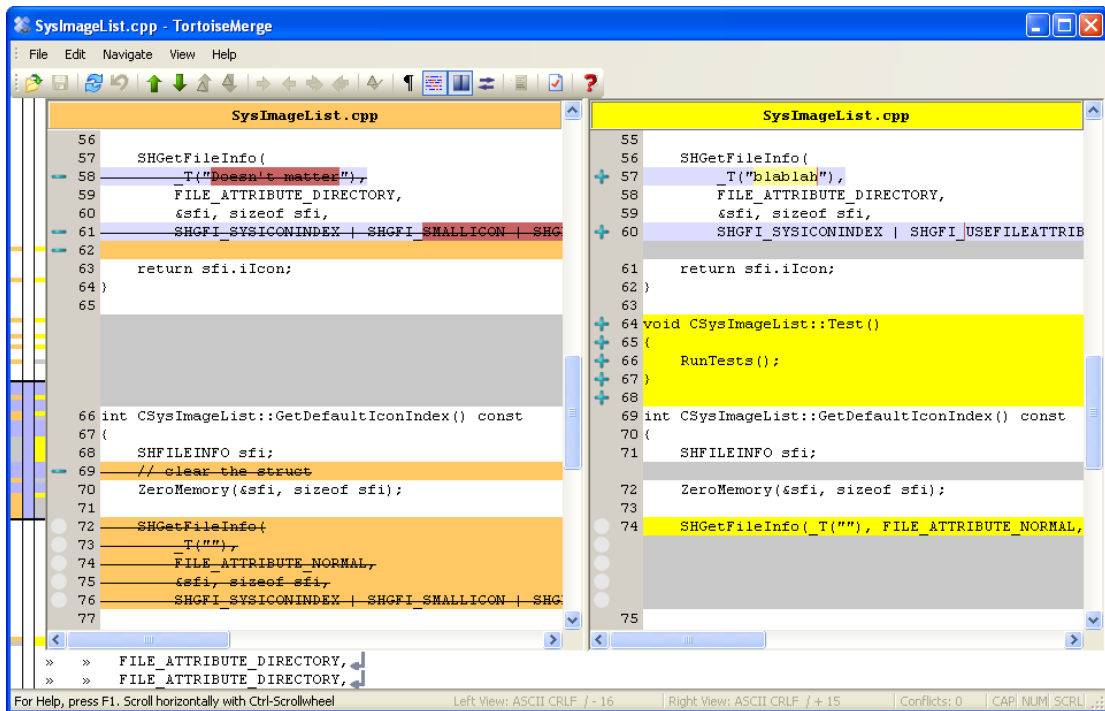


Figura 3.2. Dois Painéis de Visualização

A visualização com dois painéis tem algumas características que não estão disponíveis na visualização de um painel:

- As linhas modificadas são apresentadas em cores diferentes. Partes adicionadas são apresentadas com cores claras, mas claro que você pode configurar as cores que você quer usar. Partes removidas são mostradas com uma linha vertical marrom escura. Confira a imagem acima para entender como isso funciona.
- Reorganização de código geralmente significa uma porção de modificações com espaços em branco (espaços, tabulações, linhas novas) mas sem realmente alteração de código. Por exemplo, você dividiu uma linha extensa em várias linhas, ou você algumas vezes compactou várias linhas em uma só.

Cada uma dessas alterações são marcadas com um símbolo branco em forma de círculo no lado esquerdo do painel. Se você ve esse círculo branco, você sabe imediatamente que nenhuma modificação no código propriamente dito foi feita e você não precisa verificar todo o bloco alterado.

- A edição manual do arquivo no painel direito é possível em uma visualização com dois painéis. Cada alteração é marcada com um símbolo em forma de lápis.

Se você quer comparar/unificar três arquivos, TortoiseMerge vai mostrar para você as diferenças em três painéis. Esta visualização é também usado se você precisar resolver conflitos entre os arquivos.

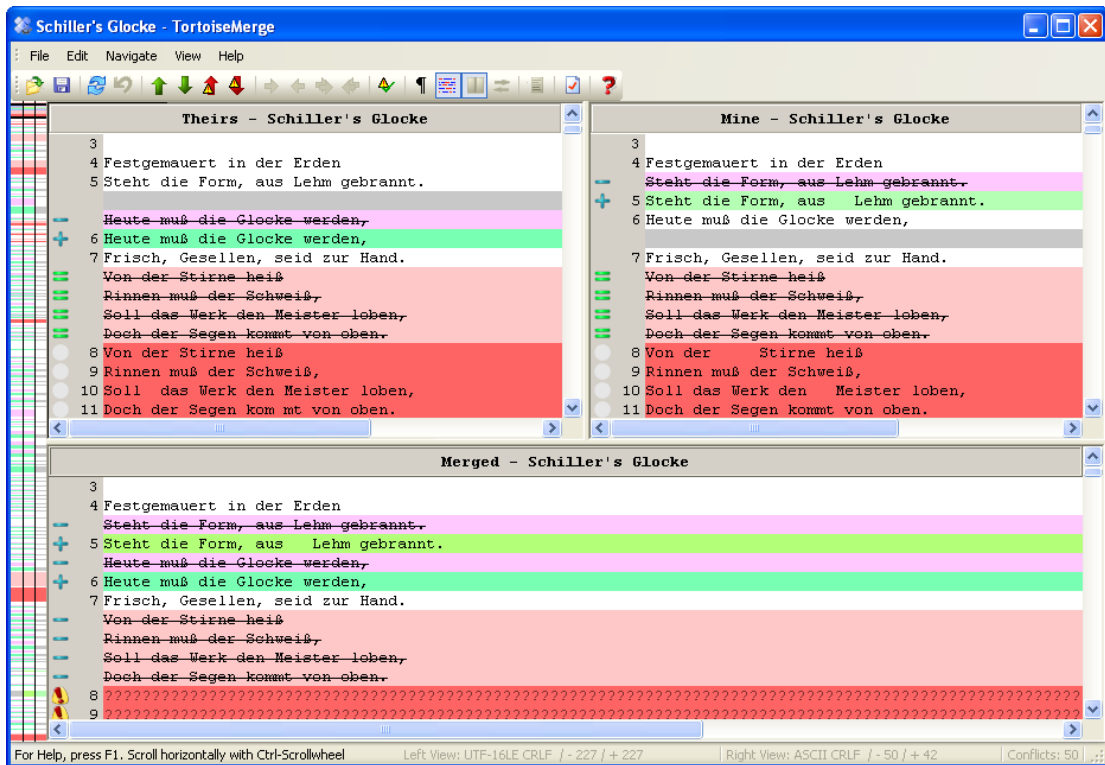


Figura 3.3. Três Painéis de Visualização

O painel da esquerda mostra as diferenças entre o arquivo *Deles* e o arquivo *Base*, enquanto o painel direito mostra para você as diferenças entre o arquivo *Meu* e o arquivo *Base*. O painel inferior mostra a você o resultado da unificação dos arquivos *Base*, *Meu* e *Deles* com possíveis conflitos.

Se você colocar o mouse sobre o título do painel, um balão será mostrado com o nome do arquivo usado para gerar as diferenças em cada painel.

3.1.2. Aplicando Correções

Depois que TortoiseMerge analisar o arquivo de correção ele irá mostrar a você uma pequena janela com todos os arquivos que possuem alterações conforme o arquivo de correção.

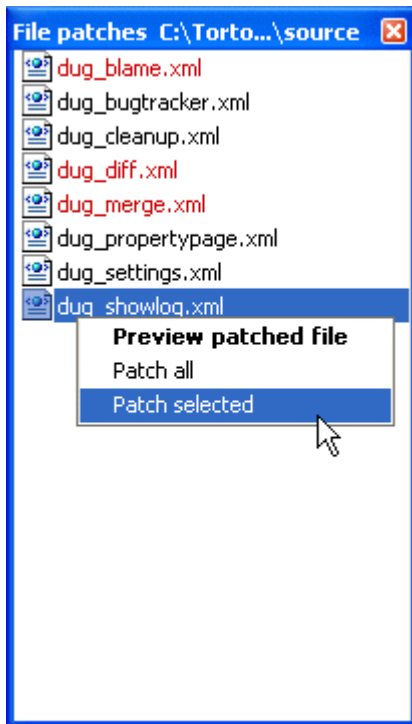


Figura 3.4. Lista de Arquivo de Correção

Se o nome do arquivo é mostrado em preto, então o arquivo de correção pode ser aplicado sem problemas. Isto significa que o arquivo não está desatualizado em relação à correção. Entretanto, se o nome do arquivo aparecer em vermelho, então a correção não pode ser aplicada diretamente porque você já tem alterado o arquivo.

Mas não se preocupe. Se você está trabalhando com o Subversion você pode recuperar uma revisão antiga do arquivo e resolver o conflito. De fato, TortoiseMerge automaticamente tenta recuperar uma versão antiga do arquivo e então mostra a você as diferenças entre os arquivos *Base* (a versão antiga do arquivo), *Meu* (o arquivo que está atualmente na cópia de trabalho) e *Deles* (a versão antiga do arquivo com a correção aplicada). Então você pode resolver possíveis conflitos manualmente.

A janela do arquivo de correção tem um menu flutuante que permite a você visualizar o resultado da correção sobre o arquivo atual (aplicar sem salvar), para aplicar e salvar as alterações para o arquivo selecionado, ou para aplicar e salvar as alterações para *todos* os arquivos da lista. Um duplo clique mostrará a visualização.

Dependendo se TortoiseMerge pode aplicar a correção diretamente ou primeiro recuperar um versão antiga do arquivo, ele mostrará a você dois painéis de visualização (correção pode ser aplicada diretamente) ou três painéis de visualização (uma versão antiga do arquivo foi recuperada).

3.2. Botões e Outros Controles



Figura 3.5. A Barra de Ferramenta

Abaixo da barra de menu fica a barra de ferramentas. Isto provê fácil acesso para as ferramentas mais usadas, que são

- Abrir acessa a janela de abertura de arquivo. O que mais seria?
- Salvar salva qualquer alteração feita.

- **Recarregar** discarta qualquer alteração feita no TortoiseMerge e recarrega o arquivo do disco.
- **Desfazer** implementa um desfazer de vários níveis, permitindo voltar qualquer alteração feita dentro do TortoiseMerge.
- **Ant/Prox Diferença** vai diretamente para o próximo ponto onde os arquivos estão diferentes. Desta forma não é necessário passar por partes não alteradas do arquivo.

Quando você usa o modo de 2-painéis de diferenças, a barra de detalhes na parte inferior da janela vai mostrar a você as diferenças na linha para a primeira linha, independente do lugar apontado pelo mouse.

- **Ant/Prox Conflito** vai diretamente para o próximo ponto onde os arquivos estão em conflito e que precisa ser resolvido.
- **Usar Deles/Usar Meu** seleciona se vai usar o bloco do painel esquerdo ou do painel direito. Veja a próxima seção para mais informações sobre edição de alterações e conflitos.
- **Deles antes Meu/Meu antes Deles** é usado quando você quer combinar as alterações em um arquivo conflitante. Veja a próxima seção para mais informações sobre edição de alterações e conflitos.
- **Marcar como Resolvido** Quando você resolveu todos os conflitos do arquivo e salvou as alterações, você pode marcar o arquivo como resolvido dentro do TortoiseMerge, ao invés de voltar para o explorer e marcar como resolvido lá.
- **Mostrar Espaços em Branco** alterna entre mostrar espaços e caracteres de tabulação como símbolos então você pode distinguir alterações de espaços.
- **Diferenças numa linha palavra por palavra** alterna o modo usado para mostrar as diferenças na linha. No modo palavra por palavra, quando um caracter é alterado, a palavra inteira é destacada. No modo caracter por caracter, cada caracter alterado é destacado.

Palavra por palavra é o modo padrão e na maioria dos casos é a melhor forma. Por exemplo, se você mudou o nome de uma variável, o nome inteiro é destacado no modo palavra por palavra, enquanto no modo caracter por caracter tende a ficar uma bagunça de caracteres adicionados e excluídos, o que não faz muito sentido. Entretanto, em alguns conteúdos o modo palavra por palavra não resolve de forma clara, como um arquivo hexadecimal por exemplo, e neste caso, caracter por caracter funciona melhor.

- **Alterar Visualização** seleciona entre um painel e dois painéis de visualização. Desabilitado no modo 3-painéis.
- **Alterar Visualização Esquerda e Direita** visualizando dois painéis, troca a janela da esquerda pela direita. Quando você está comparando dois arquivos de diferentes estruturas, como duas revisões diferentes do mesmo arquivo, TortoiseSVN não sabe qual deverá ficar no painel esquerdo como versão “original/base”. Se ao abrir estiver trocado, você pode trocar para visualizar mais facilmente.
- **Mostrar/Esconder Lista de Correções** Quando TortoiseMerge possui uma lista de arquivos, este botão permitirá a você esconder a lista (então você pode ver ambas as janelas), ou mostrá-la (então você pode rever a lista de arquivos).
- **Configurações** abre a janela de configurações do TortoiseMerge.

No lado esquerdo da janela está a barra de localização. Ela provê uma referência visual rápida como onde as alterações foram feitas no arquivo. A barra possui três colunas. A coluna da esquerda refere-se ao painel esquerdo, a coluna direito refere-se ao painel esquerdo, e a coluna do meio refere-se ao painel inferior (se estiver presente). Ao visualizar apenas um painel somente a coluna da esquerda é usada. A barra de localização pode também ser usada com uma barra de rolagem para se mover em todos os painéis simultaneamente.

Abaixo da janela inferior está a barra de situação. Ela mostra o número de linhas adicionadas ou removidas nos arquivos **Deles** e **Meu**, e o número de conflitos não resolvidos restantes. Você pode também ver qual formato de fim de linha está sendo usado, qual codificação (ASCII, UTF-8, etc) está sendo usada pelo

TortoiseMerge, e se existe algum byte-order-mark (BOM) presente em arquivos UTF-8. O lado esquerdo da barra de situação é usado para mostrar dicas quando você coloca o mouse sobre algum controle.

3.3. Ícones da Situação da Linha

Linhas alteradas são marcadas com um ícone para indicar que tipo de alteração sofreu.



Uma linha foi adicionada.



Uma linha foi removida.



Uma alteração foi desfeita usando a opção de reverter para o conteúdo original da linha.



Esta linha possui somente alterações de espaços. Onde várias linhas estão marcadas, o parágrafo deve ter sido remanejado, causando alguma mistura nas linhas adjacentes.



Uma linha foi editada manualmente, usando o TortoiseMerge como editor de texto.



Esta linha está em conflito.



Esta linha está em conflito, mas o efeito está escondido por causa de configurações de espaço em branco e fim-de-linha.

3.4. Unificando / Editando Conflitos

TortoiseMerge não somente *mostra* a você as diferenças entre os arquivos como também permite a você resolver os conflitos ou aplicar as alterações.

Se você está visualizando dois painéis, então você pode somente editar o arquivo do painel direito (Meu). Para aplicar alterações feitas no arquivo do painel esquerdo (Deles), clique-direito sobre a linha alterada e então escolha Menu do Contexto → Usar o bloco de texto “deles”. Então as alterações do arquivo do painel esquerdo serão adicionadas para o arquivo do painel direito.

Algumas vezes você quer usar ambos os blocos de texto, e o menu do contexto também oferece a você Menu do Contexto → Usar ambos os blocos de texto (este primeiro) e Menu do Contexto → Usar ambos os blocos de texto (este depois).

Você pode também editar o arquivo como você faria em um editor de texto. Cada uma das linhas são marcadas com um ícone em forma de lápis. Por favor note que se você quer fazer qualquer alteração nas linhas/blocos base descritas acima, é melhor fazer isso primeiro já que uma vez que você começar a editar o arquivo você mesmo se torna impossível para o TortoiseMerge manter a relação com o arquivo original.

Se você está visualizando três painéis (algumas vezes chamado *visualização combinada*) você pode somente editar o arquivo no painel inferior (Combinado). Como numa visualização com dois painéis você pode clique-direito sobre as linhas conflitantes e então selecionar Menu do Contexto → Usar o bloco de texto “deles” ou Menu do Contexto → Usar o bloco de texto “meu”. Para complementar, se você quer *ambos* os blocos, você pode selecionar Menu do Contexto → Usar bloco de texto “meu” antes “deles” ou Menu do Contexto Usar bloco de texto deles antes meu. De acordo com o comando que vocCombinado.

Algumas vezes um arquivo será marcado como conflitante no Subversion, mesmo quando você vê-lo usando o TortoiseMerge sem nenhum conflito. Isto acontece por causa das configurações de espaço em branco. Se você escolhe ignorar alterações de fim-de-linha ou espaço em branco, essas linhas serão marcadas usando o ícone de *Conflito Ignorado*. Para resolver esse conflito você precisará escolher qual versão você quer usar.

Algumas vezes o conflito é muito difícil de resolver apenas escolhendo Deles e Meu e requer uma edição manual. TortoiseMerge permite que você faça todas as alterações fáceis primeiro e então salve o arquivo, deixando somente o que é mais difícil para ser resolvido manualmente.



Importante

Note que se você está usando TortoiseMerge num mesmo arquivo novamente, qualquer alteração feita em sua cópia de trabalho, seja ela feita no TortoiseMerge seja em outro editor, *será discardada* e o arquivo vai aparecer como ele era quando a edição do conflito foi iniciada.

3.5. Arquivos Abertos

Quando você iniciar o TortoiseMerge sem qualquer opção na linha de comando você terá de abrir os arquivos manualmente, usando Arquivo → Abrir.

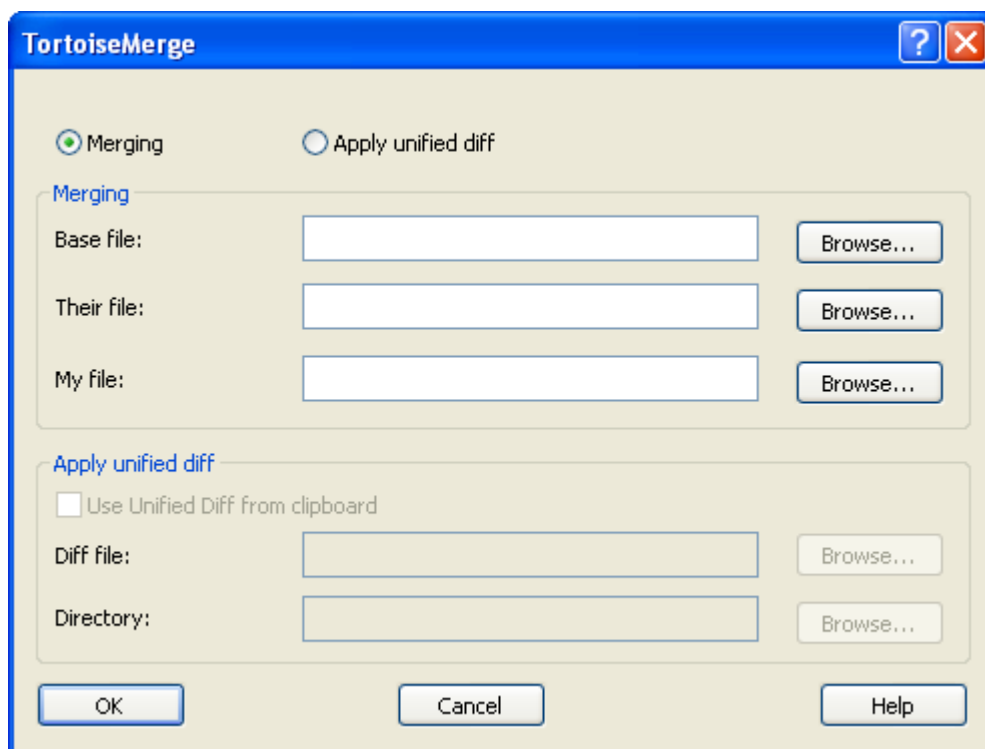


Figura 3.6. A Janela para Abrir Arquivos

A primeira coisa que você precisa fazer é decidir se você apenas quer comparar/combina arquivos, ou se você quer aplicar uma arquivo de correção. Dependendo da sua escolha, as caixas de edição e botões correspondentes serão ativados.

3.5.1. Visualizando / Unificando

Se você quer comparar/combina arquivos você tem que definir pelo menos dois de três possíveis caminhos para Base, Meu e Deles. Se você selecionar apenas dois arquivos, então TortoiseMerge vai mostrar a você as diferenças entre os dois arquivos, cada um em um painel ou em um único painel.

Se você quer combinar três arquivos, TortoiseMerge vai mostrar a você as diferenças em três painéis. Esta visualização é geralmente usada se você precisa resolver arquivos conflitantes. O arquivo de saída não possui nome, no caso, e você terá que usar o Arquivo → Salvar como...

3.5.2. Aplicando Correções

Se você quer aplicar um arquivo de correção você precisa definir ambos os caminhos para o arquivo de correção e para o diretório onde o arquivo de correção deverá ser aplicado.



Cuidado

Atualmente, apenas arquivos de correção no formato *Unified Diff* são suportados. E somente arquivos que foram gerados em uma cópia de trabalho do Subversion. Outros formatos como arquivos de correções gerados pelo CVS não são atualmente suportados!

3.6. Configurações

3.6.1. Páginas das Configurações Principais

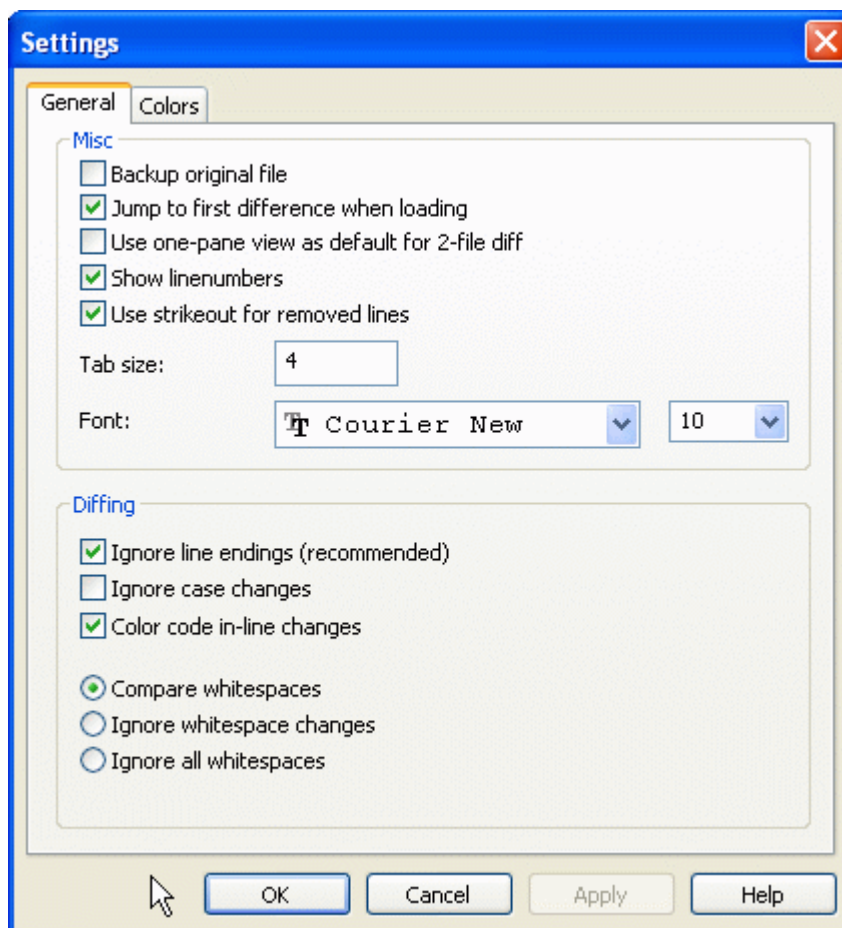


Figura 3.7. A Principal Página de Configurações

Muitas das opções aqui são auto-explicativas, mas alguns pontos precisam ser explicados.

Cópia de Segurança do arquivo original renomeia o arquivo original na CT para nomearquivo.bak antes de salvar o arquivo modificado.

Ignorar fim-de-linha oculta alterações feitas apenas na forma usada para indicar fim de linha.

Ignorar alterações na caixa da fonte esconde alterações que são apenas na caixa da fonte. Isto pode ser prático em aplicações como Visual Basic, no caso de alterações de caixa em variáveis sem qualquer aviso.

Cor de alterações do conteúdo da linha usa o mesmo esquema como a barra de difereças da linha para mostrar os detalhes do conteúdo da linha alterado dentro do painel principal. Linhas que possuem muitas modificações não serão mostradas usando este método pois o resultado seria confuso. Neste caso a linha será mostrada como substituída (a linha antiga apagada e a nova adicionada).

Comparar espaços em branco mostra todas as alterações indentadas e os espaços em branco da linha como adicionados/removidos (isto é como o Subversion trata cada alteração).

Ignorar alterações de espaço em branco esconde as alterações que são apenas de remoção ou inclusão de espaço em branco, ex: alterar a indentação ou substituir tabulação por espaço. Adicionar espaços em branco onde não havia nada, ou remover completamente espaços em branco é também apresentado como uma alteração.

Ignorar todos os espaços em branco esconde toda alteração de espaço em branco.

3.6.2. Páginas de Configuração de Cores

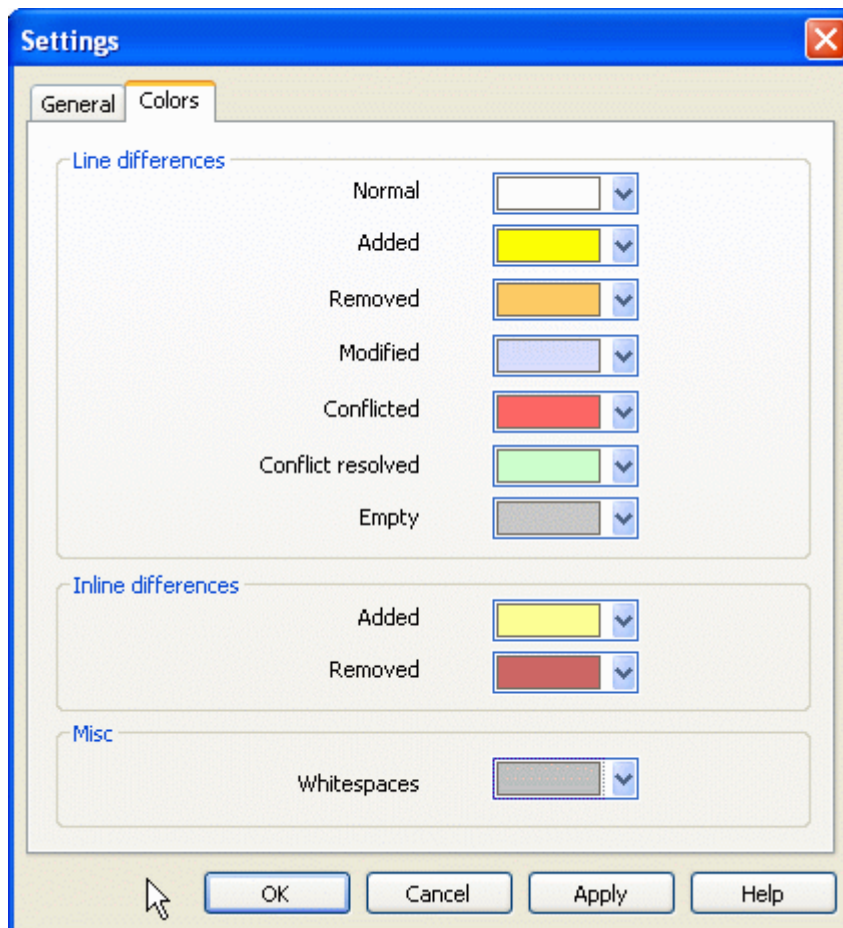


Figura 3.8. A Página de Configuração de Cores

Esta janela permite a você selecionar a cor de fundo do código usada para destacar as diferentes alterações de linha.

Normal

Todas as linhas das quais não estão modificadas, ou onde as alterações estão ocultas.

Adicionado

Linhas que foram adicionadas

Removidas

Linhas que foram removidas

Modificadas

Linhas das quais possuem relativamente pequenas mudanças, são mostradas com as diferenças do conteúdo da linha. Esta cor é usada para o *não modificadas* parte do conteúdo da linha. Partes alteradas serão mostradas usando as cores de conteúdo de linha adicionado ou excluído descritos abaixo. Se *não modificadas* não está habilitado, esta cor não será usada e as alterações da linha serão sempre apresentadas como substituídas.

Conflitantes

Onde a mesma linha foi alterada em ambos os arquivos.

Conflito resolvido

Onde a mesma linha foi alterada em ambos os arquivos, e você precisa agora selecionar qual versão deverá ser usada.

Vazio

Onde as linhas foram adicionadas no painel oposto, e nenhuma linha existe neste painel.

Texto adiciona na linha

Quando as diferenças do conteúdo da linha são apresentados, o texto adicionado é mostrado neste cor.

Texto removido da linha

Quando as diferenças na linha são apresentadas, textos removidos são mostrados nesta cor.

Espaços em branco

Caracteres usados para indicar Espaços em Branco são mostrados com cores diferentes da cor do texto normal.

Apêndice A. atalhos do teclado

Lista de atalhos do teclado e comandos.

A.1. Atalhos do teclado

- Ctrl-Q
Fechar o programa
- Ctrl-C
Copiar o texto selecionado para a área de transferência
- Ctrl-F
Abrir a janela de Busca para procurar textos
- Ctrl-O
Abrir arquivos para diferenças/unificação
- Ctrl-S
Salvar as alterações
- Ctrl-Shift-S
Salvar como..
- F7
Ir para a próxima diferença
- Shift-F7
Ir para a diferença anterior
- F8
Ir para o próximo conflito
- Shift-F8
Ir para o conflito anterior
- Ctrl-D
Anternar entre um painel com diferenças e dois painéis com diferenças
- Ctrl-R
Recarregar os arquivos e reverter todas as alterações feitas
- Ctrl-T
Alterna entre mostrar os espaços em branco ou não
- Ctrl-botão rolagem
Rolar a tela esquerda/diretia

Apêndice B. Automatizando TortoiseMerge

TortoiseMerge pode ser executado com parâmetros a partir da linha de comando evitando que você mesmo precise selecionar o arquivo através da janela Abrir. Isto também é útil se você quer usar TortoiseMerge através de outra aplicação.

B.1. Opções da Linha de Comando do TortoiseMerge

A maioria dos parâmetros requer informações adicionais como caminhos ou algum outro texto. Nestes casos, concatena um ':' ao parâmetro e coloque o texto/caminho depois do sinal. Exemplo:

```
/base:"c:\diretorio\meu arquivo base.txt"
```

Comando	Descrição
/?	Mostra uma janela com as opções mais importantes da linha de comando
/help	O mesmo que ?.
/base	Especifica o <i>base</i> arquivo usado na visualização de diferenças com três painéis. Este é o ancestral comum dos arquivos que serão comparados, apesar de que este arquivo não será mostrado em uma janela separada. Para a visualização com dois painéis, este é o arquivo da esquerda.
/basename	O nome do arquivo base. É mostrado no título do painel ao invés do caminho do arquivo. Para a visualização com três painéis o nome é mostrando em uma caixa de ajuda no título do painel.
/theirs	Especifica o <i>theirs</i> arquivo usado na visualização em três painéis, mostrado no painel esquerdo.
/theirsname	O nome do arquivo deles. É mostrado no título do painel ao invés do caminho do arquivo.
/mine	Especifica o <i>mine</i> arquivo usado na visualização com três painéis, mostrado no painel direito. Para a visualização de diferenças com dois painéis, este é o arquivo da direita.
/minename	O nome do meu arquivo. É mostrado no título do painel ao invés do caminho do arquivo.
/merged	Especifica o resultado <i>merged</i> arquivo usado na visualização com três painéis. Este é o caminho do arquivo onde o resultado da combinação/conflito resolvido é salvo. Se ainda não foi definido, então TortoiseMerge vai pergunta para o usuário onde o resultado será salvo.
/mergedname	O nome do arquivo combinado. É mostrado no título do painel ao invés do caminho do arquivo.
/patchpath	O caminho onde o arquivo de correção será aplicado. Se você não definir este caminho, então TortoiseMerge vai tentar encontrar o caminho por conta própria que corresponda aos caminhos no arquivo de correção. mas isto pode <i>muito longo</i> .
/patchoriginal	O nome do arquivo original da correção. Usado no título do painel.
/patchpatched	O nome do arquivo com as correções aplicadas. Usado no título do painel.
/diff	O caminho do arquivo de correção/diferença para aplicar para o diretório.

Comando	Descrição
/oneway	Força o TortoiseMerge a iniciar com a visualização de um caminho ao invés de usar o que o usuário configurou.
/reversedpatch	Muda a visualização da esquerda para a direita das diferenças de dois arquivos.
/createunifieddiff	Cria um arquivo de diferenças (arquivo de correção) de dois arquivos especificados com /origfile:"caminho_para_arquivo_original" e /modifiedfile:"caminho_para_arquivo_modificado", O arquivo final é definido com /outfile:"caminho_para_arquivo_correções". Se /outfile não está definido, a janela para salvar o arquivo é mostrada para o usuário escolher o local onde o arquivo de correção será salvo. Nota: Se /createunifieddiff está definido, todos os outros parâmetros são ignorados.

Tabela B.1. Lista de opções disponíveis na linha de comando

Você pode também fornecer apenas nomes de arquivos na linha de comando para compatibilidade com outros programas de diferenças. Esta é a forma simples de linha de comando

```
TortoiseMerge CaminhoArquivoBase CaminhoMeuArquivo [ CaminhoArquivoDeles ]
```

Se dois arquivos são informados eles vão ser comparados um com o outro. Se três arquivos são informados, o primeiro é usado como arquivo BASE, e os outros dois são comparados com este e apresentados usando a visualização com três painéis.

Glossary

Adicionar	A Subversion command that is used to add a file or directory to your working copy. The new items are added to the repository when you commit.
Apagar	When you delete a versioned item (and commit the change) the item no longer exists in the repository after the committed revision. But of course it still exists in earlier repository revisions, so you can still access it. If necessary, you can copy a deleted item and “resurrect” it complete with history.
Atualizar	This Subversion command pulls down the latest changes from the repository into your working copy, merging any changes made by others with local changes in the working copy.
Autoria	This command is for text files only, and it annotates every line to show the repository revision in which it was last changed, and the author who made that change. Our GUI implementation is called TortoiseBlame and it also shows the commit date/time and the log message when you hover the mouse of the revision number.
BDB	Berkeley DB. A well tested database backend for repositories, that cannot be used on network shares. Default for pre 1.2 repositories.
Bloquear	When you take out a lock on a versioned item, you mark it in the repository as non-committable, except from the working copy where the lock was taken out.
Combinar	<p>The process by which changes from the repository are added to your working copy without disrupting any changes you have already made locally. Sometimes these changes cannot be reconciled automatically and the working copy is said to be in conflict.</p> <p>Merging happens automatically when you update your working copy. You can also merge specific changes from another branch using TortoiseSVN's Merge command.</p>
Conflito	When changes from the repository are merged with local changes, sometimes those changes occur on the same lines. In this case Subversion cannot automatically decide which version to use and the file is said to be in conflict. You have to edit the file manually and resolve the conflict before you can commit any further changes.
Cópia de Trabalho	This is your local “sandbox”, the area where you work on the versioned files, and it normally resides on your local hard disk. You create a working copy by doing a “Checkout” from a repository, and you feed your changes back into the repository using “Commit”.
Copiar	In a Subversion repository you can create a copy of a single file or an entire tree. These are implemented as “cheap copies” which act a bit like a link to the original in that they take up almost no space. Making a copy preserves the history of the item in the copy, so you can trace changes made before the copy was made.
Correção	If a working copy has changes to text files only, it is possible to use Subversion's Diff command to generate a single file summary of those changes in Unified Diff format. A file of this type is often referred to as a “Patch”, and it can be emailed to someone else (or

			to a mailing list) and applied to another working copy. Someone without commit access can make changes and submit a patch file for an authorized committer to apply. Or if you are unsure about a change you can submit a patch for others to review.
Diff			Shorthand for “Show Differences”. Very useful when you want to see exactly what changes have been made.
Exportar			This command produces a copy of a versioned folder, just like a working copy, but without the local <code>.svn</code> folders.
FSFS			A proprietary Subversion filesystem backend for repositories. Can be used on network shares. Default for 1.2 and newer repositories.
GPO			Objeto de política de grupo
Histórico			Show the revision history of a file or folder. Also known as “Log”.
Importar			Subversion command to import an entire folder hierarchy into the repository in a single revision.
Limpar			To quote from the Subversion book: “ Recursively clean up the working copy, removing locks and resuming unfinished operations. If you ever get a <i>working copy locked</i> error, run this command to remove stale locks and get your working copy into a usable state again. ” Note that in this context <i>lock</i> refers to local filesystem locking, not repository locking.
Log			Show the revision history of a file or folder. Also known as “History”.
Obter			A Subversion command which creates a local working copy in an empty directory by downloading versioned files from the repository.
Propriedade			In addition to versioning your directories and files, Subversion allows you to add versioned metadata - referred to as “properties” to each of your versioned directories and files. Each property has a name and a value, rather like a registry key. Subversion has some special properties which it uses internally, such as <code>svn:eol-style</code> . TortoiseSVN has some too, such as <code>tsvn:logminsize</code> . You can add your own properties with any name and value you choose.
Propriedade (revprop)	da	Revisão	Just as files can have properties, so can each revision in the repository. Some special revprops are added automatically when the revision is created, namely: <code>svn:date</code> <code>svn:author</code> <code>svn:log</code> which represent the commit date/time, the committer and the log message respectively. These properties can be edited, but they are not versioned, so any change is permanent and cannot be undone.
Ramificação			A term frequently used in revision control systems to describe what happens when development forks at a particular point and follows 2 separate paths. You can create a branch off the main development line so as to develop a new feature without rendering the main line unstable. Or you can branch a stable release to which you make only bug fixes, while new developments take place on the unstable trunk. In Subversion a branch is implemented as a “cheap copy”.
Reposicionar			If your repository moves, perhaps because you have moved it to a different directory on your server, or the server domain name

has changed, you need to “relocate” your working copy so that its repository URLs point to the new location.

Note: you should only use this command if your working copy is referring to the same location in the same repository, but the repository itself has moved. In any other circumstance you probably need the “Switch” command instead.

Repositório	A repository is a central place where data is stored and maintained. A repository can be a place where multiple databases or files are located for distribution over a network, or a repository can be a location that is directly accessible to the user without having to travel across a network.
Resolver	When files in a working copy are left in a conflicted state following a merge, those conflicts must be sorted out by a human using an editor (or perhaps TortoiseMerge). This process is referred to as “Resolving Conflicts”. When this is complete you can mark the conflicted files as being resolved, which allows them to be committed.
Reverter	Subversion keeps a local “pristine” copy of each file as it was when you last updated your working copy. If you have made changes and decide you want to undo them, you can use the “revert” command to go back to the pristine copy.
Revisão	<p>Every time you commit a set of changes, you create one new “revision” in the repository. Each revision represents the state of the repository tree at a certain point in its history. If you want to go back in time you can examine the repository as it was at revision N.</p> <p>In another sense, a revision can refer to the set of changes that were made when that revision was created.</p>
Revisão BASE	The current base revision of a file or folder in your <i>working copy</i> . This is the revision the file or folder was in, when the last checkout, update or commit was run. The BASE revision is normally not equal to the HEAD revision.
Submeter	This Subversion command is used to pass the changes in your local working copy back into the repository, creating a new repository revision.
SVN	<p>A frequently-used abbreviation for Subversion.</p> <p>The name of the Subversion custom protocol used by the “svnserve” repository server.</p>
Switch	Just as “Update-to-revision” changes the time window of a working copy to look at a different point in history, so “Switch” changes the space window of a working copy so that it points to a different part of the repository. It is particularly useful when working on trunk and branches where only a few files differ. You can switch your working copy between the two and only the changed files will be transferred.
Última Revisão	The latest revision of a file or folder in the <i>repository</i> .