

# **TortoiseMerge**

**一种Windows下的比较及合并差异的工具**

**Version 1.5.0 - r12606**

**Stefan Küng  
Lübbe Onken  
Simon Large**

---

# **TortoiseMerge: 一种Windows下的比较及合并差异的工具: Version 1.5.0 - r12606**

由 Stefan Küng、Lübbe Onken和Simon Large

译者: Dongsheng Song (dongsheng.song@gmail.com), xingyu.wang (xingyu.wang@gmail.com), Stan J (thinkernel@yeah.net)

出版方 2008-06-22 23:55:47 +0800

---

---

# 目录

前言 .....	vi
TortoiseMerge 是免费的！ .....	vi
致谢 .....	vi
1. 介绍 .....	1
概述 .....	1
TortoiseMerge 的历史 .....	2
2. Basic Concepts .....	3
Viewing and Merging Differences .....	3
Editing Conflicts .....	3
应用补丁 .....	4
3. 使用 TortoiseMerge .....	5
Viewing Modes .....	5
查看/合并 .....	5
应用补丁 .....	7
按钮及其他控制 .....	8
Line Status Icons .....	9
合并/编辑冲突 .....	10
打开文件 .....	10
查看/合并 .....	11
应用补丁 .....	11
设置 .....	12
常用设置页 .....	12
颜色设置页 .....	13
A. 快捷键 .....	14
快捷键 .....	14
B. TortoiseMerge 操作 .....	15
TortoiseMerge 命令行选项 .....	15
词汇表 .....	16

---

## 插图清单

1.1. 文件冲突 .....	1
2.1. 文件冲突 .....	4
3.1. 单窗口方式 .....	5
3.2. 两窗口方式 .....	6
3.3. 三窗口方式 .....	7
3.4. Patch File List .....	8
3.5. 工具条 .....	8
3.6. 打开文件对话框 .....	11
3.7. 常用设置页 .....	12
3.8. 颜色设置页 .....	13

---

## 表格清单

B.1. 命令行选项列表 .....	15
--------------------	----

# 前言



## Tortoise Merge

When working on a project, either an open-source project where several people review your code, or a commercial project where you sell a component with your source code, there will be people and customers who find bugs or want to improve your project. To do that they send you patches which you have to review and decide if you want to apply them.

TortoiseMerge 可以帮助你完成以下两件事情：

- 查看补丁
- 应用补丁

如果你使用源代码控制系统，并且和其他人共同处理相同文件时，TortoiseMerge也能帮助你挑选出发生了冲突的文件。

## TortoiseMerge 是免费的！

TortoiseMerge 是免费的。你不需要为它支付任何费用就可以任意使用。它是在GPL许可协议下开发的。

TortoiseMerge 是一个开源软件项目。换言之，你可以取得它的全部源代码。你可以访问以下链接得到源代码：<http://tortoisesvn.tigris.org/svn/tortoisesvn/>（用户名：guest，无密码）。最新的版本（我们当前正在开发的）放在 `/trunk/`，已发布版本放在 `/tags/`。

### Donate!

虽然 TortoiseSVN 和 TortoiseMerge 是免费软件，但是你可以通过发送一些补丁和积极参与开发来帮助开发者。此外，你还可以使我们在计算机前度过的漫长时间更加愉快。

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.

## 致谢

Stefan Küng  
TortoiseMerge 的主要编程人

Lübbe Onken  
for the beautiful icons, logo, bug hunting and taking care of the documentation

Tigris 网站设计风格项目  
他们的设计风格被用于本文档

贡献者  
提供了补丁、错误汇报及新的想法，并且在邮件列表上回答了其他人的问题。

赞助者  
他们送给我们的音乐陪伴了我们许多时间



# 第 1 章 介绍

## 概述

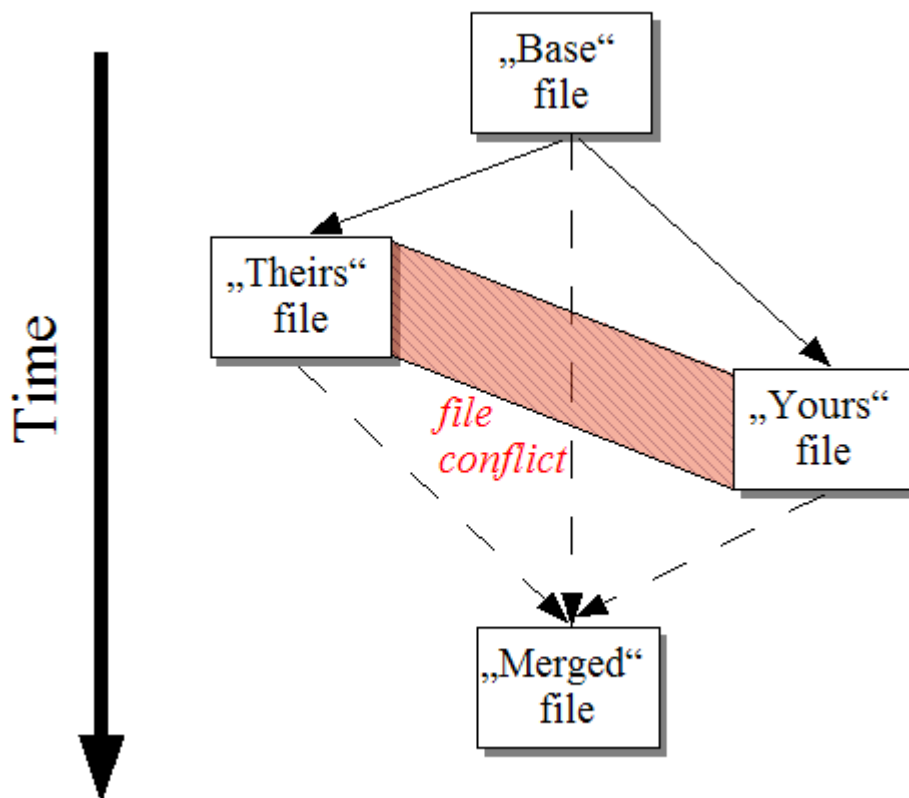
TortoiseMerge 是一个免费软件/开源软件。通过它你可以查看文本文件之间的差异，并合并这些差异；甚至审查和采用这些标准差异文件——通常称之为 补丁。

当处理文本文件时，比如用你所喜欢的编程语言写的源代码，或者 Html/XML 的文档或网站，你经常会碰到需要比较文件的不同版本的情况：有时候你从别人那里得到了一个不同版本的文件，或者有时候干脆就是你想知道你曾经做过什么修改。

如果你使用版本控制系统（如 Subversion）时，当你更新工作副本时你可能会遇到“冲突”。这种情况通常发生在其他人也修改了你当前正在处理的文件的同一部分。当遇到冲突时，你必须手工来解决它——这就是 TortoiseMerge 能帮助你地方。

下图说明了存在冲突的文件之间的关系。

图 1.1. 文件冲突



A conflict can also occur if you try to apply an outdated patch file. In such cases, without TortoiseMerge, you would either have to ask the person you got that patch file from for a current version or try to apply that patch file manually.

But now with TortoiseMerge this isn't necessary anymore - at least not if you're working with a version control system like Subversion. TortoiseMerge parses the patch file for the revision/version the patch was created for and automatically tries to fetch that specific version from your source-control. Then it applies the patch and lets you review both your changes to the file, and theirs. Then you can decide which changes you want to keep.



# TortoiseMerge 的历史

当我们开发 [TortoiseSVN](http://tortoisesvn.tigris.org) [http://tortoisesvn.tigris.org] 时发现，只要用户发送补丁增强这个项目，或者只是修正问题，我们就需要为应用这些补丁付出许多工作量。并且很多时候，这些补丁是过时的，不能使用命令行工具 <http://unxutils.sourceforge.net/> 打补丁，因为我们正在修改这些文件，有时候已经修改了。

这也是最初的 TortoiseSVN 没有实现 TortoiseSVN → 创建补丁 的主要理由：我们期望用户发送给我们整个文件，这样容易打补丁。

So we spent a lot of time, searching the Internet for a GUI tool which could apply patch files, but all we found was that such a tool didn't exist. So to cut a long story short: We decided that if no-one else could provide such a tool we would have to write our own. That's when TortoiseMerge was born.

Since TortoiseSVN already uses the Subversion library, which has a nice diffing engine built in, it was just natural to use this library instead of the GNU diff.

---

## 第 2 章 Basic Concepts

TortoiseMerge has several different uses:

- Viewing the difference between two files and merging changes into one of them, or removing unwanted changes.
- Editing conflicts between local changes and changes in the Subversion repository following an update.
- Applying patch files. This is the reason TortoiseMerge was created :)

### Viewing and Merging Differences

In this mode you are comparing two files. The file in the left pane is considered to be the original file (sometimes referred to as *Theirs*, and the file in the right pane is the modified file (sometimes referred to as *Mine*).

You can make simple line-based changes to the file in the right pane which include:

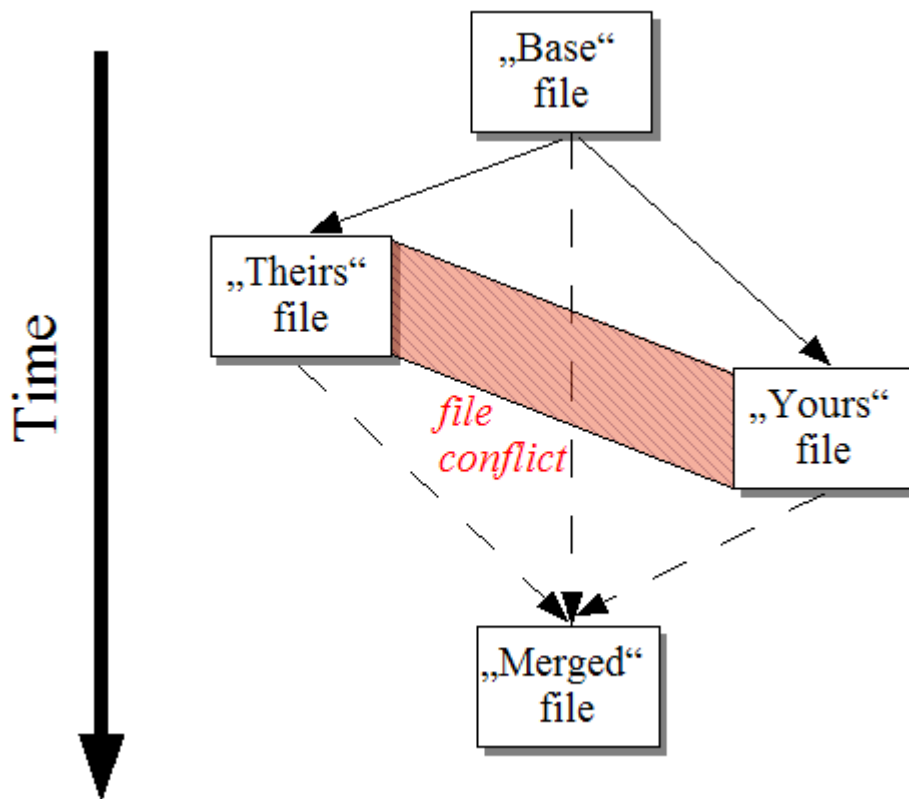
- Reverting changed lines back to the text shown in the left pane.
- Using a combination of both blocks, either *Theirs* before *Mine* or *Mine* before *Theirs*.

You can also edit the file in the right pane just as you would in a text editor. Such lines are marked using a pencil icon. Please note that if you want to make any of the line/block-based changes described above, it is better to do those first since once you start editing the file yourself it becomes impossible for TortoiseMerge to keep track of the relationship to the original files.

### Editing Conflicts

This is sometimes referred to as a three-way merge, and is shown in three panes. However there are actually four files involved. The file which is not shown is the common base file, the last common ancestor of the two files which are now in conflict. The relationships between the three files are explained in the diagram below:

图 2.1. 文件冲突



The base file represents the oldest version of a file, from where You and They start making changes. Mine represents the base file with all the changes you made, and Theirs is the file with all the changes someone else made to the file. The left pane shows the changes in Theirs relative to the base file and the right pane shows the changes in Mine relative to the base file. The bottom pane is the output file which is where you are trying to resolve the conflicts.

In conflict resolution view, you can choose to use blocks from Mine or Theirs or both. But in this case, the changes are shown in the bottom pane.

## 应用补丁

A patch file is a Subversion unified diff file, which contains the information required to apply changes to a set of files. The patch may have been supplied by another developer so that you can see changes he has made and possibly commit them to the repository. Or it may have been generated internally by TortoiseSVN when comparing two folders. This happens when you compare a repository revision with your working copy folder, or if you compare two different repository revisions, or if you compare two different repository paths.

In either case TortoiseMerge will show a small window listing the files included in the patch. By double clicking on one of these files you fetch the relevant file and apply the changes. The left pane shows the original file content and the right pane shows it after the patch is applied.

You can edit the file in the right pane exactly as you would in compare/edit mode.

# 第 3 章 使用 TortoiseMerge

## Viewing Modes

TortoiseMerge has three main viewing modes: one-pane, two-pane and three-pane view. One/two-pane view is used to view changes and three-pane view is used to resolve conflicts.

## 查看/合并

图 3.1. 单窗口方式

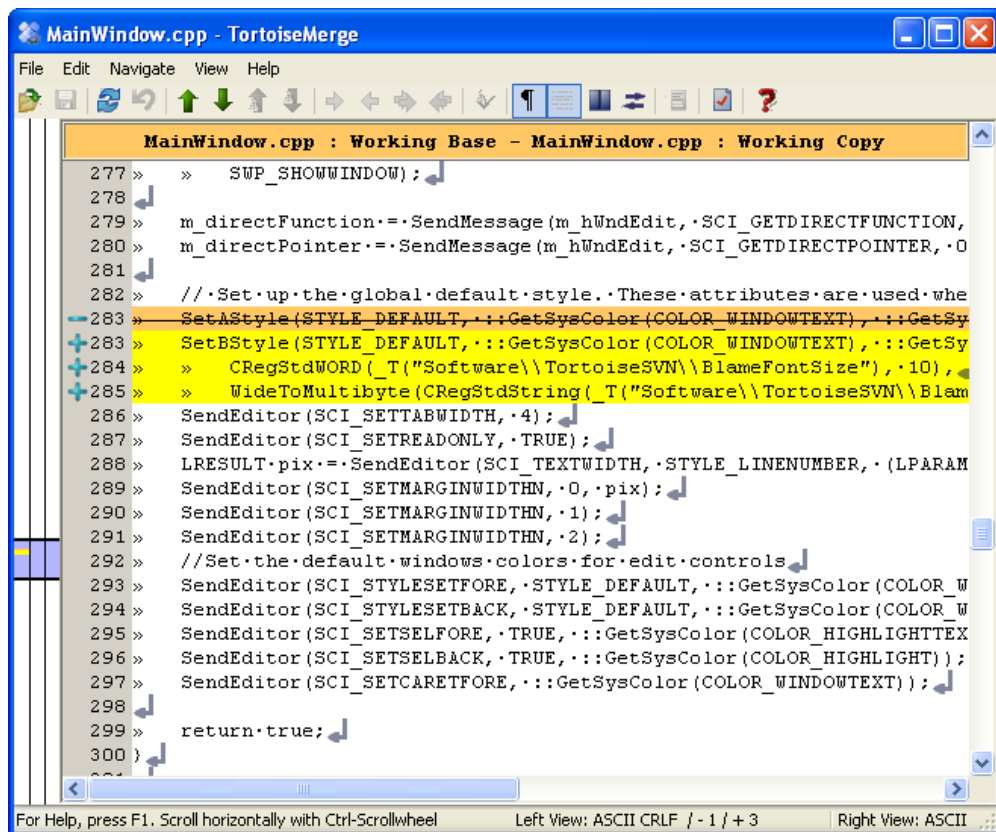
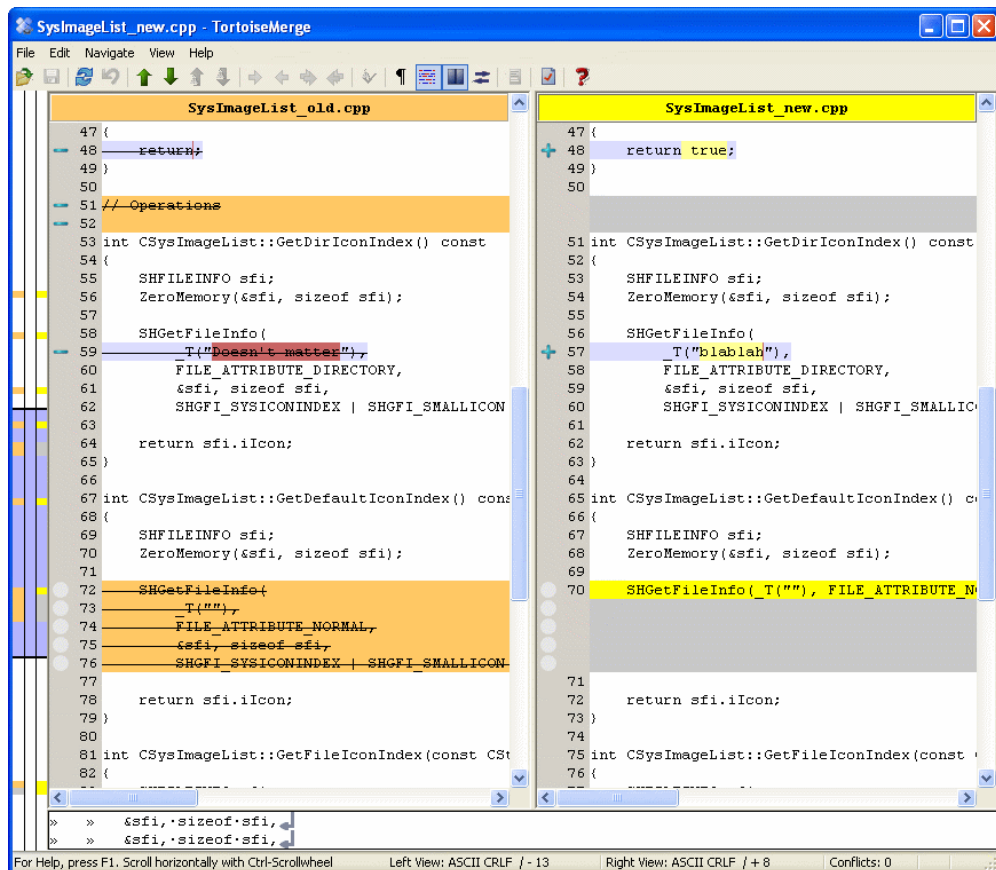


图 3.2. 两窗口方式



两窗口方式有单窗口方式不具备的一些特性：

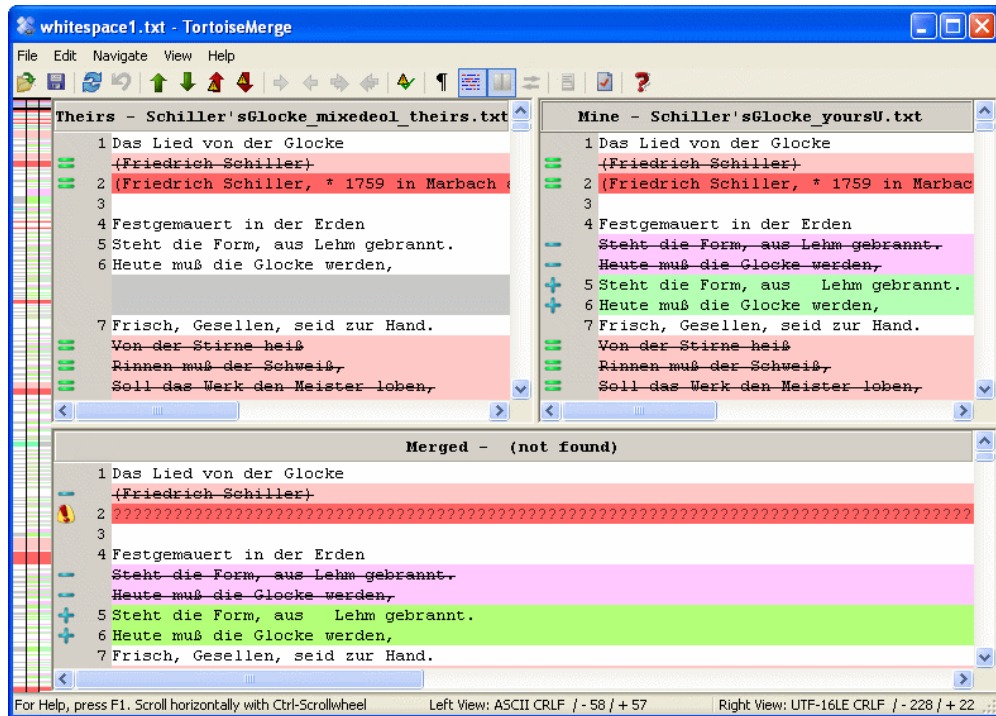
- 用不同的颜色来显示修改行。增加的部分使用浅色，当然你可以配置使用什么颜色。删除部分使用深褐色删除线显示。检查屏幕快照了解它们如何工作。
- 代码重构经常意味着许多空白字符(空格，制表，新行)变化，但是没有实际上的代码改变。举例来说，你可以将一个很长的行拆分为几行，或者将几行合并为一行。

这些改变在视图的左边用白色圆形符号标记。如果你看到了这些白色圆形，就立即知道这里没有实际代码改变，不需要进一步检查这个代码块

- Hand editing of the file in the right pane is possible in two-pane view. Such changes are marked with a pencil symbol.

如果你要比较/合并三个文件，TortoiseMerge 将会使用三窗口方式显示差异。这个视图也用来解决冲突文件。

图 3.3. 三窗口方式



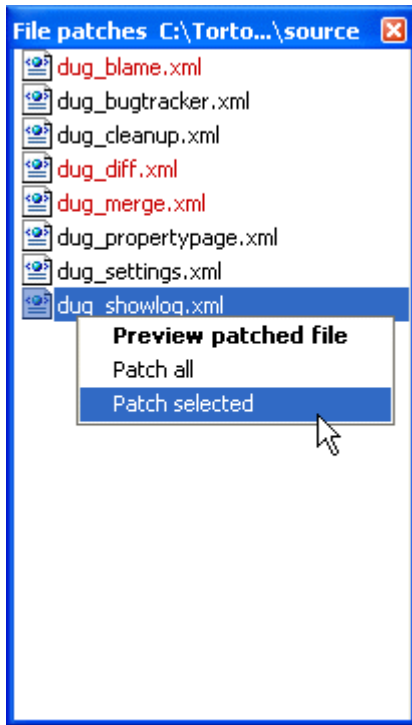
The left pane shows you the differences between Their file and the Base file, while the right pane shows you the differences between Mine file and the Base file. The bottom pane shows you the result of merging Base, Mine and Theirs with possible conflicts.

If you hover the mouse over the pane title, a tooltip will show the filenames used for the diff in each pane.

## 应用补丁

After TortoiseMerge parses the patch file it will show you a small window with all the files which have changes according to the patch file.

图 3.4. Patch File List



If the filename is shown in black, then the patch can be applied without any problems. That means the file is not outdated according to the patch. However, if the filename is shown in red, then the patch can't get applied directly because you already changed that file.

But don't worry. If you're working with Subversion you can retrieve an older version of the file and resolve the conflict. In fact, TortoiseMerge automatically tries to get that older version of the file and then shows you the diff between Base (the older version of the file), Mine (the file as it is currently in your working copy) and Theirs (the older version of the file with the patch already applied). Then you can resolve possible conflicts manually.

The patch file window has a context menu which allows you to preview the effect of the patch on the current file (apply without saving), to apply and save the changes for the selected file, or to apply and save changes to all files in the list. The double-click action is to preview.

依赖于 TortoiseMerge 能直接使用补丁，或者先取出老版本文件，它显示双窗口视图(可以直接打补丁)，或者显示三窗口视图(取出了老版本文件)。

## 按钮及其他控制

图 3.5. 工具条



下列菜单显示在工具条里。可以方便地使用多数常见功能，它们分别是：

- 打开 除了打开文件对话框，还能是什么？
- 保存 保存你所做的任何修改。
- 重载 放弃你在 TortoiseMerge 中所做的任何修改，重新载入文件。
- 撤销实现多级撤销，允许你放弃在 TortoiseMerge 中所做的任何修改。

- 上一个/下一个差异 直接跳转到文件中的上一个/下一个不同的地方。它可以快速地跳过文件未变化过的部分。当你处于双窗口方式时，无论是否鼠标划过该区域，在屏幕底部的细节状态栏都将显示首行的不同。
- 上一个/下一个冲突 直接跳转到文件中的上一个/下一个需要解决的冲突。
- Use Theirs/Use Mine selects whether to use a block from the left or the right pane. Refer to the next section for more information about editing changes and conflicts.
- Theirs before Mine/Mine before Theirs is used when you want to combine changes in a conflicted file. Refer to the next section for more information about editing changes and conflicts.
- 标记为已解决 当你解决了文件中的全部冲突并保存了修改后，你可以在 TortoiseMerge 中标记为已解决冲突，而不用返回资源管理器去标记它们。
- 显示空白字符 切换是否使用符号显示空格和制表符，以便你能注意到空白字符的变化。
- Inline Diff Word-wise toggles the method used to display inline diffs. In word-wise mode, when a character is changed, the whole word is highlighted. In char-wise mode, each individual changed character is highlighted.

Word-wise is the default and in most cases it works better. For example if you change a variable name, the whole name is highlighted in word-wise mode, whereas in char-wise mode you tend to get a mess of added and deleted characters which doesn't make much sense. However, some content does not resolve neatly into words, for example a hex file, and in that case, char-wise display works better.

- 切换窗口方式 在单窗口方式和双窗口方式间进行切换。在三窗口方式时无效。
- 交换左右窗口 在双窗口方式中，互换左右窗口的内容。当你比较两个不同分支中的两个文件时，而不是比较同一个文件的两个修订版本时，TortoiseSVN 并不知道哪个文件应该放在左边的窗口作为“原始/基础版本”。如果放错了，为了便于查看，你能够互换它们。
- 显示/隐藏补丁列表 当 TortoiseMerge 有补丁列表时，这个按钮允许你隐藏它(于是你可以看到两个窗口)，或者显示它(于是你可以复审文件列表)。
- 设置 弹出 TortoiseMerge 的设置对话框。

窗口的左边是定位拉杆。它提供了对文件改变的位置的快速可视化引用。这个拉杆有三列。左列引用左边的窗口，右列引用右边的窗口，中间列引用底部窗口(如果存在)。在单窗口模式下，只使用了左列。定位拉杆也可以作为滑杆使用，让所有窗口同时滑动。

底部窗口下面是状态栏。它显示了##与##增加和删除的行数，以及尚存的未解决冲突数目。你还可以看到使用的行结束符，编码(ASCII，UTF-8等)，以及在UTF-8编码的文件中是否有字节顺序标记(BOM)。当你的鼠标逗留在控件上的时候，状态栏的左边会显示提示。

## Line Status Icons

Changed lines are marked with an icon to indicate what type of change has occurred.



A line has been added.



A line has been removed.



A change has been undone by reverting to the original content of the line.





This line contains only whitespace changes. Where several consecutive lines are marked, the paragraph may have been re-wrapped, causing words to shuffle onto adjacent lines.



A line has been edited by hand, using TortoiseMerge as a text editor.



This line is in conflict.



This line is conflicted, but the effect is hidden by whitespace or line-ending settings.

## 合并/编辑冲突

TortoiseMerge 不仅显示了你文件间的不同，而且还能让你解决存在的冲突或者采用修改。

如果你在双窗口模式，你只能编辑右窗口中的文件(##)。要应用左边的文件修改(###)，在修改行上右击，选择上下文菜单 → 使用“他人的”文本块。于是左边文件的修改就增加到右边的文件。

有时候，你实际上需要双方的文本块，上下文菜单 → 使用双方的文本块 (这个先出现) 和 上下文菜单 → 使用双方的文本块 (这个后出现)可以满足你的要求。

You can also edit the output file just as you would in a text editor. Such lines are marked using a pencil icon. Please note that if you want to make any of the line/block-based changes described above, it is better to do those first since once you start editing the file yourself it becomes impossible for TortoiseMerge to keep track of the relationship to the original files.

如果你在三窗口模式(有时称为合并视图)，你只能编辑底部窗口中的文件(####)。象在两窗口模式那样，你可以在冲突行上右击，或者选择上下文菜单 → 使用“他人的”文本块，或者上下文菜单 → 使用“我的”文本块。此外，如果你要两者的文本块，可以选择上下文菜单 → 在“他人的”前面使用“我的”文本块，或者 上下文菜单 → 在“我的”前面使用“他人的”文本块。通过你选择的命令，改变结果在####文件。

Sometimes a file will be marked as conflicted in Subversion, yet when you view it using TortoiseMerge there are no conflicts shown. This may be due to the whitespace handling you have chosen. If you choose to ignore line-endings or whitespace changes, those lines will be marked using the Conflict-Ignored icon. To resolve the conflict you still need to pick which version you want to use.

有时冲突很难通过简单选择###和##来解决，需要手工编辑。TortoiseMerge 允许你先在可视化模式做所有容易的合并，然后保存文件，将困难的合并用手工解决。

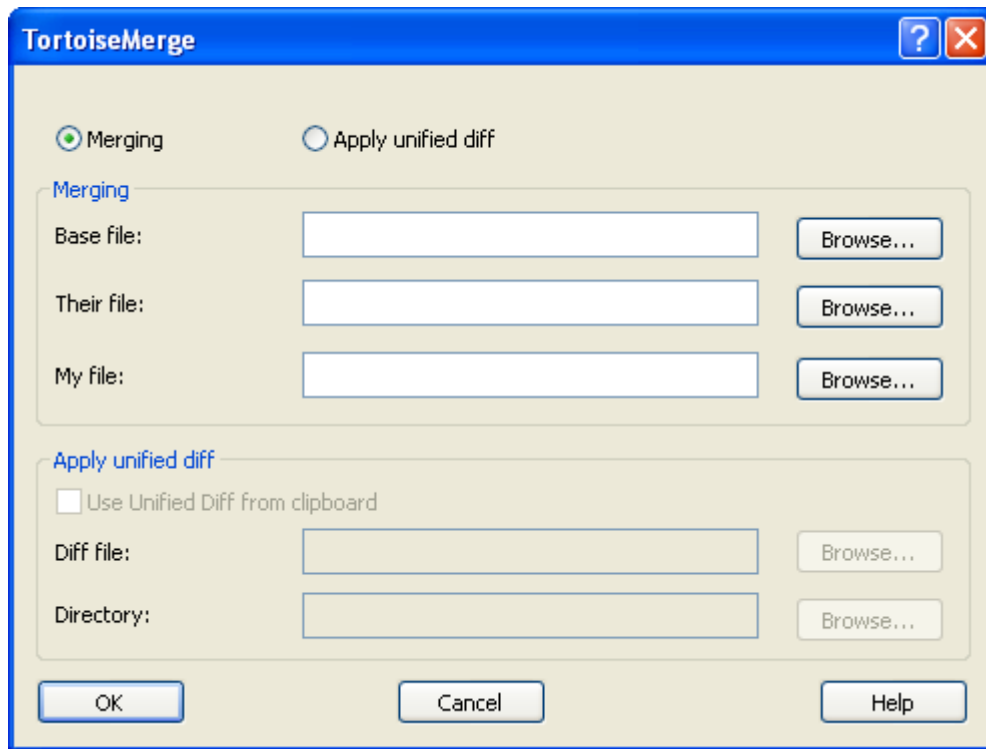
### 重要

注意：如果你在同一文件重新使用 TortoiseMerge，你对工作版本做的任何修改，无论是 TortoiseMerge，还是手工编辑，都将被丢弃，文件恢复到开始冲突编辑时的状态。

## 打开文件

当不使用任何命令行参数启动 TortoiseMerge 时，你必须使用文件 → 打开来手工打开文件。

图 3.6. 打开文件对话框



The first thing you have to do is to decide whether you just want to compare/merge files, or whether you want to apply a patch file. Depending on what you choose, the corresponding edit boxes and browse buttons are activated.

## 查看/合并

If you want to compare / merge files you have to set at least two of the three possible paths for `Base`, `Mine` and `Theirs`. If you select only two files, then TortoiseMerge will show you the differences between those two files, either in a two pane view or in a one pane view.

If you want to merge three files, TortoiseMerge will show you the differences in a three pane view. This view is generally used if you need to resolve conflicted files. The output file is not named in this case and you will have to use the File → Save As... to save the results.

## 应用补丁

If you want to apply a patch file you have to set both the path to the patch file itself and the path to the folder where the patch file should be applied.

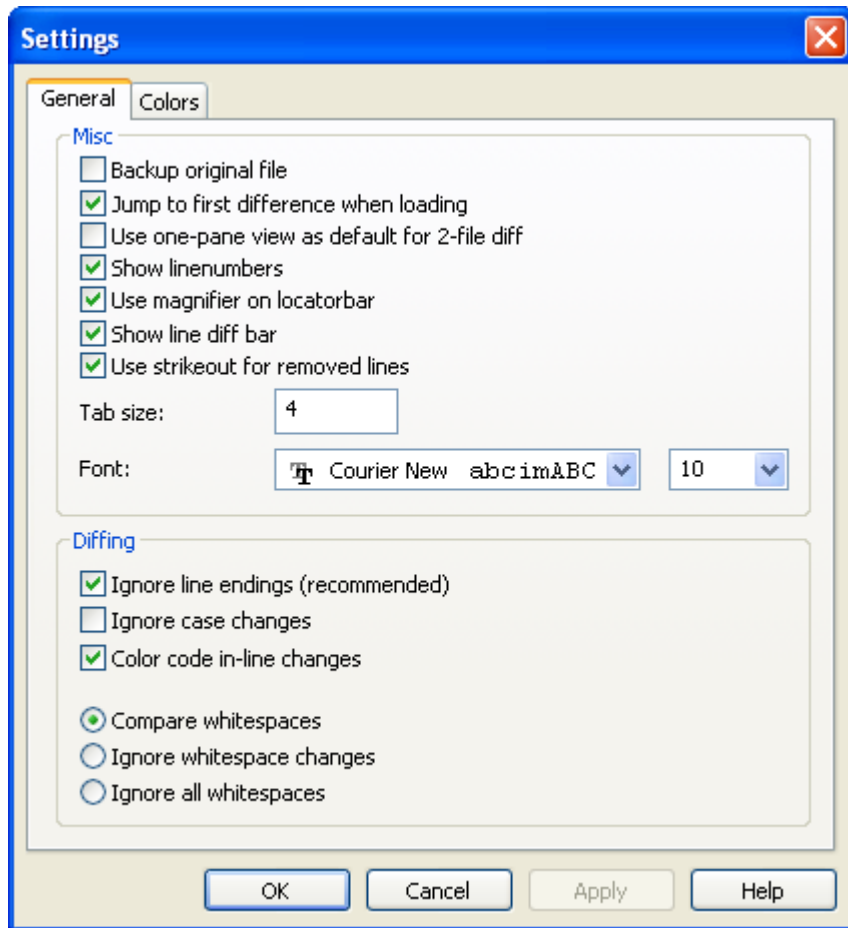
### 小心

Currently, only patch files in Unified Diff Format are supported. And only files which were generated from a Subversion working copy. Other formats like patch files generated by CVS are currently not supported!

# 设置

## 常用设置页

图 3.7. 常用设置页



多数选项是不需要过多解释，只需要稍微明确一下。

备份原始文件 在保存修改后的版本时，先将工作拷贝中的原始文件重命名为filename.bak。

Ignore line endings hides changes which are due solely to difference in line-end style.

忽略大小写变化 隐藏仅文本大小写变化的差异。这对于像 Visual Basic 等忽略变量大小写的应用程序比较有用。

为改变的嵌入代码着色在主窗口，使用与行差异拉杆相同的方案显示代码改变的细节。修改很多的行不会这样显示，因为它会让人糊涂。这时会用替换方式显示(旧行删除，新行增加)。

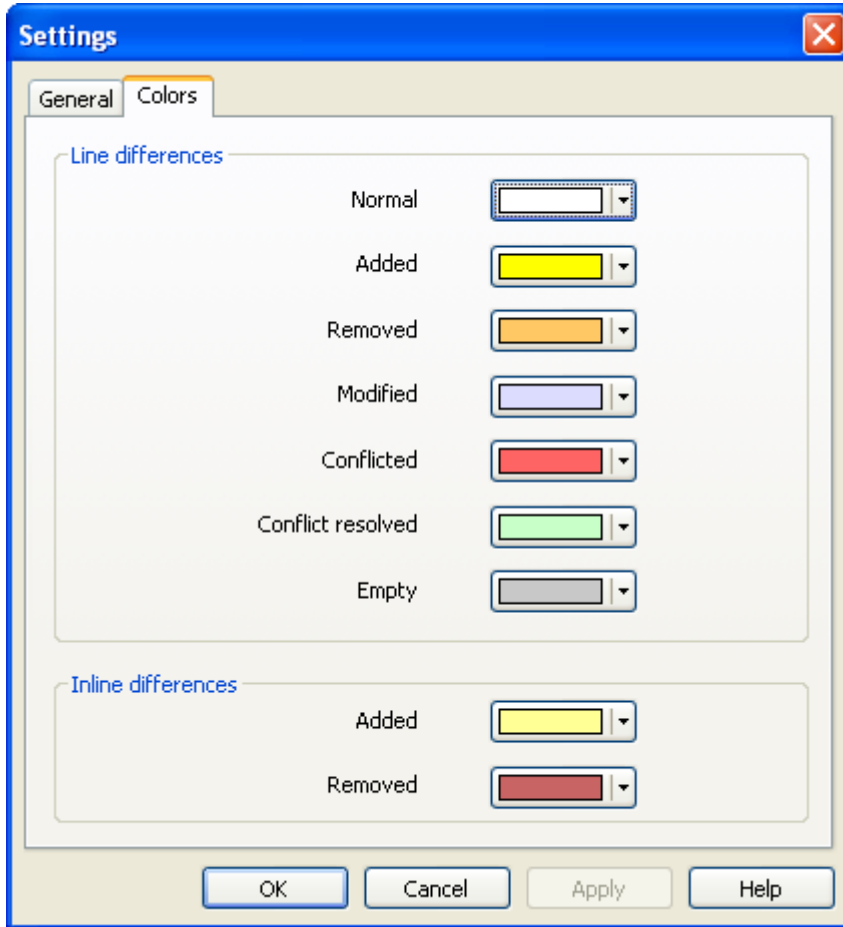
比较空白字符 将全部缩进变化，或行内空白字符的变化按照添加/删除的行显示（即 Subversion 处理变化的方式）。

忽略空白字符变化 隐藏仅仅只是空白字符数量和类型的变化。例如，缩进，制表符变成了空格。在以前没有空白字符的地方增加空白字符，或者完全删除空白字符，仍旧视为变化。

忽略全部空白字符 隐藏所有空白字符的变化。

## 颜色设置页

图 3.8. 颜色设置页



这个对话框允许你选择一个背景颜色代码，它们用于高亮被修改的行。

### 无变化

这些行都没有改变过，或改变被隐藏了。

### 新增

这些行是新增加的。

### 删除

这些行被删除了。

### 已修改

相对小的行改变，用嵌入差异显示。未改变部分着色。改变的部分用嵌入的增加和删除形式展现，其颜色下述。如果未启用着色改变的代码，不会使用此颜色，改变的行用替换形式显示。

### 冲突

两个文件都修改了这些行。

### 空白

在另一个面板中增加了一些行，且这个面板中没有对应的行。

### 增加的文本

当显示嵌入差异时，用此颜色显示增加的文本。

### 删除的文本

当显示嵌入差异时，用此颜色显示删除的文本。

---

# 附录 A. 快捷键

列出所有的快捷键和命令。

## 快捷键

- Ctrl-Q  
退出程序
- Ctrl-C  
Copy the selected text to the clipboard
- Ctrl-F  
打开文本查找对话框
- Ctrl-O  
打开需要比较/合并的文件
- Ctrl-S  
保存改变
- Ctrl-Shift-S  
另存为...
- F7  
跳转到下一个不同点
- Shift-F7  
跳转到上一个不同点
- F8  
跳转到下一个不同点
- Shift-F8  
跳转到上一个不同点
- Ctrl-D  
切换单窗口方式或双窗口方式
- Ctrl-R  
重新载入文件并放弃所做的任何改变
- Ctrl-T  
切换是否显示空白字符
- Ctrl-mousewheel  
滚动左右两边的显示

# 附录 B. TortoiseMerge 操作

TortoiseMerge 可以通过命令行选项指定参数运行，而不必非通过打开对话框来选择文件。这便于其它程序调用 TortoiseMerge。

## TortoiseMerge 命令行选项

多数选项需要路径及其它信息。这种情况下，选项后用“:”跟一个字符串或路径。例如：

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

表 B.1. 命令行选项列表

命令	描述
?	弹出一个显示常用命令行参数的对话框。
/help	同 ? 一样。
基础版本文件	指定三路比较的文件 基础版本文件。这是所有参与比较文件的祖先，尽管它不在单独的窗口显示。对于两路比较，它是左边的文件。
基础版本文件	基础版本文件名称。它代替文件路径在窗口标题栏显示。在三窗口方式时，它出现在窗口标题的提示中。
他人的文件	指定用于三窗口方式的 他人的文件，显示在左窗口。
他人的文件	他人的文件名称。代替文件路径在窗口标题栏显示。
我的文件	指定用于三窗口方式的 我的文件，它显示在右窗口。对于双面板方式，是指右边的文件。
我的文件	我的文件名称。代替文件路径在窗口标题栏显示。
合并后的文件	指定用于三窗口方式的 合并后的文件。它是保存合并/解决冲突的结果的文件路径。如果没有指定，TortoiseMerge 将询问用户保存结果的路径。
合并后的文件	合并结果的文件名称。显示在窗口标题显示路径的地方。
/patchpath	The path where a patch should be applied to. If you don't set this path, then TortoiseMerge will try to find the path itself to match the paths in the patch file, but that can take 很长.
/patchoriginal	对应补丁的原始文件。用于窗口标题。
/patchpatched	对应补丁的文件名称。用于窗口标题。
差异	将要应用补丁的目录
/oneway	强制 TortoiseMerge 以单窗口视图启动，忽略用户的配置。
/reversedpatch	左右交换两个需要比较差异的文件。
/createunifieddiff	Creates a unified diff file (patch file) of two files specified with /origfile:"path_to_original_file" and /modifiedfile:"path_to_modified_file". The target path is set with /outfile:"path_to_resulting_patchfile". If /outfile is not set, a file save dialog is shown so the user can choose the location to save the patch file. Note: If /createunifieddiff is set, all other parameters are ignored.

为了兼容其它差异比较程序，你也可以在命令行提供简单的文件名。这种简单命令行格式是

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

如果给出两个文件，它们会互相比。如果给出三个文件，那么第一个文件视为基础文件，其它两个文件采用三路比较算法。

---

# 词汇表

增加	用于增加文件或目录到你的工作副本中的 Subversion 命令。当你提交时，这些新项目会增加到版本库中。
基础版本	工作拷贝中的文件或文件夹的当前基础版本。它是文件或文件夹上次检出、更新或提交后的版本。通常基础版本与最新版本并不相同。
追溯	这个命令只能用于文本文件，它显示每行由哪个版本最后修改，以及修改的作者。我们的 GUI 实现称之为 TortoiseBlame，当你将鼠标停靠在版本号上时，还会显示提交的日期/时间，和日志。
BDB	伯克利 DB ( BDB )。 一个经过广泛测试的版本库的数据库后端，不能使用网络文件系统(例如 NFS 或 SMBFS)。是 1.2 版本及以前的版本库之默认后端。
分支	一个经常被开发者在特定时期跟踪两个独立路径时使用的术语，它描述了在版本控制系统中发生的事情。你可以从主开发线创建分支，从而开发新特性，不会使主开发线陷入不稳定状态。或者你创建一个稳定发布版的分支，仅用于修复问题，新的开发位于不稳定的主开发线。在 Subversion 中，分支用“廉价拷贝”实现。
检出	此 Subversion 命令从版本库下载受版本控制的文件，在空目录创建本地工作副本。
清理	To quote from the Subversion book: “ Recursively clean up the working copy, removing locks and resuming unfinished operations. If you ever get a working copy locked error, run this command to remove stale locks and get your working copy into a usable state again. ” Note that in this context lock refers to local filesystem locking, not repository locking.
提交	此 Subversion 命令用于将你的本地修改传递到版本库中，创建新版本。
冲突	当仓库的改变合并到本地修改时，有时会在相同的行修改。在这种情况下，Subversion 不能自动决定采用哪个版本，文件就处于冲突状态。你必须手工编辑文件，在你提交之前解决冲突。
复制	在 Subversion 版本库中，你可以创建单个文件或一个目录树的副本。此操作用“廉价拷贝”实现，它有些象符号连接到原始文件，几乎不耗费空间。创建的副本保留了历史，所以你可以跟踪副本创建之前的修改。
删除	当你删除一个受版本控制的项(并且提交了修改)，在那么在此提交版本后，它就消失于版本库中。当然，它仍旧包含在版本库的早期版本中，所以你还可以存取它。如果需要，你可以复制一个已删除项，包含历史的彻底“复活”它。
差异	“显示差异”的简写。当你想精确的察看变化时很有用。
导出	此命令产生一个受版本控制的文件夹之副本，很像工作副本，只是没有本地的 .svn 文件夹。
FSFS	FS 文件系统。一个专用于 Subversion 版本库的文件系统后端，可以使用网络文件系统(例如 NFS 或 SMBFS)。是 1.2 版本及其后的版本库之默认后端。
GPO	组策略对象
最新版本	版本库 中的文件或文件夹的最新版本。
导入	用于一次导入整个文件夹(包含子目录)的 Subversion 命令。
加锁	当你加锁一个版本控制项目，除了在发出加锁命令的工作目录，它在版本库中被标记为不可提交的。
日志	显示文件或目录的版本历史。也称为“历史”。
历史	显示文件或目录的版本历史。也称为“日志”。

合并	<p>将版本库中的修改增加到你的工作副本，不破坏本地的任何改变。有时这些修改不能自动协调，工作副本的状态称为冲突。</p> <p>当你更新工作副本时，自动发生合并。你可以使用 TortoiseSVN 的合并命令合并其它分支的指定修改。</p>
补丁	<p>如果工作副本只修改了文本文件，就可以使用 Subversion 的差异命令产生一个标准差异格式文件。这种文件类型通常称为“补丁”，可以应用到其他工作副本。没有提交权的人可以在修改后，发送补丁文件给一个认可的提交者来提交。或者不能确定此修改是否恰当，你可以发送补丁给其他人评审。</p>
属性	<p>在版本控制文件和目录之外，Subversion 还允许你增加版本控制元数据 - 称为“属性” - 到你的版本控制文件和目录。每个属性都有名字和值，很像注册表键。Subversion 有内部使用的专用属性，例如 <code>svn:eol-style</code>。TortoiseSVN 也是如此，例如 <code>tsvn:logminsize</code>。你可以增加自己的属性，允许选择任意名称和取值。</p>
重新定位	<p>如果你的版本库移动了，或许是在服务器上将它移到其它目录，或者服务器域名改变了，你需要“重新定位”工作目录，让其版本库地址指向新的位置。</p> <p>注意：你只能在同一版本库中，工作副本引用相同的位置，仅版本库移动的情况下使用这个命令。在其它情况下，或许你需要用“切换”命令。</p>
版本库	<p>版本库是存储和维护数据的中央部分。一个版本库的数据库或文件可以放在通过网络可访问的位置，或者存放在一个不需要通过网络就可以访问的地方。</p>
解决	<p>当工作副本中的文件在合并后处于冲突状态时，这些冲突使用使用编辑器(或许是 TortoiseMerge)手工清理。这个步骤称为“解决冲突”。然后你就能标记这些冲突文件已经解决，允许提交这些文件。</p>
恢复	<p>当最后更新工作目录时，Subversion 为每个文件保留了一份本地“原始”副本。如果你做了修改，决定撤销，可以使用“恢复”命令得到原始副本。</p>
版本	<p>在你每次提交改变集时，就在版本库创建了一个新“版本”。每个版本都表示版本库历史上的确定状态点。如果你想回溯历史，可以使用版本 N 来检查版本库。</p> <p>换个说法，版本可以引用它创建时产生的改变集。</p>
版本属性(revprop)	<p>正如文件可以有属性，版本库中的版本也有。当创建版本时，会自动创建版本属性，即：<code>svn:date</code> <code>svn:author</code> <code>svn:log</code>，分别描述了提交日期/时间，提交者，日志信息。这些信息可以编辑，但是没有版本，所以任何修改都是永久的，不能撤销。</p>
SVN	<p>一个友好的 Subversion 缩写。</p> <p>被“<code>svnserve</code>”版本库服务器使用的专用协议名称。</p>
切换	<p>就像“更新到版本”改变了工作副本的时间窗口，可以察看历史上的不同时间点一样，“切换”改变了工作副本的空间窗口，它可以指向版本库中的不同部分。当两个开发分支仅有少数文件不同时，它非常有用。你可以在两者之间切换工作副本，只有修改的文件内容被传送。</p>
更新	<p>此命令从 Subversion 版本库下载最新的修改到你的工作副本，合并任何本地修改。</p>
工作副本	<p>这是一个本地“沙箱”，你在受版本控制的文件上工作的区域，它通常位于你的本地硬盘上。你可以用“检出”从版本库创建工作副本，使用“提交”将你的改变传回版本库。</p>