

TortoiseMerge

针对 Windows 平台的比较与合并工具

版本 1.6.16

Stefan Küng
Lübbe Onken
Simon Large

目录

前言	v
1. TortoiseMerge 是免费的!	v
2. 致谢	v
1. 介绍	1
1.1. 概述	1
1.2. TortoiseMerge 的历史	1
2. 基本概念	3
2.1. 查看与合并差异	3
2.2. 编辑冲突	3
2.3. 应用补丁	4
3. 使用 TortoiseMerge	5
3.1. 查看方式	5
3.1.1. 查看/合并	5
3.1.2. 应用补丁	7
3.2. 按钮及其他控制	8
3.3. 行状态图标	9
3.4. 合并/编辑冲突	10
3.5. 打开文件	10
3.5.1. 查看/合并	11
3.5.2. 应用补丁	11
3.6. 设置	12
3.6.1. 常用设置页	12
3.6.2. 颜色设置页	13
A. 快捷键	15
A.1. 快捷键	15
B. TortoiseMerge 操作	16
B.1. TortoiseMerge 命令行选项	16
术语表	18

插图清单

1.1.	文件冲突	1
2.1.	文件冲突	3
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.	补丁文件列表	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	16
---------------------	----

前言



Tortoise Merge

当开发一个项目的时候，无论是只有几个人审查你的代码的开源项目还是包括源代码的商业软件，总有一些人或客户会发现错误或希望改进你的项目。因此，他们发送给你补丁，你需要来审查并决定是否使用它们。

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。

1. □ 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。这意味着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://code.google.com/p/tortoisesvn/source/browse/>。你会被提示输入用户名称和密码。用户名称是 `guest`，不能填写密码。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/`。

捐赠！

尽管TSVN和TortoiseMerge是免费的，你也可以通过提交补丁和在项目开发中扮演一些积极角色，来支持我们的开发人员。你同样也可以给点鼓励，它们会让每天在计算机前没日没夜拼命工作的我们感到非常的振奋。

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. □ 致谢

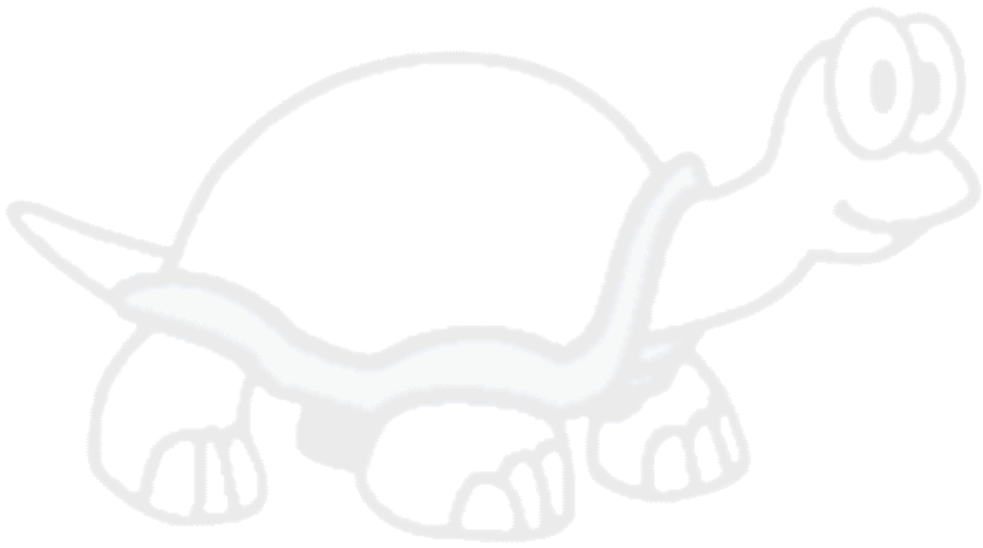
Stefan Küng
TortoiseMerge 的主要编程人

Lübbe Onken
制作了漂亮的图标、LOGO，检查错误及维护文档

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

1.1. 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。

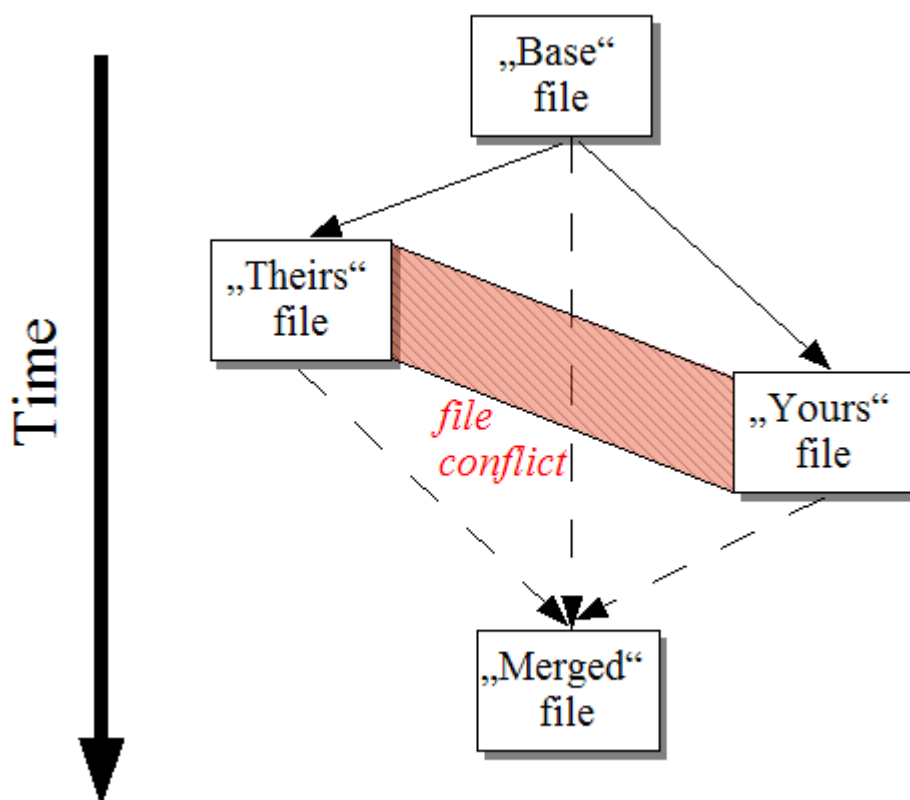


图 1.1. 文件冲突

应用一个过时的补丁时也可能出现冲突。在这种情况下，没有 TortoiseMerge 时，你要么向补丁提供者索取一个当前版本的补丁，要么手工的应用该补丁。

但是现在有了 TortoiseMerge，它就不再是必需的了——至少当你使用类似 Subversion 这样的版本控制系统时不是必需的。TortoiseMerge 解析补丁文件创建的修订版/版本号，并自动尝试从你的版本控制系统中获取对应的版本。然后应用该补丁，让你对比文件的原始内容及其修改后的内容。因此你能决定是否接受那些修改。

1.2. TortoiseMerge 的历史

当我们开发 TortoiseSVN [<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。

于是我们花费了大量时间，在互联网搜索打补丁的图形工具，但是没有发现。简而言之：既然没有人提供这样的工具，我们决定自行编写。就在那时，TortoiseMerge 出世了。

因为 TortoiseSVN 已经使用了 Subversion 库，它内置了一个很好的差异比较工具。很自然的，我们使用这个库代替 GNU diff。

第 2 章 基本概念

TortoiseMerge 有几个不同的用途:

- 察看两个文件的差异, 合并它们, 或者删除不想要的修改。
- 更新后, 编辑本地修改和版本库修改之间的冲突。
- 应用补丁。这是创建 TortoiseMerge 的原因 :)

2.1. 查看与合并差异

此模式用于比较两个文件。左窗格中的文件认为是原始文件(有时称为 他人的, 右窗格中的文件是已修改的文件(有时称为 我的)。

你可以在右窗格的文件中做简单的基于行的修改, 包含:

- 撤销这些修改行, 恢复到左窗格的状态。
- 使用所有组合, 不是他人的在我的之前, 就是我的在他的之前。

你可以象在文本编辑器中一样编辑在右边窗格中的文件。这些被修改的行用铅笔图标标记。请注意, 如果你要做上述基于行/块的修改, 最好在开始编辑文件之前完成。因为在你开始编辑之后, TortoiseMerge 已经不能跟踪它与原始文件的关联了。

2.2. 编辑冲突

这有时称为三路合并, 用三窗格显示。然而它实际上涉及到四个文件。没有显示的文件是公共基础文件, 是现在冲突的两个文件的最后祖先。这三个文件的关系用在下图解释:

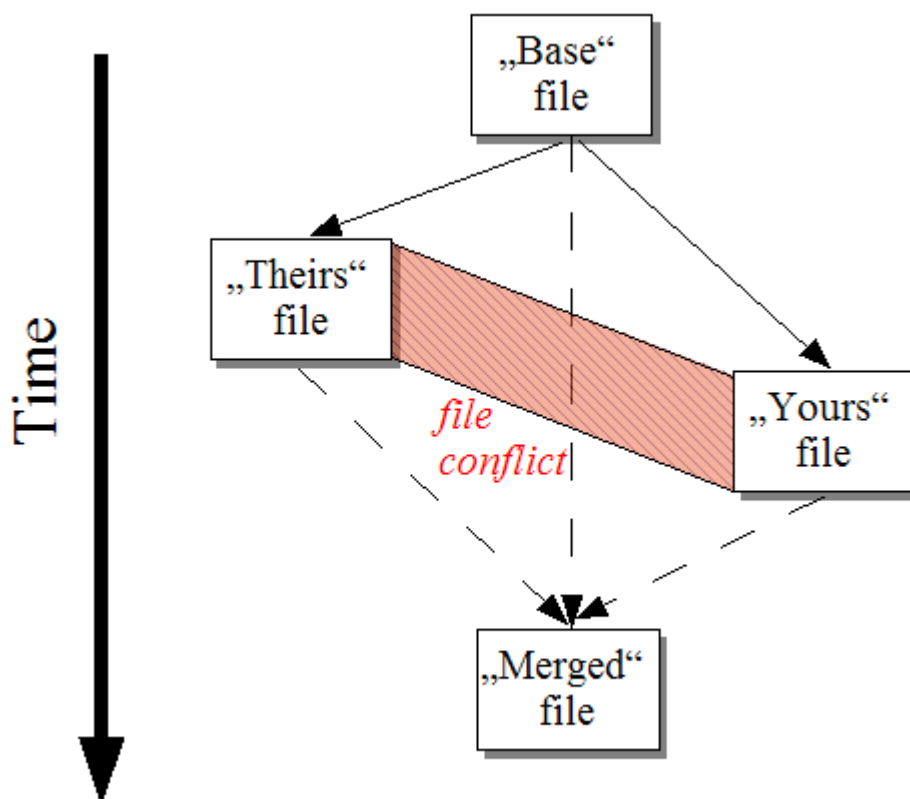


图 2.1. 文件冲突

基础文件是最旧的版本，你的文件，他人的文件，都是从这个文件开始修改的。我的文件表示你对基础文件的修改结果，他人的文件是其他人对基础文件的修改结果。左边的窗格显示他人对基础文件的修改，右边的窗格显示我对基础文件的修改。底部窗格是你尝试解决冲突的输出文件。

在解决冲突视图，你可以选择使用我的块，或者他人的块，或者两者兼有。此时，修改在底部窗格显示。

2.3. 应用补丁

补丁文件是 Subversion 统一差异格式，它包含了使用补丁的信息。补丁可能是其他人提供的，所以你可以察看其修改，或许将它提交到版本库。或者补丁是 TortoiseSVN 比较两个目录生成的。例如你比较版本库中的版本与工作副本，或者比较两个版本库版本，或者比较两个不同的版本库路径。

无论如何，TortoiseMerge 都会用小窗口列出补丁中的文件名称。双击其中一个文件，就会对其打补丁。左窗格显示原始文件内容，右窗格显示打补丁之后的文件内容。

你可以象在比较/编辑方式一样，编辑右窗格中的文件。

第 3 章 使用 TortoiseMerge

3.1. 查看方式

TortoiseMerge 有三种主要视图方式：单窗格，两窗格和三窗格。单窗格/两窗格视图用于察看修改，三窗格用来解决冲突。

3.1.1. 查看/合并

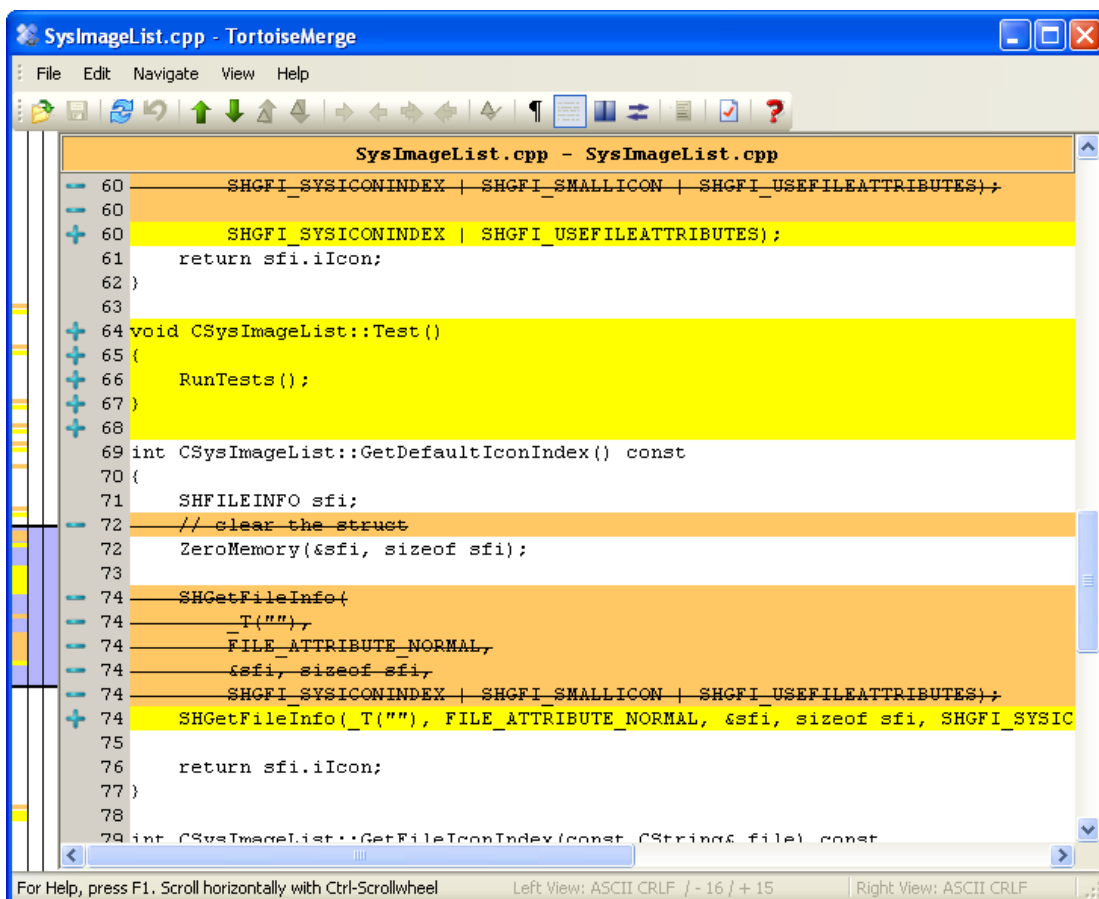


图 3.1. 单窗口方式

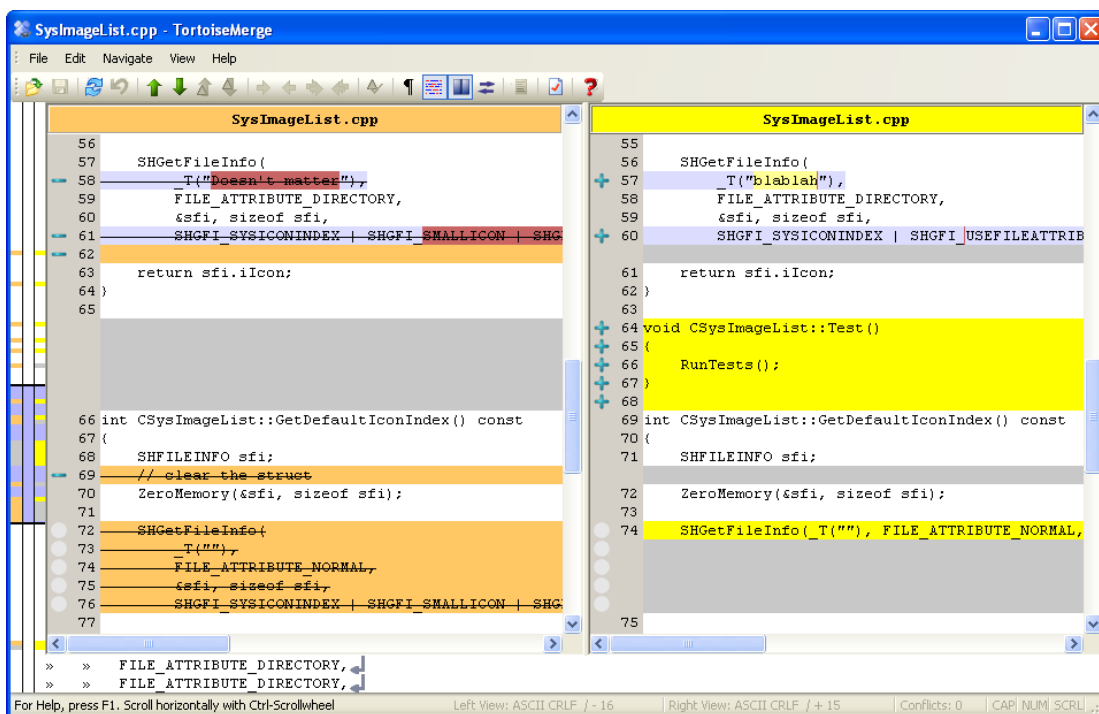


图 3.2. 两窗口方式

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

- 在两窗格视图中, 可以在右边的窗格编辑文件。这些修改用铅笔符号标记。

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。

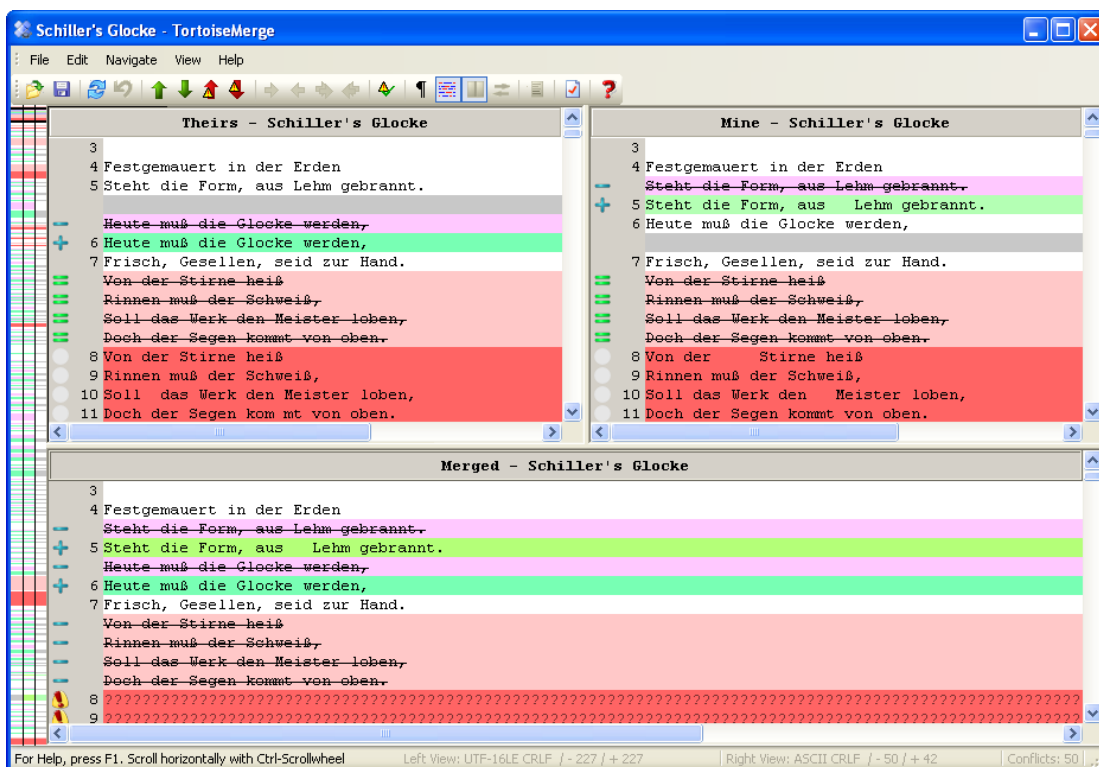


图 3.3. 三窗口方式

左窗口显示他人的文件与基础文件的差异，右窗口显示我的文件与基础文件的差异。底部窗口显示基础文件，我的文件，与 他人的文件 的合并结果，以及可能的冲突。

如果你的鼠标在窗格的标题栏逗留，会有工具提示显示此窗格用于比较的文件名称。

3.1.2. 应用补丁

如果 TortoiseMerge 能解析补丁文件，它会显示一个小窗口，给出此补丁修改的文件列表。

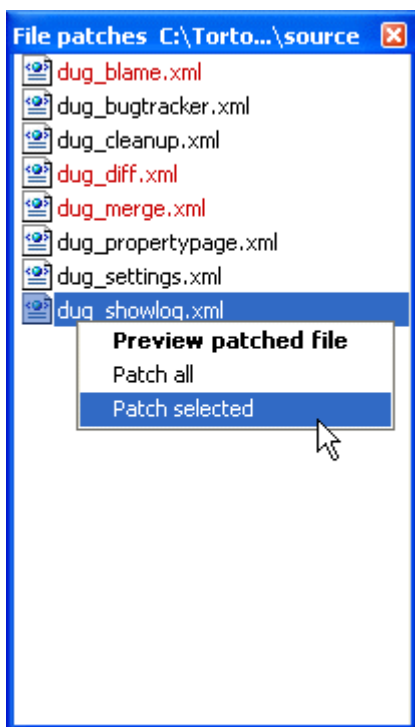


图 3.4. 补丁文件列表

如果文件名称用黑色显示，那么这个补丁可以没有任何问题的应用。它表示此补丁针对的文件没有过时。如果文件名称用红色显示，那么这个补丁不能直接应用，因为你的以往修改造成了这个文件出现冲突。

但是别担心。如果你使用 Subversion，可以取出这个文件的老版本来解决冲突。实际上，TortoiseMerge 自动尝试取出老版本，给出基础版本(此文件的老版本)，我的版本(你的工作副本中的文件)，和他人的版本(已经打补丁的老版本)之间的差异。然后你可以手工解决冲突。

补丁文件窗口有一个上下文菜单，允许你预览在当前文件打补丁的结果(没有保存)，为选择的文件打补丁和保存修改，或者为列表内的所有文件打补丁和保存修改。双击动作是预览结果。

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)。

3.2. □ 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。
- 使用他人的/使用我的 从左边或右边的窗格选择块。参阅下节以获得更多关于编辑改变和冲突的信息。
- 他人的在前/我的在前 用于在冲突的文件中合并修改。参阅下节以获得更多关于编辑改变和冲突的信息。
- 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。
- 感知单词的内嵌差异切换内嵌差异的显示方法。在感知单词模式，当改变字符时，整个单词高亮显示。在感知字符模式，高亮显示改变的字符。

感知单词是默认模式，在大多数情况下它工作的很好。例如你修改了变量名，在感知单词模式下整个名称高亮显示，然而在感知字符模式下，你仅仅知道增加或删除了字符。但是，一些内容不能解释为单词，例如二进制文件，这种情况下，感知字符模式工作的更好。

- 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。它显示了他人与本人增加和删除的行数，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。

3.3. 行状态图标

已经修改的行用图标指示发生了什么类型的修改。



已经增加一行。



已经删除一行。



此行已经通过恢复到原始内容而撤销修改。



此行只有空白的修改。当有几行连续被标记时，此段可能被重新排版，从而造成修改。



此行已经使用 TortoiseMerge 作为文本编辑器手工修改。



此行处于冲突状态。



此行处于冲突状态，但是被空白或行结束符的设置掩盖。

3.4. 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(我的)。要应用左边的文件修改(他人的)，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。

你可以象在文本编辑器中一样编辑输出文件。这些被修改的行用铅笔图标标记。请注意，如果你要做上述基于行/块的修改，最好在开始编辑文件之前完成。因为在你开始编辑之后，TortoiseMerge 已经不能跟踪它与原始文件的关联了。

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(已合并)。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，改变结果在已合并文件。

有时候在 Subversion 中的文件会被标记为冲突，但是当你用 TortoiseMerge 察看时，却没有显示冲突。这可能是你选择的空白处理方式造成的。如果你选择忽略行结束符或空白改变，这些行会用 冲突被忽略 图标标记。为了解决冲突，你仍旧需要选择你要使用的版本。

有时冲突很难通过简单选择他人的和我的来解决，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。



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。

3.5. 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。

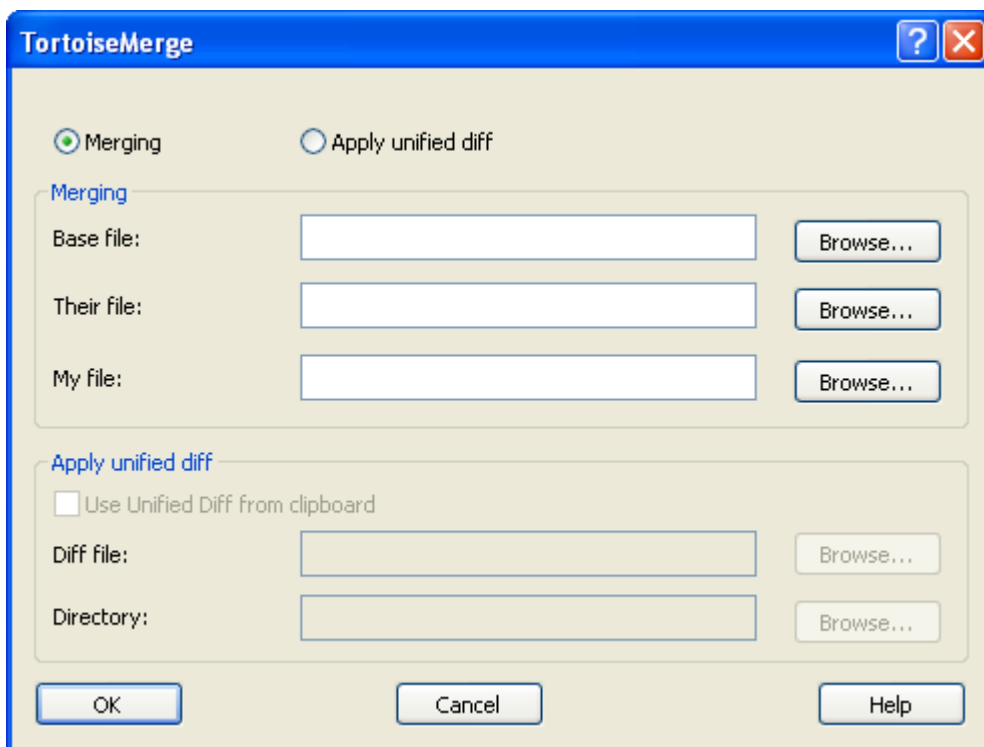


图 3.6. 打开文件对话框

你要作的第一件事，就是决定是否比较/合并文件，或者是否使用补丁。根据你的选择，相应的编辑框和浏览按钮会被激活。

3.5.1. 查看/合并

如果你要比较/合并文件，必须设置两个路径，或三个路径，即基础文件路径，我的文件路径与别人的文件路径。如果你只选择两个文件，那么 TortoiseMerge 会使用两窗口方式或单窗口方式，来显示这两个文件的差异。

如果你要合并三个文件，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。输出文件没有命名，你必须使用File → 另存为 ...来保存结果。

3.5.2. 应用补丁

如果你想应用一个补丁文件，你要设置补丁文件的路径，以及在哪个路径打补丁。



当前仅支持标准差异格式。并且是在 Subversion 工作副本中生成的补丁。其它格式，例如 CVS 生成的补丁文件不受支持！

3.6. □ 设置

3.6.1. □ 常用设置页

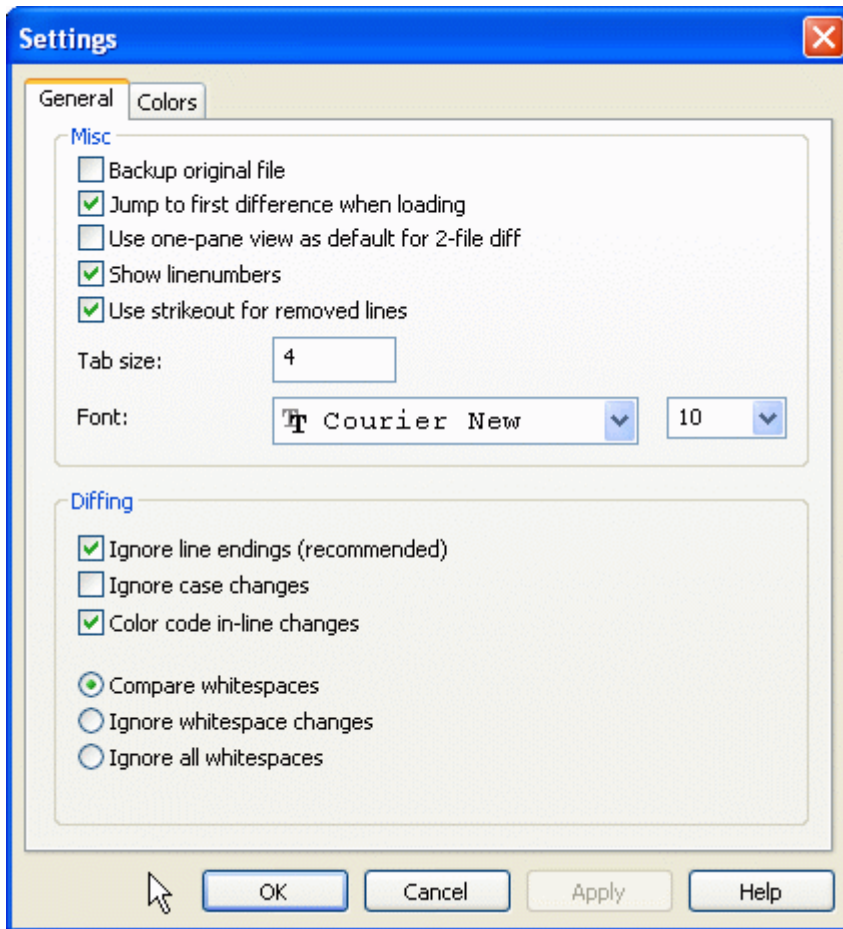


图 3.7. 常用设置页

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。

忽略换行类型 隐藏换行类型的改变。

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。

3. 6. 2. □颜色设置页

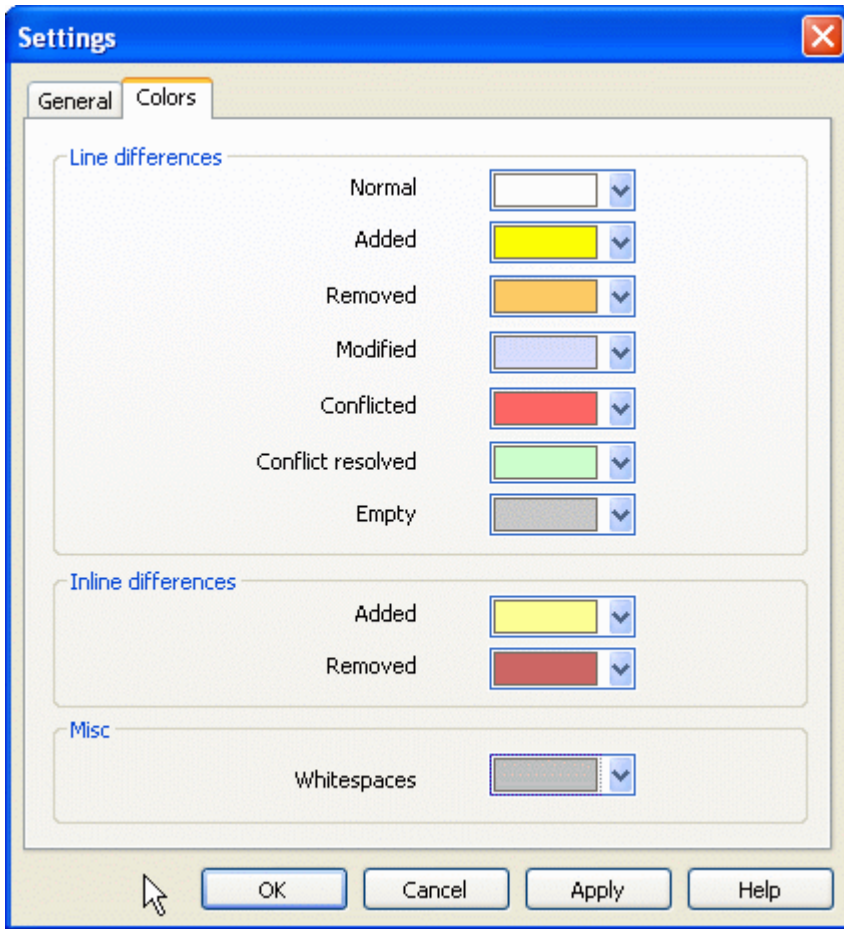


图 3.8. 颜色设置页

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。

冲突已解决

两个文件都修改了同一行，你现在需要选择使用哪个版本。

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。

混杂的空白

用来表示空白的字符，在正文中用不同的颜色来显示

附录 A. 快捷键

列出所有的快捷键和命令。

A. 1. 快捷键

Ctrl-Q

退出程序

Ctrl-C

复制选定的文本到剪贴板

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-鼠标滚轮

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。

B.1. 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"
```

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。
/base	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。
/basename	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。
/theirs	指定用于三窗口方式的 他人的文件，显示在左窗口。
/theirsname	他人的文件名称。代替文件路径在窗口标题栏显示。
/mine	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。
/minename	我的文件名称。代替文件路径在窗口标题栏显示。
/merged	指定用于三路合并的结果文件 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。
/mergedname	合并结果的文件名称。显示在窗口标题显示路径的地方。
/patchpath	使用补丁文件的路径。如果你不设置此路径，TortoiseMerge 将自己查找匹配补丁文件的路径，但会花费很长时间。
/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。
/diff	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	为指定的两个文件 /origfile:"path_to_original_file" 和 /modifiedfile:"path_to_modified_file" 创建统一差异文件(补丁文件)。目标路径设置为 /outfile:"path_to_resulting_patchfile"。如果没有设置 /outfile，会显示文件保存对话框，从而用户可以选择保存补丁文件的位置。注意：如果设置了 /createunifieddiff，所有其它参数都会被忽略。

表 B.1. 命令行选项列表

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，它们会互相比对。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。

术语表

BDB	伯克利DB(Berkeley DB), 版本库可以使用的一种经过充分测试的后台数据库实现, 不能在通过网络共享的文件系统上使用, 伯克利DB是 Subversion 1.2 版本以前的缺省版本库格式。
FSFS	A proprietary Subversion filesystem backend for repositories. Can be used on network shares. Default for 1.2 and newer repositories.
GPO	组策略对象
Revision	每当你提交一组修改, 你会在版本库创建一个“修订版本”, 每个修订代表了版本库树在历史上某一点的状态, 如果你希望回到历史, 你可以回到以前的修订版本N。 另一种情况下, 你可以把修订看作修订版本建立的修改集。
SVN	Subversion 的常见缩写形式。 Subversion 的“svnserve”版本库服务器使用的自定义协议名称。
冲突	当版本库的修改合并到本地修改, 有时候修改发生在同一行, 这种情况下, Subversion 不能自动决定使用文件的那一行, 在提交之前, 你需要手工编辑文件解决冲突。
分支	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”.
切换	就像“Update-to-revision”从历史上改变了工作副本的视点, “Switch”改变了工作副本空间上的视点。当主干和分支只有微小差别时, 这个命令非常有用, 你可以在目录之间跳转, 而只会有很小区别需要传输。
删除	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.
历史	显示文件或目录的历史修订, 也被称为“Log”。
合并	这个过程会查看版本库添加到工作副本的的修改, 而不会破坏你在本地的修改, 有时候这些修改可能不会自动的结合, 也就是冲突了。 在你更新工作副本时会自动合并, 你也可以使用TortoiseSVN的合并命令从另一条分支进行合并。

基础版本(BASE revision)	当前工作副本里的文件或目录的基础版本。是文件或目录最后被检出、更新或者提交时的版本。基础版本通常和HEAD版本不一致。
复制	在 Subversion 版本库，你可以创建一个文件或整个目录树的副本，这是通过“廉价复制”实现的，看起来很像链接到原来的位置，几乎不占用任何空间。创建一个保存历史的副本，这样你就可以跟踪副本之前的修改。
导入	在一个修订里将整个目录导入到版本库的 Subversion 命令。
属性	除了版本控制文件和目录，Subversion 允许你添加版本控制的元数据 - 被称作每个文件和目录的“属性”。每个属性都有一个名称和一个值，非常类似于注册表键。Subversion 有一些内置的特别属性，例如 svn:eol-style。TortoiseSVN 也有一些类似的，例如 tsvn:logminsize，你可以选择名称和值添加你自己的属性。
工作副本	这是你的本地“沙盒”，这个区域是你工作在版本控制文件的地方，它一般存在于你的本地磁盘，你可以使用“Checkout”从版本库创建一个工作副本，然后使用“Commit”将修改传递回版本库。
差异	“显示区别”的快捷方式，当你希望查看你所做修改的时候非常有用。
恢复	Subversion 会为每个更新到工作副本的文件保留一份“原始”副本。如果你做出了修改，并希望取消修改，你可以使用“revert”回到原始状态。
提交	一个 Subversion 操作，用来将本地修改的内容传递回版本库，创建一个新的版本库修订版本。
日志	显示一个文件或是文件夹的版本历史。也就是“历史”。
更新	这个命令将最新的修改从版本库下载到工作副本，合并其他人的修改和工作副本的本地修改。
最新版本(HEAD revision)	版本库里文件或目录的最新版本。
检出	一个 Subversion 命令在空目录通过从版本库下载版本控制的文件来创建本地工作副本。
添加	向你的工作副本中增加文件或者目录时使用的 Subversion 命令。在你提交的时候新的项就会被加入到版本库中。
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.
版本属性(revprop)	就像文件，版本库的每个修订也可以有属性。一些特殊的修订属性会在修订版本创建时自动生成，例如： svn:date svn:author svn:log代表了提交的时间，提交者和日志信息。这些属性可以编辑，但是这些属性都不是版本控制的，所以任何修改都是永久的，不可回退的。

版本库	版本库是进行数据存储和维护的中心。版本库既可以由分布在网络上的若干数据库或者文件组成，也可以存放在用户不需要通过网络就可以直接访问的某个位置。
补丁	如果工作副本只有文本文件有修改，也可以使用 Subversion 的 Diff 命令生成标准区别格式的单文件的修改摘要。这种文件通常被叫做“补丁”，可以邮寄给任何人，使之可以应用到另一个工作副本。一些没有提交访问的人可以通过提交补丁文件给有授权的人来应用补丁，或者是在不确定修改时提交补丁给别人进行评审。
解决	当合并之后版本库的文件进入了冲突状态，必须有人用编辑器解决冲突(或者是TortoiseMerge)，这个过程称作“解决冲突”，当此过程结束，你可以将冲突文件标示为解决，将会运行提交这个文件。
输出	这个命令创建了一个版本控制目录的副本，就像工作副本，但是没有.svn目录。
追溯	这个命令只能用于文本文件，它将会标记每一行来显示版本库修订版本的最后修改的修订和作出修改的人。我们的GUI实现叫做TortoiseBlame，在你将鼠标移到修订版本号码上时，它也会显示时间和日志信息。
重新定位	<p>如果你的版本库移动了，或许是因为移动到了一个新的目录，或者是域名改变，你需要“relocate”你的工作副本，这样你的版本库URL指向新的地址。</p> <p>注意：工作副本必须是指向同一个版本库的同一个位置，是版本库本身移动了。在其他几种情况下，你很有可能是需要“Switch”命令。</p>
锁	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.